July 20–23,
Virtually held at
Kalamata, Greece

General Chairs
Ioannis Z. Emiris
Lihong Zhi
Program Chair
Anton Leykin
Proceedings Editor
Angelos Mantzaflaris

ISSAC'20

Proceedings of the 45th International Symposium on Symbolic and Algebraic Computation

In-Cooperation with:

acm SIGSAM

Sponsored by:

ATHENA · CARGO Lab · Fachgruppe Computeralgebra · Inria · Maplesoft

The Association for Computing Machinery
1601 Broadway, 10th Floor
New York, NY 10019-7434

ISBN: 978-1-4503-7100-1

Additional copies may be ordered prepaid from:

ACM Order Department
PO Box 30777
New York, NY 10087-0777, USA

Phone: 1-800-342-6626 (USA and Canada)
+1-212-626-0500 (Global)
Fax: +1-212-944-1318
E-mail: acmhelp@acm.org
Hours of Operation: 8:30 am - 4:30 pm ET

Foreword

The *International Symposium on Symbolic and Algebraic Computation* (ISSAC) is the premier conference for research in symbolic computation and computer algebra. ISSAC 2020, originally planned to be held at Kalamata, Greece, is the 45th meeting in this series, which took place for the first time in 1966, and annually since 1981. ISSAC 2020 was held in cooperation with the Association for Computing Machinery (ACM) and its Special Interest Group on Symbolic and Algebraic Manipulation (SIGSAM), and was generously supported by few other institutions and organizations, as listed on the following pages.

The conference typically presents a range of invited talks, tutorials, poster sessions, software demonstrations, and vendor exhibits, with its center-piece being peer-reviewed contributed research papers. This year's meeting was scheduled from the 20th to the 23rd of July 2020 so as to include a day of tutorials, and three days of regular papers, invited talks, software demonstrations, and a poster session. However, the Covid-19 pandemic forced us to hold the meeting remotely over the Internet. In order to maximize attendance from all over the world, and taking into consideration the limited interaction possible during an online event, we decided to hold three hours of live talks and discussions daily, from July 20th to July 22nd. All parts of the symposium had to be reduced whereas a few events had to be canceled.

More specifically, regular papers, the cornerstone of the conference, were all discussed online; no talks were broadcast during the live sessions, with the exception of the distinguished paper talk. Presentation files and, optionally, video presentations were posted online before the conference for all regular papers. Software demonstrations and poster sessions were reduced in length, with the latter being somewhat transformed into a session of short communications, given the lack of live interaction. Tutorials were canceled, but we are grateful that two tutorial speakers agreed to include their abstracts in the proceedings; we believe this shows important aspects of current activity in our field, and also gives a better picture of how the live conference would have been. Lastly, we held the three invited talks as well as the *Jenks Prize* talk during the live sessions, and we are grateful to our speakers for their willingness to adapt to this new, remote format.

As always, the ISSAC meeting is a showcase for original research contributions on all aspects of computer algebra and symbolic mathematical computation, including:

Algorithmic aspects:

- Exact and symbolic linear, polynomial and differential algebra
- Symbolic-numeric, homotopy, perturbation and series methods
- Computational algebraic geometry, group theory and number theory, quantifier elimination and logic
- Computer arithmetic
- Summation, recurrence equations, integration, solution of ODEs & PDEs
- Symbolic methods in other areas of pure and applied mathematics
- Complexity of algebraic algorithms and algebraic complexity

Software aspects:

- Design of symbolic computation packages and systems
- Language design and type systems for symbolic computation

- Data representation
- Considerations for modern hardware
- Algorithm implementation and performance tuning
- Mathematical user interfaces
- Use with systems for, e.g., digital libraries, course-ware, simulation and optimization, automated theorem-proving, computer-aided design, and automatic differentiation

Application aspects:

- Applications that stretch the current limits of computer algebra algorithms or systems, use computer algebra in new areas or new ways, or apply it in situations with broad impact.

The ISSAC Program Committee has adhered to the highest standards and practices in the evaluation of submitted papers, producing an average of more than three referee reports per submission. The review process included a round of rebuttal responses for a large number of submissions. All papers submitted to ISSAC were refereed, and accepted or rejected, solely according to their scientific novelty, importance, non-triviality, and rigor. The Program Committee selected 58 papers for publication in these proceedings out of 102 submissions. We gratefully acknowledge the hard work of the Program Committee members. This work would have been impossible without the help of external reviewers. We thank all of them — we received 325 reviews in total[1] — with special thanks going to those who submitted thorough reviews and those who reviewed multiple submissions. We thank all the authors of all the submitted papers, extended abstracts for posters and software presentations, as well as tutorial and invited speakers for their contributions.
In addition to the dedicated effort of all organizers, running a large conference such as ISSAC requires the work of many volunteers and hopefully all of them are credited on the following pages. Without their contributions, the conference would not have been possible.

Ioannis Z. Emiris and **Lihong Zhi** (General Chairs)
Anton Leykin (Program Committee Chair)
Angelos Mantzaflaris (Proceedings Editor)

[1]This year we break with the tradition of publishing the names of external reviewers in the ISSAC proceedings in order to respect all communities where anonymity of referees is strictly guarded.

Table of Contents

ISSAC 2020 Conference Organization

General Chairs:	Ioannis Z. Emiris	*National and Kapodistrian University of Athens, ATHENA Research and Innovation Center, Greece*
	Lihong Zhi	*Academia Sinica, China*
Program Committee Chair:	Anton Leykin	*Georgia Tech, USA*
Local Arrangements Chair:	Ilias S. Kotsireas	*Wilfrid Laurier University, Waterloo, Canada*
Treasurer:	Christos Konaxis	*National and Kapodistrian University of Athens, ATHENA Research and Innovation Center, Greece*
Proceedings Editor:	Angelos Mantzaflaris	*Inria Méditerranée, Université Côte d'Azur, France*
Publicity Chair:	François Lemaire	*Université de Lille, France*
Tutorial Chair:	Wen-Shin Lee	*University of Stirling, UK*
Software Presentations Chair:	Jonathan Hauenstein	*University of Notre Dame, USA*
Poster Chair:	J. Maurice Rojas	*Texas A&M University, USA*

Program Committee:	Peter Bürgisser	*Berlin Technische Universität, Germany*
	Anne Frühbis-Krüger	*Leibniz Universität, Germany*
	Vladimir Gerdt	*Joint Institute for Nuclear Research, Russia*
	Évelyne Hubert	*Inria Méditerranée, France*
	Xiaohong Jia	*Chinese Academy of Sciences, China*
	Gregor Kemper	*München Technische Universität, Germany*
	Christoph Koutschan	*RICAM, Austria*
	Pierre Lairez	*Inria Saclay Île-de-France, France*
	Grégoire Lecerf	*CNRS, École polytechnique, France*
	Anton Leykin (Chair)	*Georgia Tech, USA*
	Diane Maclagan	*University of Warwick, United Kingdom*
	Michael Monagan	*Simon Fraser University, Canada*
	Gabriele Nebe	*RWTH Aachen University, Germany*
	Peter Olver	*University of Minnesota, USA*
	Mohab Safey El Din	*Sorbonne University, France*
	Allan Steel	*University of Sydney, Australia*
	Michael Stillman	*Cornell University, USA*
	Arne Storjohann	*University of Waterloo, Canada*
	Nobuki Takayama	*Kobe University, Japan*
	Maria-Laura Torrente	*University of Genoa, Italy*
	Chee Yap	*New York University, USA*

Poster Committee:	Qi Cheng	*University of Oklahoma, USA*
	Kathlén Kohn	*Kungliga Tekniska Högskolan, Stockholm, Sweden*
	J. Maurice Rojas (Chair)	*Texas A&M University, USA*
	Nitin Saxena	*Indian Institute of Technology, Kanpur, India*
	Timo De Wolff	*Technische Universität Braunschweig, Germany*

Software Presentations Committee:	Danielle Brake	*University of Wisconsin Eau Claire, USA*
	Wolfram Decker	*TU Kaiserslautern, Germany*
	Jonathan Hauenstein (Chair)	*University of Notre Dame, USA*
	Alexander Hulpke	*Colorado State University, USA*

Last but not least, the refereeing work of numerous anonymous external reviewers is gratefully acknowledged.

ISSAC 2020 Sponsors and Supporters

ISSAC acknowledges the generous support of the following institutions:

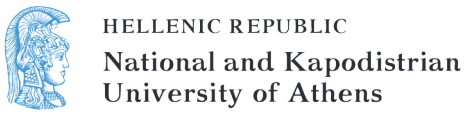
ATHENA Research and Innovation Center, Greece

Computer Algebra Research Group, Wilfrid Laurier University, Canada

Fachgruppe Computer Algebra, Germany

National Institute for Research in Digital Science and Technology, France

MapleSoft, Waterloo, Ontario, Canada



National and Kapodistrian University of Athens, Greece

http://www.issac-conference.org/2020

Invited Talk

Reflections on Elimination Theory

David A. Cox
Department of Mathematics & Statistics
Amherst College
Amherst, MA
dacox@amherst.edu

ABSTRACT

My lecture will survey developments in elimination theory from Newton and Bézout up to modern times. I will discuss the dominance of elimination theory in the 19th century and the challenges it faced in the 20th century with the rise of abstract algebraic geometry. I will also mention the role of the ISSAC community and some personal history.

CCS CONCEPTS

• **Mathematics of Computing → Roots of Nonlinear Equations**.

KEYWORDS

elimination theory, resultant

ACM Reference Format:
David A. Cox. 2020. Reflections on Elimination Theory. In *International Symposium on Symbolic and Algebraic Computation (ISSAC '20), July 20–23, 2020, Kalamata, Greece.* ACM, New York, NY, USA, 4 pages. https://doi.org/10.1145/3373207.3403977

1 EVOLUTION OF ELIMINATION THEORY

I will treat a small number of topics to trace the development of elimination theory. See [30] for a more complete account.

1.1 The 17th Century: Newton

Elimination theory was well-established by the 17th century. Newton [28] knew Bézout's Theorem 100 years before Bézout:

> *Datis duabus curvis invenire puncta intersectionis*
> this is rather a principle then a probleme. But rather propounded of y^e Algebraicall then geometricall solutions & y^t is done by eliminating one of the two unknown quantitys out of y^e equations. From whence it will appeare y^t there are soe many cut points as the rectangle of the curves dimensions.

He applies this to count the number of tangent lines to a curve through a given point:

> a line drawn from a given point may touch a curve of 2 dimensions in 1×2 points, of 3 in 2×3 points, of 4 in 3×4 points,

To see where this comes from, consider a curve $F = 0$ in $\mathbb{P}^2$ and a point $P = (\alpha, \beta, \gamma)$. The tangent lines through P are defined by

$$F = \alpha\frac{\partial F}{\partial x} + \beta\frac{\partial F}{\partial y} + \gamma\frac{\partial F}{\partial z} = 0.$$

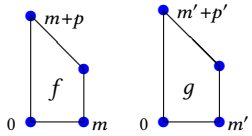
If $\deg(F) = n$, this gives $n(n-1) = (n-1) \times n$ tangent lines.

1.2 The 18th Century: Cramer and Bézout

The strategy used by Newton is also present in Cramer's 1750 [11] version of Bézout's Theorem:

> If one has two variables and two indeterminate equations … of which one is of order m and the other of order n, when, by means of these equations, one of these variables is chased out, the one that remains has a *final equation* of at most mn dimensions.

Bézout's first paper [2] on elimination appeared in 1764, where he includes some sparse versions of Bézout's theorem. For example, let $f(x, y)$ and $g(x, y)$ have Newton polytopes:



Bézout shows that when x is eliminated, the "*équation résultant*" in y has degree $mm' + mp' + m'p$. This is the origin of the term "resultant".

In 1779, Bézout published the 500 page book *Théorie Générale des Équations Algébriques* [3]. Here is his approach to elimination:

> We conceive of each given equation as being multiplied by a special polynomial. Adding up all those products together, the result is what we call the sum-equation. This sum-equation will become the final equation through the vanishing of all terms affected by the unknowns to eliminate.

Notice how the concept of *elimination ideal* is implicit in Bézout

Here is one of many versions of Bézout's Theorem in his book.

Example 1.1. A polynomial in x, y, z is of the *third species* if its monomials $x^{k_1}y^{k_2}z^{k_3}$ satisfy

$$\begin{aligned} &k_1 + k_2 + k_3 \le t \\ &0 \le k_1 \le a_1, \quad 0 \le k_2 \le a_2, \quad 0 \le k_3 \le a_3 \\ &k_1 + k_2 \le b_3, \quad k_1 + k_3 \le b_2, \quad k_2 + k_3 \le b_1, \end{aligned}$$

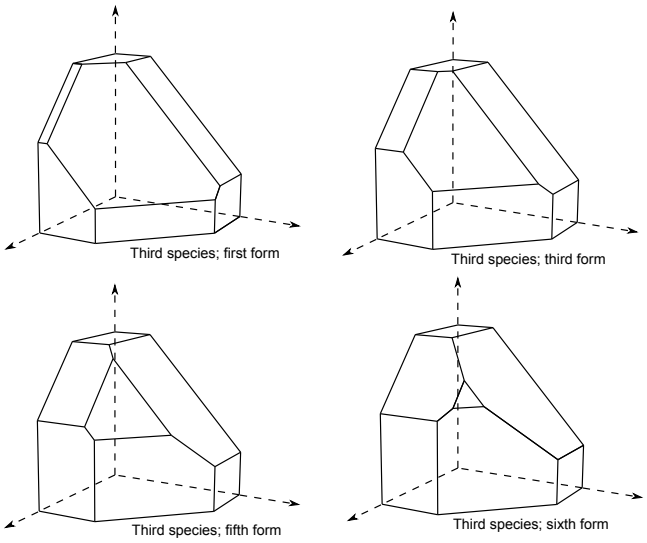
ISSAC '20, July 20–23, 2020, Kalamata, Greece

ACM ISBN 978-1-4503-7100-1/20/07…$15.00
https://doi.org/10.1145/3373207.3403977

for constants $t, a_1, a_2, a_3, b_1, b_2, b_3$. The third species has eight *forms* according to the signs of

$$t - b_1 - b_2 + a_3, \quad t - b_1 - b_2 + a_3, \quad t - b_1 - b_2 + a_3.$$

Here is a modern picture of four forms due to Penchevre [29]:



For readers versed in toric geometry, these forms correspond to chambers of the secondary fan, with flips that give wall crossings.

For the first form of the third species, Bézout calculates the degree of the resultant equation to be

$$\begin{aligned} &t^3 - 3(t-a_1)(t-a_2)(t-a_3) + 3(t-b_1)(t-b_2)(t-b_3) \\ &\quad - 3(a_2+a_3-b_1)(t-b_2)(t-b_3) - 3(a_1+a_3-b_2)(t-b_1)(t-b_3) \\ &\quad - 3(a_1+a_2-b_3)(t-b_1)(t-b_2), \end{aligned}$$

which is the normalized volume of the corresponding polytope. This is the Bernstein-Kushnirenko-Khovanskii Theorem.

Not bad for 1779!

1.3 The 19th Century: Resultants

In 1840, Sylvester [31] described the resultant of two univariate polynomials using a "solid square $(m+n)$ terms deep and $(m+n)$ terms broad", now called the *Sylvester matrix*:

$$\mathrm{Res}(f,g) = \det \underbrace{\begin{pmatrix} c_0 & & & d_0 & & \\ c_1 & \ddots & & d_1 & \ddots & \\ & \ddots & c_0 & & \ddots & d_0 \\ \vdots & & c_1 & \vdots & & d_1 \\ c_n & & \vdots & d_m & & \vdots \\ & \ddots & & & \ddots & \\ & & c_n & & & d_m \end{pmatrix}}_{m \text{ columns}} \underbrace{\qquad\qquad}_{n \text{ columns}}.$$

Resultants were studied by Cayley [7], Brill [4], Kronecker [21], Mertens [24], and *many* others in the 19th century.

Recall the basic idea of the classical resultant: Given $F_0 = \cdots = F_n = 0$ homogeneous in $x_0, \ldots, x_n$, the resultant $\mathrm{Res}(F_0, \ldots, F_n)$ is a polynomial in the coefficients of the F_i such that

$$\mathrm{Res}(F_0, \ldots, F_n) = 0 \iff \begin{array}{l} F_0 = \cdots = F_n = 0 \\ \text{has a solution in } \mathbb{P}^n. \end{array}$$

Resultants were a key tool in elimination theory, which was central to 19th century algebraic geometry. In his 1864 paper *Nouvelles recherches sur l'élimination et la théorie des courbes* [7], Cayley writes

> In the problem of elimination, one seeks the relationship that must exist between the coefficients of a function or system of functions in order that some particular circumstance (or singularity) can occur.

In 1907, Netto and Le Vavasseur wrote a 97 page survey article [27] on elimination theory in *l'Encyclopédie des sciences mathématiques pures et appliquées*. They begin:

> Le grand nombre et la variété des mémoires relatifs à l'élimination rendent difficile une classification rationnelle de ces mémoires.

2 THE FUNDAMENTAL THEOREM OF ELIMINATION THEORY

The 20th century began with a mature theory of resultants through the work of Netto [26] in1900 and Macaulay [22] in 1903 and [23] in 1916. This proved to be a high point for elimination theory; subsequent developments in algebraic geometry moved resultants and elimination theory to the sidelines for quite a while. To set the stage for this story, we will explore the *Fundamental Theorem of Elimination Theory*. Here is a classical version:

Theorem 2.1. *Let $F_1, \ldots, F_r$ be homogeneous in $x_0, \ldots, x_n$ with undetermined coefficients λ. Then there is a* ***resultant system*** *of polynomials $D_1(\lambda), \ldots, D_h(\lambda)$ such that for any choice of coefficients λ' in $\mathbb{C}$, the corresponding polynomials $F_1', \ldots, F_r'$ satisfy*

$$\begin{aligned} &F_1' = \cdots = F_r' = 0 \text{ has a nontrivial solution over } \mathbb{C} \\ &\iff D_1(\lambda') = \cdots = D_h(\lambda') = 0. \end{aligned}$$

Here is the same result in more modern language:

Theorem 2.2. *Let $F_1(\lambda, x), \ldots, F_r(\lambda, x)$ be homogeneous in $x = (x_0, \ldots, x_n)$ with parameters $\lambda = (\lambda_1, \ldots, \lambda_m)$. Then the image of the map*

$$\mathbf{V}(F_1, \ldots, F_r) \subseteq \mathbb{C}^m \times \mathbb{P}^n \longrightarrow \mathbb{C}^m$$

is a closed subvariety of $\mathbb{C}^m$ (defined by the resultant system).

We will discuss three proofs of this result, due to Mertens in 1899, van der Waerden in 1926, and Grothendieck in the 1960s.

2.1 1899: Mertens

For $F_1, \ldots, F_r$, fix $s \gg 0$ and consider products $x^\alpha F_i$ of degree s. In [25], Mertens follows ideas of Kronecker and uses the resultant $\mathrm{Res}(G_1, \ldots, G_n)$ with

$$G_j = \sum_{(\alpha,i)} u_{(\alpha,i),j}\, x^\alpha F_i$$

for new variables $u_{(\alpha,i),j}$. Expanding this as a polynomial in the $u_{(\alpha,i),j}$, the coefficients are polynomials in λ that give a first approximation of the resultant system.

His proof has many more steps and is not easy to follow. The end result is a constructive process for creating the desired resultant system.

2.2 1926: van der Waerden

For $F_1, \dots, F_r$, consider products $x^\alpha F_i$ of degree s, where s is now allowed to be arbitrary. In [34], van der Waerden expresses each $x^\alpha F_i$ as a linear combination of monomials x^β of degree s. This gives a matrix equation

$$M_s(\lambda)\begin{bmatrix}\vdots\\ x^\beta\\ \vdots\end{bmatrix} = \begin{bmatrix}\vdots\\ x^\alpha F_i\\ \vdots\end{bmatrix},$$

where the entries of the Macaulay matrix $M_s(\lambda)$ are polynomials in λ. Then the maximal minors $D_s^{(\ell)}(\lambda)$ of $M_s(\lambda)$ satisfy

$$\begin{aligned}\text{solution in } \mathbb{P}^n \text{ exists for } \lambda' &\iff M_s(\lambda') \text{ drops rank for } s \gg 0\\ &\iff D_s^{(\ell)}(\lambda') = 0 \text{ for all } \ell,\end{aligned}$$

where the first equivalence uses the Hilbert Nullstellensatz. To complete the proof, van der Waerden uses the Hilbert Basis Theorem to reduce to finitely many $D_s^{(\ell)}(\lambda)$.

This proof is nonconstructive and uses powerful tools introduced by Hilbert.

2.3 1960s: Grothendieck

Grothendieck created the language of schemes. A good reference is the book [15] by Hartshorne. A morphism of schemes $f : X \to Y$ is *closed* if it maps closed subsets to closed subsets in the Zariski topology and *universally closed* if $X \times_Y Y' \to Y'$ is closed for every morphism $Y' \to Y$. Then $f : X \to Y$ is *proper* if it is separated, of finite type, and universally closed.

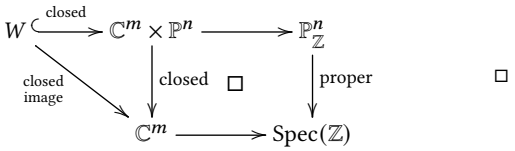
Here is a basic theorem proved by Grothendieck. Let $\mathbb{P}^n_{\mathbb{Z}}$ denote projective space over the integers.

Theorem 2.3. *$\mathbb{P}^n_{\mathbb{Z}} \to \operatorname{Spec}(\mathbb{Z})$ is proper.*

His proof uses the *valuative criterion*, to be discussed in the next section. An immediate corollary is the Fundamental Theorem of Elimination Theory:

Corollary 2.4. *Let $F_1(\lambda, x), \dots, F_r(\lambda, x)$ be as in Theorem 2.2 and let $W = \mathbf{V}(F_1, \dots, F_r) \subseteq \mathbb{C}^m \times \mathbb{P}^n$. Then the image of W under the projection $\mathbb{C}^m \times \mathbb{P}^n \to \mathbb{C}^m$ is closed.*

Proof. This follows from the diagram



□ □

This is Grothendieck's approach to algebraic geometry: once things have been set up properly, basic results become easy.

3 ELIMINATE ELIMINATION THEORY

The proofs of the Fundamental Theorem of Elimination Theory given above illustrate the dramatic changes to algebraic geometry that began with the work of Hilbert in the 1890s. We now say a few words about how we got from resultants to proper morphisms.

3.1 1926: van der Waerden

In his first paper on algebraic geometry [33], van der Waerden writes:

> The rigorous foundation of the theory of algebraic varieties … can be formulated more simply than it has been done so far, without the help of elimination theory, on the sole basis of field theory and of the general theory of ideals in ring domains.

The desire for a "rigorous foundation" came from Hilbert's 15th problem, which concerns the powerful but vague *principle of conservation of number*. A careful treatment required precise definitions of generic point, specialization, and multiplicity.

In a series of papers, van der Waerden addressed these issues. For the most part, he was able to avoid elimination theory. One exception was the *extension of specializations*, where he still needed elimination theory (via his paper [34]).

3.2 1946: Weil

Weil's 1946 book [35] on algebraic geometry uses a "device" of Chevalley to prove the extension of specializations without using elimination theory. In a footnote, Weil says that Chevalley's device

> it may be hoped, finally eliminates from algebraic geometry the last traces of elimination-theory.

Lurking behind the extension of specializations is the valuative criterion for properness used in the proof of Theorem 2.3.

3.3 1970: Abhyankar

By the time of the 1970 International Congress of Mathematics, the Grothendieck revolution in algebraic geometry was in full swing. Van der Waerden's lecture on algebraic geometry in the 20th century mentioned Weil's opinion of elimination theory. In the audience was Ahbyankar, a great fan of 19th century algebraic geometry. This inspired him to write a poem [1] containing the lines

> Eliminate, eliminate, eliminate
> Eliminate the eliminators of elimination theory.
>
> As you must resist the superbourbaki coup
> So must you fight the little bourbakis too.

4 THE REVIVAL OF ELIMINATION THEORY

In turns out that the "resistance" imagined in Abhyankar's poem was already underway.

4.1 The Introduction of Gröbner Bases

Buchberger defined Gröbner bases in his 1965 PhD thesis [5] and gave algorithms in 1970 [6]. Their use in elimination was established by Trinks in 1978 [32]. When combined with computer implementations, the result was a robust elimination theory.

4.2 The Influence of ISSAC

In 1966, the precursor to ISSAC met in Washington DC under the name SYMSAM (Symposium on Symbolic and Algebraic Manipulation) [13]. Speakers included George Collins, Tony Hearn, and Joel Moses, who were key players in cylindrical algebraic decompositions, REDUCE, and MACSYMA respectively.

After a variety of names (SYMSAM, EUROSAM, SYMSAC, EUROCAM, ...), the name ISSAC was adopted in 1988 to reflect the international scope of the enterprise. The ISSAC community has been a key player in Abhyankar's resistance.

4.3 The Modern Theory of Resultants

After years of obscurity, resultants came back in two phases:

- In 1980, Jouanolou began a series of papers [16–20] that studied the classical resultant from a modern point of view.
- In the late 1980s, sparse resultants were introduced and studied. See [14] for a powerful presentation.

As an example of recent work, we mention D'Andrea, Jeromino and Sombra, who in April 2020 posted the preprint [12] proving a conjecture of Canny and Emiris about a quotient formula for the sparse resultant. This generalizes a formula due to Macaulay [22] from 1903 for the classical resultant.

5 PERSONAL REFLECTIONS

I entered graduate school in 1970, a year mentioned in Section 3, and learned algebraic geometry in full Grothendieck mode. My thesis was *Tubular neighborhoods in the étale topology*, and my first published paper was *Homotopy limits and the homotopy type of functor categories*. This is not computational algebraic geometry!

In the 1980s, my research shifted to elliptic surfaces. I remember using the original *Macaulay* in 1985 to study an example. Being clueless about complexity issues, each coefficient was a new variable. Needless to say, my computation was not a success.

In 1988, Don O'Shea visited Kaiserslautern and learned about Gröbner bases. He suggested that we write an undergraduate text on this subject. After teaming up with John Little (see [9] for the full story), our book *Ideals, Varieties and Algorithms* [8] was published in 1992. The book has been successful, in part because we wrote it for American undergraduates. This forced us to take a down-to-earth approach with few pre-requisites, which made the book accessible to people in many fields outside of mathematics.

Not surprisingly, my favorite chapter of the book is Chapter 3, titled *Elimination Theory*.

ACKNOWLEDGMENTS

This lecture is based on my CBMS lecture series *Applications of Polynomial Systems*, given at Texas Christian University in Fort Worth, Texas in Summer 2018. An expanded version appears in the book [10] of the same title.

The account of elimination theory given in the lectures and the book failed to acknowledge the role of the ISSAC community in the revival of elimination theory. I am pleased to have the opportunity to correct this omission.

REFERENCES

[1] S. Abhyankar, 1972. *Polynomials and power series*, Math. Intelligencer 3, Springer. Reprinted in *Algebra, Arithmetic and Geometry with Applications* (C. Christensen, G. Sundaram, A. Sathaye and C. Bajaj, eds.), Springer, New York, 2004, 783–784.
[2] E. Bézout, 1764. *Sur le degré des équations résultantes de l'évanouissement des inconnues*, Histoire de l'Académie Royale des Sciences, 288–338.
[3] E. Bézout, 1779. *Théorie générale des équations algebriques*, Ph.-D. Pierres, Paris. English translation *General Theory of Algebraic Equations* by Eric Feron, Princeton Univ. Press, Princeton, NJ, 2006.
[4] A. Brill, 1880. *Ueber eine Eigenschaft der Resultante*, Math. Annalen 16, 345–347.
[5] B. Buchberger, 1965. *Ein Algorithmus zum Auffinden der Basiselemente des Restklassenringes nach einem nulldimensionalen Polynomideal*, Ph.D. Thesis, University of Innsbruck.
[6] B. Buchberger, 1970. *Ein algorithmisches Kriterium für die Lösbarkeit eines algebraischen Gleichungssystems*, Aequationes mathematicae 4, 374–383.
[7] A. Cayley, 1864. *Nouvelles recherches sur l'élimination et la théorie des courbes*, J. Reine Angew. Math. 63, 34–39.
[8] D. Cox, J. Little and D. O'Shea, 2015. *Ideals, Varieties and Algorithms*, Fourth Edition, Springer, New York.
[9] D. Cox, J. Little and D. O'Shea, 2016. *The story of Ideals, Varieties and Algorithms*, Notices of the AMS 63, 6, 626–628.
[10] D. Cox, with contributions by C. D'Andrea, A. Dickenstein, J. Hauenstein, H. Schenck and J. Sidman, 2020. *Applications of Polynomial Systems*, AMS, Providence, RI.
[11] G. Cramer, 1750. *Introduction à l'analyse des lignes courbes algébriques*, Frères Cramer et Cl. Philibert, Genêve.
[12] C. D'Andrea, G. Jeronimo and M. Sombra, 2020. *The Canny-Emiris conjecture for the sparse resultant*, arXiv:2004.14622[math.AC].
[13] R. W. Floyd (Ed.), 1966. *Proc. of ACM Symposium on Symbolic and Algebraic Manipulation (SYMSAM '66), Washington D.C.* Comm. ACM 9, 547–643.
[14] I. Gel'fand, M. Kapranov and A. Zelevinsky, 1994. *Discriminants, Resultants, and Multidimensional Determinants*, Birkhäuser, Boston.
[15] R. Hartshorne, 1977. *Algebraic Geometry*, Springer, New York.
[16] J.-P. Jouanolou, 1980. *Idéaux résultants*, Adv. Math. 37, 212–238.
[17] J.-P. Jouanolou, 1991. *Le formalisme du résultant*, Adv. Math. 90, 117–263.
[18] J.-P. Jouanolou, 1995. *Aspects invariants de l'élimination*, Adv. Math. 114, 1–174.
[19] J.-P. Jouanolou, 1996. *Résultant anisotrope, compléments et applications*, Electron. J. Combin. 3, no. 2, Research Paper 2.
[20] J.-P. Jouanolou, 1997, *Formes d'inertie et résultant: un formulaire*, Adv. Math. 126, 119–250.
[21] L. Kronecker, 1882. *Grundzüge einer arithmetischen Theorie der algebraischen Grössen*, J. Reine Angew. Math. 92, 1–122.
[22] F. Macaulay, 1903. *On some Formulæ in Elimination*, Proc. London Math. Soc. 35, 3–27,
[23] F. Macaulay, 1916. *The Algebraic Theory of Modular Systems*, Cambridge Univ. Press, Cambridge.
[24] F. Mertens, 1886. *Über die bestimmenden Eigenschaften der Resultante von n Formen mit n Veränderlichen*, Sitzungsberichte der Mathematisch-Naturwissenschaftlichen Classe der Kaiserlichen Akademie der Wissenschaften, II. Abtheilung 93, 527–566.
[25] F. Mertens, 1899. *Zur Theorie der Elimination*, Teil II, Sitzungsber. Akad. Wien 108, 1344–1386.
[26] E. Netto, 1900. *Vorlesungen über Algebra*, Volume II, Teubner, Leipzig.
[27] E. Netto and R. Le Vavasseur, 1907. *Fonctions rationnelles*, in *Encyclopédie des sciences mathématiques pures et appliquées* Tome I, Volume 2 (J. Molk, ed.), Gauthiers-Villars, Paris and B. G. Teubner, Leipzig, 1–232.
[28] I. Newton, 1972. *The Mathematical Papers of Isaac Newton*, Volume II, (D. T. Whiteside, ed.) Cambridge Univ. Press, Cambridge, p. 177.
[29] E. Penchèvre, 2016. *Etienne Bezout on elimination theory*, arXiv: 1606.03711[math.HO].
[30] E. Penchèvre, 2006, *Histoire de la théorie de l'élimination*, Ph.D. Thesis, l'Université Paris VII.
[31] J. Sylvester, 1840. *A Method of determining by mere inspection the derivatives from two equations of any degree*, Philos. Mag. XVI, 132–135.
[32] W. Trinks, 1978. *Über B. Buchbergers Verfahren, Systeme algebraischer Gleichungen zu lösen*, J. Number Theory 10, 475–488.
[33] B. van der Waerden, 1926. *Zur Nullstellentheorie der Polynomideale*, Math. Annalen 96, 183–208.
[34] B. van der Waerden, 1926. *Ein algebraisches Kriterium für die Lösbarkeit von homogenen Gleichungen*, Nederl. Akad. Wetensch. Proc. 29, 142–149.
[35] A. Weil, 1946. *Foundations of Algebraic Geometry*, AMS, Providence, RI.

Positive Solutions of Sparse Polynomial Systems

Alicia Dickenstein
Universidad de Buenos Aires and IMAS (UBA-CONICET)
Buenos Aires, Argentina
alidick@dm.uba.ar

ABSTRACT

My lecture will survey some classical and recent lower and upper bounds for the number of positive solutions of systems of n sparse polynomial systems in n variables, including basic questions that are open. This is only a short summary of the talk.

CCS CONCEPTS

• **Mathematics of Computing → Roots of Nonlinear Equations**.

KEYWORDS

sparse polynomial systems, positive roots

ACM Reference Format:
Alicia Dickenstein. 2020. Positive Solutions of Sparse Polynomial Systems. In *International Symposium on Symbolic and Algebraic Computation (ISSAC '20), July 20–23, 2020, Kalamata, Greece*. ACM, New York, NY, USA, 3 pages. https://doi.org/10.1145/3373207.3403978

1 CLASSICAL UNIVARIATE BOUNDS

The beautiful classical Descartes' rule of signs to bound the number of positive roots of univariate polynomials (counted with multiplicity) was proposed by René Descartes in 1637 in *La Géometrie*, an appendix to his *Discours de la Méthode* [26]. There are other several classical results in the univariate case, as the Budan-Fourier Theorem to bound the number of roots in an interval $(a, b]$ or Sturm's theorem, which gives an exact count in this interval of the number of distinct real roots. We refer the reader to the "Bible" [1] for further theoretical as well as algorithmic details.

Given a univariate real polynomial

$$f(x) = f_0 + \sum_{j=1}^{r} f_j x^j,$$

Descartes'rule of signs says that the number of positive real roots $n_+(f)$ of f is bounded by the number of sign variations $v(f)$ in the ordered sequence $\sigma(f_0), \ldots, \sigma(f_r)$ of the signs of the coefficients (where we discard the 0's in this sequence and we add a 1 each time two consecutive signs are different). In fact, Descartes' rule of signs holds for generalized polynomials in which the exponents are real. For instance, if $f = f_0 + 3x - 90x^6 + 2x^8 + x^{111}$, the sequence of signs (discarding 0's) is: $\sigma(f_0), +, -, +, +$. So, $v(f)$ equals 2 if $f_0 \geq 0$ and 3 if $f_0 < 0$. Then, f has at most 2 or 3 positive real roots. Moreover, $n_+(f)$ has the same parity as $v(f)$, so in the second case we can assert that $n_+(f) > 0$.

It is interesting to remark the following facts for a real polynomial $f(x)$ as before:

(i) This rule is extremely simple and it is sharp in the sense that for any sign variation there are polynomials f for which $v(f) = n_+(f)$. This happens in case all roots are real, for instance when f is the characteristic polynomial of a symmetric matrix.
(ii) Considering $v(f(-x))$ we can likewise bound the number of negative roots of f. It follows that the number of real roots of f can be bounded in terms of the number of nonzero coefficients and not by the degree.
(iii) Indeed, Descartes' rule is not a result about our particular polynomial f but rather about the family of all polynomials with the same monomials and coefficients of the same sign as those of f. Moreover, it is even independent of the *ordered* sequence of exponents with nonzero coefficients, in the following sense: if we replace the exponents of f by another increasing sequence of the same length, the bound is the same.

Sturm's theorem translates into an algorithm that has many branchings when one tries to use it for parametric polynomials. The reader can just experiment with the family of degree 3 univariate polynomials $c_0 + c_1x + c_2x^2 + c_3x^3$, where $c_0, \ldots, c_3$ do not take fixed a priori values.

2 SOME BY NOW CLASSICAL MULTIVARIATE RESULTS

Deciding the number of positive real roots is indeed a question of quantifier elimination and thus, it can in theory be effectively computed. Besides the algorithms implementing quantifier elimination, there are other general algorithmic tools. For a given choice of coefficients, a particular system can be solved using different methods of numerical algebraic geometry. The properties of Hermite's quadratic form give rise to algorithms to symbolically compute the number of positive real roots (that is, roots with positive coordinates) for systems of multivariate polynomials with a finite number of complex roots [1]. This is implemented for example in Singular [12], where there is a command called `firstoct`. This beautiful result is not practical to decide on the number of positive roots of parametric systems.

If we have a sparse system of n Laurent polynomials in n variables of the form

$$f_i = \sum_{a \in A} c_{ia}\, x^a, \quad c_{ia} \in \mathbb{R}, \quad i = 1 \ldots n, \tag{1}$$

ISSAC '20, July 20–23, 2020, Kalamata, Greece

ACM ISBN 978-1-4503-7100-1/20/07…$15.00
https://doi.org/10.1145/3373207.3403978

where $A \subset \mathbb{Z}^n$ is a finite set, the number of isolated common roots in the complex torus $(\mathbb{C}^*)^n$ is bounded by the normalized volume $\mathrm{vol}_{\mathbb{Z}}(A)$ of the convex hull of A, and this bound is sharp for generic coefficients, generalizing and improving the Bézout bound in terms of the degrees. As conjectured from fact (ii) above, Khovanskii [20] proved the important result that the maximal number of non-degenerate positive solutions (that is, with non-vanishing Jacobian determinant) of $f_1, \ldots, f_n$ is bounded above in terms of the cardinality $|A|$ of A, namely by $2^{\binom{|A|-1}{2}}(n+1)^{|A|-1}$, which is independent of the degrees and of $\mathrm{vol}_{\mathbb{Z}}(A)$. This bound, which was a breakthrough, is far from sharp. It was significantly reduced in [7]. This bound is still not sharp, but for $|A| - n$ fixed it is asymptotically of the right order as n goes to infinity. There are very few particular sharp bounds. Moreover, these bounds take into account the monomials, but not the particular coefficients as in the case of Descartes' rule of signs.

3 RECENT UPPER BOUNDS IN THE MULTIVARIATE CASE

There is not a single generic answer to the number of common real roots of polynomials $f_1, \ldots, f_n$ as in (1). Indeed, if one follows a curve of coefficients (with fixed supports) the only way in which the number of real roots with nonzero coordinates can change is when the curve *crosses* the discriminant hypersurface in coefficient space. For general configurations A, the closure of the locus of coefficients (c_{ia}) for which the system $f_1 = \ldots = f_n = 0$ has a degenerate root, is a hypersurface defined by the vanishing of the associated discriminant. In the complex case, it has real codimension two and so it does not disconnect the space of coefficients. When we restrict to real coefficients, its complement consists of disconnected chambers and in each of these the number of real roots is constant. If we moreover intersect the discriminant complement with the complement of the resultant varieties defined for $i = 1, \ldots, n$ as the closure of the coefficients for which $f_1, \ldots, f_n$ have a common root in the i-th coordinate hyperplane, then in each chamber the signs of the coordinates of the common roots do not change. So, one could in principle find one choice of coefficients in each of the finitely many chambers and then compute the number of real roots for all the associated systems. This computation is in theory effective but in practice unfeasible for most systems of interest.

Ideally, we would be interested in a *simple* Decartes' rule of signs in the multivariate sparse case that gives a sharp upper bound taking into account both the exponents A and the coefficients (c_{ia}), as well as their interactions, and that moreover satisfies properties similar to the facts (i), (ii), (iii) we detailed in the univariate case. Interestingly, the first partial multivariate generalization was devised in the paper [21], that arose from the study of the problem of determining multistationarity in chemical reaction networks, that is, the occurrence of parameters for which a polynomial system associated to the kinetics has at least two positive roots. Part of this result was hidden in the article [11] in the area of geometric modeling. In the paper [9], a variety of symbolic approaches with multiple algorithms and computer algebra systems is used to identify multistationarity parameters in models of biological networks. We used in [15] the existence of triangular forms for this problem. We refer to Chapter 5 in [10] and [13, 14] for general notions and algebro-geometric questions in the study of biochemical reaction networks under mass-action kinetics.

We found in [21] reasonably simple conditions on the signs of the minors of a matrix containing the exponents as columns and a matrix formed by the coefficients of the polynomials in the system (that is, on their associated oriented matroids), to ensure that the number of positive real roots cannot be bigger than one, that is, the absence of multistationarity. This is an important question in many applied domains and translates the injectivity of a family of polynomial maps to sign conditions of vectors in a linear subspace. This question of signs was already studied in [23]. A more detailed study of sign conditions for the injectivity of a family of generalized polynomial map over the positive orthant is presented in [22]. We considered in [2] the circuit case in which $|A| = n + 2$, and already in this instance one can see that the multivariate question is far more involved and that an ordering of the minors and linear combinations of them are needed. We were able to bound the number of positive solutions of the system by the sign variation of an associated sequence, which is sharp in the sense that it is possible to find supports A for which it is attained. Moreover, we gave conditions under which this sign variation has the same parity of the number of positive solutions. We then refined this bound in [3] to get a sharp estimate for any possible A which is a circuit. These results hold for families polynomials with fixed support and parametric coefficients and satisfy conditions similar to the facts (i), (ii) and (iii) above.

It is a wide open question to devise a general multivariate version of Descartes' rule of signs (even a conjectural one).

4 RECENT LOWER BOUNDS IN THE MULTIVARIATE CASE

Most lower bounds for the number of positive solutions of a system of n sparse polynomials in n variables are not about a particular system but concern a parametric family of sparse polynomials. In the paper [19] there is a lower bound for the maximal number of positive solutions for multivariate sparse systems with possibly different fixed supports. In [24], it is shown how to construct sparse polynomial systems with non-trivial lower bounds on their numbers of real solutions, using combinatorial tools. We also refer to [25]. Tools from tropical geometry are used in [16] to construct real bivariate polynomial systems with five monomials that have more than the previously known lower bound of six positive solutions. In [6], a version of Viro's method based on regular triangulations of the support of the given polynomials is used to get new lower bounds for the maximal number of positive solutions to polynomial systems with prescribed numbers of monomials and variables, as well as the asymptotics of these bounds. We extended in [4, 17] their ideas and applied them to finding open regions in parameters space where the number of positive solutions is at least two in some networks of interest in systems biology, as enzymatic cascades in which the number of variables and parameters increase linearly with n. We were also able to find open regions of parameters with a number of positive roots growing with the number of variables for multisite phosphorylation networks in [18], together

with a computer algebra implementation that works for low dimensions.

These previous results concern the existence of parameters for which the number of positive solutions can be bounded below, but they don't give in general answers for particular sparse polynomial systems. Using Brouwer degree theory and Gale duality we found in [5] sufficient sign conditions on a given sparse system $f_1 = \cdots = f_n = 0$ as above, that ensure the existence of at least one solution (see also [27]). These sufficient conditions are for sure not necessary. In the case of integer exponents, they are related to algebraic properties studied in the context of lattice ideals associated with the configuration A. In a different direction, there is a nice result about the existence of positive solutions of the steady state equations for weakly reversible networks in [8].

It would be interesting to have other general results based on the structure and signs of the system that ensure the existence of positive roots.

ACKNOWLEDGMENTS

I am very grateful to the Program Committee of ISSAC 2020 for the invitation to speak. Alas, this ISAAC edition will not take place in Greece as planned, but we will all do our best to have a successful online conference.

REFERENCES

[1] S. Basu, R. Pollack and M.-F. Roy, 2016. *Algorithms in Real Algebraic Geometry*, volume 10 of Algorithms and Computation in Mathematics. Springer-Verlag Berlin Heidelberg, 2nd. ed.

[2] F. Bihan and A. Dickenstein, 2017. *Descartes' Rule of Signs for Polynomial Systems supported on Circuits*, Int. Math. Res. Notices Vol. 2017 (22), 6867–6893.

[3] F. Bihan, A. Dickenstein and J. Forsgård, 2020. *Optimal Descartes' Rule for Systems supported on Circuits*, to be posted soon.

[4] F. Bihan, A. Dickenstein and M. Giaroli, 2020. *Lower bounds for positive roots and regions of multistationarity in chemical reaction networks*, J. Algebra 542, 367–411.

[5] F. Bihan, A. Dickenstein and M. Giaroli, 2019. *Sign conditions for the existence of at least one positive solution of a sparse polynomial system*, arXiv:1908.05503.

[6] F. Bihan, F. Santos and P.-J. Spaenlehauer, 2018. *A polyhedral method for sparse systems with many positive solutions*, SIAM J. Appl. Algebra Geometry 2 (4), 620–645.

[7] F. Bihan and F. Sottile, 2007. *New fewnomial upper bounds from Gale dual polynomial systems*. Mosc. Math. J. 7 (3).

[8] B. Boros, 2019. *Existence of positive steady states for weakly reversible mass-action systems*, SIAM J. Math. Anal. 51 (1), 435–449.

[9] R. Bradford, J. H. Davenport, M. England, H. Errami, V. Gerdt, D. Grigoriev, C. Hoyt, M. Kosta, O. Radulescu, T. Sturm, A. Weber, 2020. *Identifying the parametric occurrence of multiple steady states for some biological networks*, J. Symb. Comput. 98, 84–119.

[10] D. Cox, with contributions by C. D'Andrea, A. Dickenstein, J. Hauenstein, H. Schenck and J. Sidman, 2020. *Applications of Polynomial Systems*, AMS, Providence, RI.

[11] G. Craciun, L. García Puente and F. Sottile, 2010. *Some geometrical aspects of control points for toric patches, in Mathematical methods for curves and surfaces*, M. Dæhlen, M. Floater, T. Lyche, J.-L. Merrien, K. Mørken and L. Schumaker, eds., Lecture Notes in Comput. Sci. 5862, Springer, Berlin, 111–135.

[12] W. Decker, G.-M. Greuel, G. Pfister and H. Schönemann, 2018. SINGULAR 4-1-1 — A computer algebra system for polynomial computations. http://www.singular.uni-kl.de.

[13] A. Dickenstein, 2016. *Biochemical reaction networks: an invitation for algebraic geometers*, MCA 2013, Contemporary Mathematics 656, 65–83.

[14] A. Dickenstein, 2019. *Algebra and Geometry in the Study of Enzymatic Cascades*, in: World Women in Mathematics 2018, Proceedings of the First World Meeting for Women in Mathematics $(WM)^2$, Araujo, C., Benkart, G., Praeger, C., Tanbay, B. (Eds.), 57–81.

[15] A. Dickenstein, M. Pérez Millán, A. Shiu and X. Tang, 2019. *Multistationarity in Structured Reaction Networks*, Bull. Math. Biol. 81, 1527–1581.

[16] B. El Hilany, 2020. *Constructing polynomial systems with many positive solutions using tropical geometry*, To appear: Revista Matematica Complutense, arXiv:1703.02272.

[17] M. Giaroli, A. Dickenstein and F. Bihan, 2019. *Regions of multistationarity in cascades of Goldbeter-Koshland loop*, J. Math. Biol. 78(4), 1115–1145.

[18] M. Giaroli, R. Rischter, M. Pérez Millán and A. Dickenstein, 2019. *Parameter regions that give rise to 2[n/2]+1 positive steady states in the n-site phosphorylation system* Math. Biosci. Eng. 16 (6), 7589–7615.

[19] I. Itenberg and M. F. Roy, 1996. *Multivariate Descartes' rule*, Beitr. Algebra Geom. 37 (2), 337–346.

[20] A. G. Khovanskiĭ, 1991. *Fewnomials*, Translations of Mathematical Monographs 88, American Mathematical Society, Providence, RI.

[21] S. Müller, E. Feliu, G. Regensburger, C. Conradi, A. Shiu and A. Dickenstein, 2016. *Sign conditions for injectivity of generalized polynomial maps with appl. to chemical reaction networks and real algebraic geometry*, Found. Comput. Math. 16 (1), 69–97..

[22] S. Müller, J. Hofbauer, and G. Regensburger, 2019. *On the bijectivity of families of exponential/generalized polynomial maps*, SIAM Journal on Appl. Algebra Geometry 3, 412–438.

[23] R.T. Rockafellar, 1969. *The elementary vectors of a subspace of* $\mathbb{R}^n$. Combinatorial Mathematics and Its Applications. In Proc. of the Chapel Hill Conf., University of North Carolina Press, 104–127.

[24] E. Soprunova and F. Sottile, 2006. *Lower bounds for real solutions to sparse polynomial systems*. Adv. Math. 204 (1), 116–151.

[25] F. Sottile, 2011. *Real solutions to equations from geometry*, University Lecture Series 57, American Mathematical Society, Providence, RI.

[26] D. J. Struik (ed.), 1969. *A source book in mathematics, 1200-1800*, Source Books in the History of the Sciences. Cambridge, Mass., Harvard University Press XIV.

[27] J. Wang, 2019. *Systems of polynomials with at least one positive real zero*, J. Algebra Appl., 2050183.

Ubiquity of the Exponent of Matrix Multiplication

Lek-Heng Lim*
lekheng@galton.uchicago.edu
University of Chicago
Chicago, IL, USA

Ke Ye*
keyk@amss.ac.cn
Chinese Academy of Sciences
Beijing, China

ABSTRACT

The asymptotic exponent of matrix multiplication is the smallest ω such that one may multiply two $n \times n$ matrices or invert an $n \times n$ matrix in $O(n^{\omega+\varepsilon})$-complexity for $\varepsilon > 0$ arbitrarily small. One of the biggest open problem in complexity theory and numerical linear algebra is its conjectured value $\omega = 2$. This article is about the universality of ω. We will show that ω is not only the asymptotic exponent for the product operation in matrix algebras but also that for various infinite families of Lie algebras, Jordan algebras, and Clifford algebras. In addition, we will show that ω is not just the asymptotic exponent for matrix product and inversion but also that for the evaluation of any matrix-valued polynomial and rational functions of matrix variables.

CCS CONCEPTS

• **Theory of computation → Algebraic complexity theory**.

KEYWORDS

Matrix multiplication, bilinear complexity, Lie algebras, Jordan algebras, Clifford algebras, matrix polynomials

ACM Reference Format:
Lek-Heng Lim and Ke Ye. 2020. Ubiquity of the Exponent of Matrix Multiplication. In *International Symposium on Symbolic and Algebraic Computation (ISSAC '20), July 20–23, 2020, Kalamata, Greece*. ACM, New York, NY, USA, 4 pages. https://doi.org/10.1145/3373207.3403979

1 INTRODUCTION

Fifty years ago, Strassen announced a surprising algorithm that gives the product of two $n \times n$ matrices in $O(n^{\log_2 7})$ multiplications [17]. He also showed that the number of additions may be bounded by a constant factor of the number of multiplications and that an $n \times n$ linear system can be solved in the same time complexity it takes to multiply two matrices. The best possible exponent ω so that one may multiply two $n \times n$ matrices in $O(n^{\omega+\varepsilon})$-time for any $\varepsilon > 0$ is now known as the asymptotic *exponent of matrix multiplication* (the extra ε term is so that one may account for algorithms running in, say, $O(n^{\omega} \log n)$-time).

*Both authors contributed equally to this research.

ISSAC '20, July 20–23, 2020, Kalamata, Greece

ACM ISBN 978-1-4503-7100-1/20/07…$15.00
https://doi.org/10.1145/3373207.3403979

Strassen's upper bound for ω, i.e., $\log_2 7 \approx 2.8074$, has been improved over the years. Roughly speaking, there are three methods to sharpen the upper bound. The first is the line of argument introduced by Strassen in his *laser method* and generalized in [4, 16]. Coppersmith and Winograd employed it to improve the upper bound to 2.3755 [10], a record that held for twenty five years until it was improved by Vassilevska Williams [19] and Le Gall [14] to 2.373. Nevertheless, it is now known [1] that the laser method has its limitations and cannot yield an upper bound for ω that is better than 2.3. The second approach, based on algebraic geometry and pioneered by Landsberg about fifteen years ago [11] provides powerful techniques for obtaining *lower bounds* for the complexity of matrix multiplication [3, 6, 7, 11–13]. The third approach is the representation theoretic framework pioneered by Cohn and Umans [8, 9, 18] by embedding matrices into a group algebra of a suitably chosen group and using group representation theory to study the complexity of matrix multiplication.

The results in this article may be viewed as a fourth approach. We show that ω is the asymptotic exponent for the product operations in many different types of algebras including Lie, Jordan, and Clifford algebras. Consequently, the complexity of the product operation in any of these algebras provides an alternative route towards determining or bounding the value of ω. Furthermore, we show that the asymptotic complexity of evaluating univariate or multivariate matrix-valued functions of matrices is also given by ω, providing yet another route towards obtaining its value.

This article is an extended abstract of [15], intended only to highlight a few key results therein. The reader will find the proofs of these results as well as a far more extensive discussion in [15].

2 BILINEAR AND MULTIPLICATIVE COMPLEXITY

Let U, V, W be vector spaces over a field $\mathbb{F}$ and let $\beta : U \times V \to W$ be a bilinear map. The *bilinear complexity* of β is defined to be the minimum number of multiplications over $\mathbb{F}$ required to evaluate β. Equivalently, the bilinear complexity of β is the *rank* of its structure tensor $T_\beta \in U^* \otimes V^* \otimes W$, which is denoted by $\operatorname{rank}(T_\beta)$ and is defined to be the minimal r such that

$$T_\beta = \sum_{j=1}^{r} u_j^* \otimes v_j^* \otimes w_j, \quad u_j^* \in U^*, v_j^* \in V^*, w_j \in W, j = 1, \dots, r.$$

We refer readers to [5, 20] for more information. Let

$$S := \{(U_n, V_n, W_n, \beta_n) : n \in \mathbb{N}\} \tag{1}$$

be a sequence of bilinear maps $\beta_n : U_n \times V_n \to W_n$. We define the *asymptotic exponent* of S to be

$$\omega(S) := \liminf_{n\to\infty} \log_n(\operatorname{rank}(T_{\beta_n})). \tag{2}$$

For the special case when $U_n = V_n = W_n$ for all $n \in \mathbb{N}$, we will just write

$$S = \{(V_n, \beta_n) : n \in \mathbb{N}\}$$

instead of the more cumbersome $S = \{(V_n, V_n, V_n, \beta_n) : n \in \mathbb{N}\}$ as in (1). This will be case when V_n is an algebra and $\beta_n : V_n \times V_n \to V_n$ the product operation of the algebra.

Before going further we shall look at two examples. Let

$$\mathrm{M}(\mathbb{F}) \coloneqq \{(\mathbb{F}^{n\times n}, \mu_n) : n \in \mathbb{N}\}, \tag{3}$$

where $\mathbb{F}^{n\times n}$ is the vector space of $n \times n$ matrices over $\mathbb{F}$ and μ_n is the matrix multiplication map sending a pair of matrices (X, Y) to their product XY. Then $\omega(\mathrm{M}(\mathbb{F}))$ is exactly the exponent of matrix multiplication ω described in Section 1.

Take another example where the matrix product in (3) is replaced by commutator product $[X, Y] = XY - YX$,

$$\mathfrak{gl}(\mathbb{F}) \coloneqq \{(\mathbb{F}^{n\times n}, [\cdot, \cdot]) : n \in \mathbb{N}\}.$$

Note that previously we have $\mathrm{M}_n(\mathbb{F})$, the usual matrix algebra, i.e., the set $\mathbb{F}^{n\times n}$ equipped with standard matrix multiplication, but here we have $\mathfrak{gl}_n(\mathbb{F})$, i.e., the set $\mathbb{F}^{n\times n}$ equipped with commutator product. In this case $\mathfrak{gl}_n(\mathbb{F})$ is the Lie algebra of the general linear group $\mathrm{GL}_n(\mathbb{F})$. These two products are of a very different nature — for example, the commutator product is not associative but it satisfies the Jacobi identity.

It is easy to see that $\omega(\mathfrak{gl}(\mathbb{F})) \leq \omega(\mathrm{M}(\mathbb{F})) = \omega$ and, with a bit more work, that equality holds.

Proposition 2.1. $\omega(\mathfrak{gl}(\mathbb{F})) = \omega(\mathrm{M}(\mathbb{F})) = \omega$.

We will see in Section 4 that Proposition 2.1 extends to all infinite families of semisimple Lie algebras $\mathfrak{sl}_{n+1}(\mathbb{F})$, $\mathfrak{so}_{2n+1}(\mathbb{F})$, $\mathfrak{sp}_{2n}(\mathbb{F})$, $\mathfrak{so}_{2n}(\mathbb{F})$ equipped with the commutator product.

In order to discuss operations that are not bilinear, for example, evaluations of functions like $\mathbb{F}^{n\times n} \to \mathbb{F}^{n\times n}$, $X \mapsto X^2$, we require a notion similar to bilinear complexity. Let U, V be vector spaces over $\mathbb{F}$ and let $\varphi : U \to V$ be a map. The *multiplicative complexity* [5] of φ is defined to be the minimal number of multiplications and divisions over $\mathbb{F}$ required to evaluate φ and is denoted by $L(\varphi)$. We set $L(\varphi) = \infty$ if φ cannot be evaluated with just field arithmetic, i.e., additions, subtractions, multiplications, divisions. As in the case of bilinear maps, let

$$T \coloneqq \{(U_n, V_n, \varphi_n) : n \in \mathbb{N}\}$$

be a sequence of maps $\varphi_n : U_n \to V_n$, we define its *asymptotic exponent* to be

$$L(T) \coloneqq \liminf_{n\to\infty} \log_n(L(\varphi_n)). \tag{4}$$

While in general the multiplicative complexity of a bilinear map will not be the same as its bilinear complexity, we will see next that, in an appropriate sense, they are equivalent asymptotically. So there is no ambiguity in giving the same name to (2) and (4).

3 ASYMPTOTIC EXPONENTS

A blinear map $\beta : U \times V \to W$ may also be regarded as a map $\widetilde{\beta} : U \oplus V \to W$. In general, $\omega(\beta) \neq L(\widetilde{\beta})$. Nevertheless, given any sequence $S = \{(U_n, V_n, W_n, \beta_n) : n \in \mathbb{N}\}$ of bilinear maps and its corresponding sequence $\widetilde{S} = \{(U_n \oplus V_n, W_n, \widetilde{\beta}_n) : n \in \mathbb{N}\}$ of maps, the following always holds [15].

Proposition 3.1. *Let $S, \widetilde{S}$ be as above. Then $\omega(S) = L(\widetilde{S})$.*

Let $\mathbb{L}$ be a finite extension of the field $\mathbb{F}$. If S is a sequence of bilinear maps over $\mathbb{L}$, then both $\omega_{\mathbb{F}}(S)$ and $\omega_{\mathbb{L}}(S)$ are well-defined. Fortunately, they are also unambiguous.

Proposition 3.2. *If $\dim_{\mathbb{F}} \mathbb{L} < \infty$, then $\omega_{\mathbb{F}}(S) = \omega_{\mathbb{L}}(S)$.*

For a sequence $S = \{(U_n, V_n, W_n, \beta_n) : n \in \mathbb{N}\}$ of bilinear maps over $\mathbb{F}$, we define its $\mathbb{L}$-extension by

$$S_{\mathbb{L}} \coloneqq \{(U_n \otimes_{\mathbb{F}} \mathbb{L}, V_n \otimes_{\mathbb{F}} \mathbb{L}, W_n \otimes_{\mathbb{F}} \mathbb{L}, \beta_n^{\mathbb{L}}) : n \in \mathbb{N}\},$$

where the $\mathbb{L}$-bilinear map $\beta_n^{\mathbb{L}} : (U_n \otimes_{\mathbb{F}} \mathbb{L}) \times (V_n \otimes_{\mathbb{F}} \mathbb{L}) \to (W_n \otimes_{\mathbb{F}} \mathbb{L})$ is the natural extension of the $\mathbb{F}$-bilinear map $\beta_n : U_n \times V_n \to W_n$.

Proposition 3.3. *If $\dim_{\mathbb{F}} \mathbb{L} < \infty$, then $\omega_{\mathbb{L}}(S_{\mathbb{L}}) = \omega_{\mathbb{F}}(S)$.*

As a consequence, we obtain the following:

Corollary 3.4. (i) *For any sequence of bilinear maps S over $\mathbb{C}$,*

$$\omega_{\mathbb{C}}(S) = \omega_{\mathbb{R}}(S).$$

(ii) *For any sequence of bilinear maps S over $\mathbb{R}$,*

$$\omega_{\mathbb{C}}(S \otimes_{\mathbb{R}} \mathbb{C}) = \omega_{\mathbb{R}}(S).$$

By Corollary 3.4, the exponent of matrix multiplication over $\mathbb{R}$ and over $\mathbb{C}$ are the same, i.e., $\omega(\mathrm{M}(\mathbb{R})) = \omega(\mathrm{M}(\mathbb{C}))$ and we may write ω without ambiguity.

4 ASYMPTOTIC EXPONENTS OF ALGEBRAS

We will now look into the asymptotic exponents of the products in various infinite families of algebras. The bottom line is that under some very mild conditions, they are all equal to ω. In this and the next section, the field $\mathbb{F}$ will either be $\mathbb{R}$ or $\mathbb{C}$.

4.1 Lie algebras

A *Lie algebra* is a vector space $\mathfrak{g}$ over a field $\mathbb{F}$ equipped with a skew symmetric bilinear map called the *Lie bracket*,

$$[\cdot, \cdot] : \mathfrak{g} \times \mathfrak{g} \to \mathfrak{g},$$

that satisfies the Jacobi identity:

$$[x, [y, z]] + [y, [z, x]] + [z, [x, y]] = 0, \quad x, y, z \in \mathfrak{g}.$$

We say a Lie algebra is *simple* if it does not contain any nontrivial ideal and *semisimple* if it is a direct sum of simple Lie algebras. For example, the vector space $\mathfrak{so}_n(\mathbb{R})$ consisting of $n \times n$ skew-symmetric matrices, equipped with the commutator product $[A, B] = AB - BA$ is a simple Lie algebra; on the other hand, $\mathfrak{gl}_n(\mathbb{R})$, as described in Section 2, is a semisimple Lie algebra.

Theorem 4.1 (Lie algebras). *For each $n \in \mathbb{N}$, let $\mathfrak{g}_n$ be a semisimple Lie algebra over $\mathbb{F}$ with $\dim \mathfrak{g}_n \leq cn^2$ for some constant $c > 0$. Then for the sequence of Lie brackets*

$$\mathfrak{g}(\mathbb{F}) = \{(\mathfrak{g}_n, [\cdot, \cdot]) : n \in \mathbb{N}\},$$

we have

$$\omega(\mathfrak{g}(\mathbb{F})) \leq \omega,$$

with equality as long as $\mathfrak{g}_n$ is not a direct sum of exceptional Lie algebras for sufficiently large n.

It follows that the Lie brackets given by the commutator product $[A, B] = AB - BA$ on the Lie algebras $\mathfrak{gl}_n(\mathbb{R})$, $\mathfrak{sl}_{n+1}(\mathbb{F})$, $\mathfrak{so}_{2n+1}(\mathbb{F})$, $\mathfrak{sp}_{2n}(\mathbb{F})$, $\mathfrak{so}_{2n}(\mathbb{F})$ all have the same asymptotic complexity as matrix multiplication.

4.2 Jordan algebras

A *Jordan algebra* is a symmetric analogue of a Lie algebra, and is defined to be a nonassociative algebra $\mathcal{J}$ satisfying

$$xy = yx, \quad (xy)x^2 = x(yx^2), \quad x, y \in \mathcal{J}.$$

Similar to Lie algebras, a Jordan algebra is called *simple* if it does not have a nontrivial ideal and *semisimple* if it can be decomposed as a direct sum of simple Jordan algebras. Moreover, a Jordan algebra $\mathcal{J}$ is said to be *formally real* if

$$x_1^2 + \cdots + x_n^2 = 0 \quad \Longrightarrow \quad x_1 = \cdots = x_n = 0$$

for any positive integer n and $x_1, \ldots, x_n \in \mathcal{J}$. For example, the vector space $\mathrm{Sym}_n(\mathbb{R})$ (resp. $\mathrm{Herm}_n(\mathbb{C})$) of $n \times n$ symmetric (resp. Hermitian) matrices equipped with the Jordan product $A \circ B := (AB + BA)/2$ is a formally real simple Jordan algebra.

Theorem 4.2 (Jordan algebras). *For each $n \in \mathbb{N}$, let $\mathcal{J}_n$ be a Jordan algebra that is either semisimple over $\mathbb{C}$ or formally real over $\mathbb{R}$ with the following properties:*

(i) *there exists constant $c_1 > 0$ such that for sufficiently large n, $\dim_{\mathbb{R}}(\mathcal{J}_n) \le c_1 n^2$;*
(ii) *there exist constants $0 < c_2 < c_3$ such that for sufficiently large n, $\mathcal{J}_n$ contains at least one simple ideal $\mathcal{I}_n$ where $c_2 n^2 \le \dim_{\mathbb{R}}(\mathcal{I}_n) \le c_3 n^2$.*

Then for the sequence of Jordan products

$$\mathcal{J}(\mathbb{F}) = \{(\mathcal{J}_n, \circ) : n \in \mathbb{N}\},$$

we have

$$\omega(\mathcal{J}(\mathbb{F})) = \omega.$$

It follows that the Jordan product $A \circ B = (AB + BA)/2$ on $\mathrm{Sym}_n(\mathbb{R})$ or $\mathrm{Herm}_n(\mathbb{C})$ has the same asymptotic complexity as matrix multiplication.

4.3 Clifford algebras

A *Clifford algebra* $\mathrm{Cl}(V, q)$ is an algebra associated with a vector space V and a quadratic form q on V, defined by

$$\mathrm{Cl}(V, q) := \mathcal{T}(V)/\mathcal{I}_q,$$

where $\mathcal{T}(V) = \bigoplus_{n=0}^{\infty} V^{\otimes n}$ is the tensor algebra of V and $\mathcal{I}_q$ is the ideal generated by elements of the form $v \otimes v - q(v)$, $v \in V$. The most common example of Clifford algebras [2] is $\mathrm{Cliff}(n) := \mathrm{Cl}(\mathbb{R}^n, q)$ where

$$q(x) := x_1^2 + \cdots + x_n^2$$

for any $x = (x_1, \ldots, x_n) \in \mathbb{R}^n$. Then $\mathrm{Cliff}(n)$ is a real vector space of dimension 2^n, equipped with a noncommutative product $x \cdot y$ for any $x, y \in \mathbb{R}^n$ that satisfies

$$x \cdot x = \sum_{j=1}^{n} x_j^2.$$

In this case, we have

$$\mathrm{Cliff}(0) = \mathbb{R}, \quad \mathrm{Cliff}(1) = \mathbb{C}, \quad \mathrm{Cliff}(2) = \mathbb{H},$$
$$\mathrm{Cliff}(3) = \mathbb{H}^2, \quad \mathrm{Cliff}(4) = \mathbb{H}^{2\times 2}, \quad \mathrm{Cliff}(5) = \mathbb{C}^{4\times 4},$$
$$\mathrm{Cliff}(6) = \mathbb{R}^{8\times 8}, \quad \mathrm{Cliff}(7) = \mathbb{R}^{8\times 8} \oplus \mathbb{R}^{8\times 8},$$

and the recursion $\mathrm{Cliff}(n+8) = \mathrm{Cliff}(n) \otimes \mathbb{R}^{16\times 16}$ determines all subsequent $\mathrm{Cliff}(n)$, $n \ge 8$.

Theorem 4.3 (Clifford algebras). *For each $n \in \mathbb{N}$, let $\mathrm{Cl}_{2^n} := \mathrm{Cl}(V_n, q_n)$ be a Clifford algebra with the following properties:*

(i) *there exist constants $0 < c_1 < c_2$ such that for sufficiently large n, V_n is a vector space over $\mathbb{R}$ or $\mathbb{C}$ with $c_1 n \le \dim V_n \le c_2 n$;*
(ii) *q_n is a non-degenerate quadratic form on V_n.*

Then for the sequence of Clifford products

$$\mathrm{Cl}(\mathbb{F}) = \{(\mathrm{Cl}_{2^n}, \cdot) : n \in \mathbb{N}\},$$

we have

$$\omega(\mathrm{Cl}(\mathbb{F})) = \omega.$$

5 ASYMPTOTIC EXPONENT OF MATRIX FUNCTIONS

We may view the matrix product AB, Lie bracket $[A, B] = AB - BA$, Jordan product $A \circ B = (AB + BA)/2$ as bivariate polynomial functions of the matrix variables (A, B). From the previous sections, we know that they all share the same asymptotic exponent. It is natural to ask if this extends to other polynomial functions of matrices. We remind the reader that $\mathbb{F} = \mathbb{R}$ or $\mathbb{C}$ in this section. $\mathbb{F}\langle X_1, \ldots, X_k\rangle$ will denote the *free algebra* in the noncommuting variables $X_1, \ldots, X_k$ with coefficients in $\mathbb{F}$.

Let $f \in \mathbb{F}\langle X_1, \ldots, X_k\rangle$ be a polynomial in the noncommuting variables $X_1, \ldots, X_k$. For any matrices $A_1, \ldots, A_k \in \mathbb{F}^{n\times n}$, one may evaluate f to obtain a matrix $f(A_1, \ldots, A_k) \in \mathbb{F}^{n\times n}$. We denote the *evaluation map* by

$$\mathrm{ev}_n : \mathbb{F}\langle X_1, \ldots, X_k\rangle \times \underbrace{\mathbb{F}^{n\times n} \times \cdots \times \mathbb{F}^{n\times n}}_{k \text{ copies}} \to \mathbb{F}^{n\times n}.$$

In other words,

$$\mathrm{ev}_n(f, A_1, \ldots, A_k) = f(A_1, \ldots, A_k).$$

Theorem 5.1. *Let $f \in \mathbb{F}\langle X_1, \ldots, X_k\rangle$ be fixed. If*

$$\mathrm{ev}_f := \left\{\left(\mathbb{F}^{n\times n} \times \cdots \times \mathbb{F}^{n\times n}, \mathbb{F}^{n\times n}, \mathrm{ev}_n(f, \cdot, \ldots, \cdot)\right) : n \in \mathbb{N}\right\},$$

then

$$L(\mathrm{ev}_f) = \omega.$$

A noteworthy point is that Theorem 5.1 holds regardless of the degree of f and the number of variables k involved. However complicated f is, the asymptotic exponent of evaluating f on k matrices is the same regardless, namely, equal to that of multiplying two matrices.

To extend Theorem 5.1 to matrix rational functions, we first observe that for $f, g \in \mathbb{F}\langle X_1, \ldots, X_k\rangle$ and $A_1, \ldots, A_k \in \mathbb{F}^{n\times n}$, inverting the matrix $g(A_1, \ldots, A_n)$ also has asymptotic exponent ω, and thus

$$L(\mathrm{ev}_{f/g}) \le \omega, \tag{5}$$

where

$$\mathrm{ev}_n(f/g, A_1, \ldots, A_k) := f(A_1, \ldots, A_k)g(A_1, \ldots, A_k)^{-1}.$$

The reverse inequality and thus equality also holds in (5) but the argument is more intricate and is deferred to [15].

ACKNOWLEDGMENTS

We thank Dario Bini and Mateusz Michałek for very useful discussions. LHL is partially supported by National Science Foundation IIS-1546413 and DMS-1854831. KY is partially supported by National Science Foundation of China (Grant nos. 11801548 and 11688101), National Key R & D Program of China (Grant no. 2018YFA0306702), and the Recruitment Program of Global Experts of China.

REFERENCES

[1] Andris Ambainis, Yuval Filmus, and François Le Gall. 2014. Fast Matrix Multiplication: Limitations of the Laser Method. *arXiv e-prints*, Article arXiv:1411.5414 (2014).

[2] John C. Baez. 2002. The octonions. *Bulletin of the American Mathematical Society (N.S.)* 39, 2 (2002), 145–205.

[3] Grey Ballard, Christian Ikenmeyer, Joseph M. Landsberg, and Nick Ryder. 2019. The geometry of rank decompositions of matrix multiplication II: 3×3 matrices. *Journal of Pure and Applied Algebra* 223, 8 (2019), 3205–3224.

[4] Dario Bini. 1980. Relations between exact and approximate bilinear algorithms. Applications. *Calcolo* 17, 1 (1980), 87–97.

[5] Peter Bürgisser, Michael Clausen, and M. Amin Shokrollahi. 1997. *Algebraic complexity theory*. Grundlehren der Mathematischen Wissenschaften, Vol. 315. Springer-Verlag, Berlin.

[6] Luca Chiantini, Jonathan D. Hauenstein, Christian Ikenmeyer, Joseph M. Landsberg, and Giorgio Ottaviani. 2018. Polynomials and the exponent of matrix multiplication. *Bulletin of the London Mathematical Society* 50, 3 (2018), 369–389.

[7] Luca Chiantini, Christian Ikenmeyer, Joseph M. Landsberg, and Giorgio Ottaviani. 2019. The Geometry of Rank Decompositions of Matrix Multiplication I: 2×2 Matrices. *Experimental Mathematics* 28, 3 (2019), 322–327.

[8] Henry Cohn and Christopher Umans. 2003. A group-theoretic approach to fast matrix multiplication. *44th Annual IEEE Symposium on Foundations of Computer Science, 2003. Proceedings.* (2003), 438–449.

[9] Henry Cohn and Christopher Umans. 2013. Fast matrix multiplication using coherent configurations. *Proceedings of the Twenty-Fourth Annual ACM-SIAM Symposium on Discrete Algorithms* (2013), 1074–1087.

[10] Don Coppersmith and Shmuel Winograd. 1987. Matrix Multiplication via Arithmetic Progressions. In *Proceedings of the 19th Annual ACM Symposium on Theory of Computing, 1987, New York, New York, USA*.

[11] Joseph M. Landsberg. 2006. The border rank of the multiplication of 2×2 matrices is seven. *Journal of the American Mathematical Society* 19, 8 (2006), 447–459.

[12] Joseph M. Landsberg. 2014. New Lower Bounds for the Rank of Matrix Multiplication. *SIAM J. Comput.* 43, 1 (2014), 144–149.

[13] Joseph M. Landsberg and Giorgio Ottaviani. 2015. New Lower Bounds for the Border Rank of Matrix Multiplication. *Theory of Computing* 11, 11 (2015), 285–298.

[14] François Le Gall. 2014. Powers of Tensors and Fast Matrix Multiplication. In *Proceedings of the 39th International Symposium on Symbolic and Algebraic Computation* (Kobe, Japan) *(ISSAC '14)*. Association for Computing Machinery, New York, NY, USA, 296—-303.

[15] Lek-Heng Lim and Ke Ye. 2020. Ubiquity of the Exponent of Matrix Multiplication. *preprint* (2020).

[16] Arnold Schönhage. 1981. Partial and Total Matrix Multiplication. *SIAM J. Comput.* 10, 3 (1981), 434–455.

[17] Volker Strassen. 1969. Gaussian elimination is not optimal. *Numerische mathematik* 13, 4 (1969), 354–356.

[18] Christopher Umans. 2006. Group-Theoretic Algorithms for Matrix Multiplication. In *Proceedings of the 2006 International Symposium on Symbolic and Algebraic Computation* (Genoa, Italy) *(ISSAC '06)*. Association for Computing Machinery, New York, NY, USA, 5.

[19] Virginia Vassilevska Williams. 2012. Multiplying Matrices Faster than Coppersmith-Winograd. In *Proceedings of the Forty-Fourth Annual ACM Symposium on Theory of Computing* (New York, New York, USA) *(STOC '12)*. Association for Computing Machinery, New York, NY, USA, 887—-898.

[20] Ke Ye and Lek-Heng Lim. 2018. Fast Structured Matrix Computations: Tensor Rank and Cohn–Umans Method. *Foundations of Computational Mathematics* 18, 1 (2018), 45–95.

What do Sparse Interpolation, Padé Approximation, Gaussian Quadrature and Tensor Decomposition Have in Common?

Annie Cuyt
annie.cuyt@uantwerpen.be
Department of Computer Science, University of Antwerp, Campus CMI, Middelheimlaan 1, B-2020 Antwerpen, Belgium
College of Mathematics and Statistics, Shenzhen University, Shenzhen, Guangdong 518060, China

KEYWORDS

exponential analysis, Prony's method, generalized eigenvalue, Hankel matrix

ACM Reference Format:
Annie Cuyt. 2020. What do Sparse Interpolation, Padé Approximation, Gaussian Quadrature and Tensor Decomposition Have in Common?. In *International Symposium on Symbolic and Algebraic Computation (ISSAC '20), July 20–23, 2020, Kalamata, Greece.* ACM, New York, NY, USA, 1 page. https://doi.org/10.1145/3373207.3403983

We present the problem statement of sparse interpolation of data f_i collected uniformly at points $x_i = i\Delta$, as

$$\sum_{j=1}^{n} \alpha_j \exp(\phi_j x_i) = f_i, \quad \alpha_j, \phi_j \in \mathbb{C}, \quad |\Im(\phi_j)\Delta| < \pi, \qquad (1)$$

and its basic mathematical and computational methods to solve it.

The original solution is presented already in 1795 by de Prony [6]. Much later it is expressed in terms of the generalized eigenvalue problem [8]

$$H^{(1)}_{n,n} v_j = \exp(\phi_j\Delta) H^{(0)}_{n,n} v_j, \quad j = 1, \ldots, n,$$
$$H^{(r)}_{n_1,n_2} = \begin{pmatrix} f_r & \cdots & f_{r+n_2-1} \\ \vdots & \iddots & \vdots \\ f_{r+n_1-1} & \cdots & f_{r+n_1+n_2-2} \end{pmatrix}$$

for the ϕ_j and the subsequent solution of the structured linear system (1) for the α_j.

When considering the limited number of regularly collected samples f_i as Taylor series coefficients,

$$\sum_{i=0}^{\infty} f_i z^i = \sum_{j=1}^{n} \frac{\alpha_j}{1 - \exp(\phi_j\Delta) z},$$

the problem statement easily connects to Padé approximation [4, 9].

The Padé approximant denominators are in turn closely related to the formally orthogonal Hadamard polynomials and Gaussian quadrature [1, 7] by

$$\prod_{j=1}^{n} (z - \exp(\phi_j\Delta)) = \begin{vmatrix} f_0 & \cdots & f_n \\ \vdots & & \vdots \\ f_{n-1} & \cdots & f_{2n-1} \\ 1 & \cdots & z^n \end{vmatrix} / |H^{(0)}_{n,n}|.$$

Results on their zeroes and certain convergence properties shed new light [2] on some computational problems in sparse interpolation.

The problem statement can also be viewed as an m-order tensor decomposition problem where $3 \le m \le 2n-1$ [3, 5], namely

$$\sum_{j=1}^{n} \alpha_j \begin{pmatrix} 1 \\ \exp(\phi_j\Delta) \\ \vdots \\ \exp(\phi_j\Delta)^{n_1-1} \end{pmatrix} \circ \cdots \circ \begin{pmatrix} 1 \\ \exp(\phi_j\Delta) \\ \vdots \\ \exp(\phi_j\Delta)^{n_m-1} \end{pmatrix}$$
$$= (f_{k_1+\ldots+k_m-m})_{1 \le k_\ell \le n_\ell}, \quad 1 \le \ell \le m, 2 \le n_\ell \le n,$$

with the connection

$$H^{(k_3+\ldots+k_m-m+2)}_{n_1,n_2} = (f_{i+j+k_3+\ldots+k_m-m})_{1 \le i \le n_1, 1 \le j \le n_2}.$$

Here $\circ$ denotes the outer product and the decomposition problem is solved using techniques from multilinear algebra. Through the latter reformulation the toolkit of algorithms for sparse interpolation is further enlarged.

REFERENCES

[1] C. Brezinski. 1980. *Padé type approximation and general orthogonal polynomials.* ISNM 50, Birkhäuser Verlag, Basel.
[2] M. Briani, A. Cuyt, and W.-s. Lee. 2017. *VEXPA: Validated EXPonential Analysis through regular subsampling.* ArXiv e-print 1709.04281 [math.NA]. Universiteit Antwerpen.
[3] A. Cuyt, F. Knaepkens, and W.-s. Lee. 2018. From exponential analysis to Padé approximation and Tensor decomposition, in one and more dimensions. In *LNCS11077*, V.P. Gerdt et al. (Eds.). 116–130. Proceedings CASC 2018, Lille (France).
[4] A. Cuyt and W.-s. Lee. 2016. Sparse interpolation and Rational approximation *(Contemporary Mathematics)*, D. Hardin, D. Lubinsky, and B. Simanek (Eds.), Vol. 661. American Mathematical Society, Providence, RI, 229–242. https://doi.org/10.1090/conm/661/13284
[5] A. Cuyt, W.-s. Lee, and X. Yang. 2016. On tensor decomposition, sparse interpolation and Padé approximation. *Jaén J. Approx.* 8, 1 (2016), 33–58.
[6] R. de Prony. 1795. Essai expérimental et analytique sur les lois de la dilatabilité des fluides élastiques et sur celles de la force expansive de la vapeur de l'eau et de la vapeur de l'alkool, à différentes températures. *J. Ec. Poly.* 1 (1795), 24–76.
[7] P. Henrici. 1974. *Applied and computational complex analysis I.* John Wiley & Sons, New York.
[8] Y. Hua and T. K. Sarkar. 1990. Matrix pencil method for estimating parameters of exponentially damped/undamped sinusoids in noise. *IEEE Trans. Acoust., Speech, Signal Process.* 38 (1990), 814–824. https://doi.org/10.1109/29.56027
[9] L. Weiss and R. N. McDonough. 1963. Prony's method, Z-transforms, and Padé approximation. *SIAM Rev.* 5 (1963), 145–149.

ISSAC '20, July 20–23, 2020, Kalamata, Greece

ACM ISBN 978-1-4503-7100-1/20/07.
https://doi.org/10.1145/3373207.3403983

Real Quantifier Elimination by Cylindrical Algebraic Decomposition, and Improvements by Machine Learning

Matthew England
Matthew.England@coventry.ac.uk
Coventry University
Coventry, UK

CCS CONCEPTS

• **Computing methodologies → Equation and inequality solving algorithms**; **Machine learning**.

KEYWORDS

quantifier elimination, cylindrical algebraic decomposition, software optimisation by machine learning

ACM Reference Format:
Matthew England. 2020. Real Quantifier Elimination by Cylindrical Algebraic Decomposition, and Improvements by Machine Learning. In *International Symposium on Symbolic and Algebraic Computation (ISSAC '20), July 20–23, 2020, Kalamata, Greece.* ACM, New York, NY, USA, 1 page. https://doi.org/10.1145/3373207.3403981

Real QE by CAD

Given a quantified logical formula whose atoms are polynomial constraints with real valued variables, Real Quantifier Elimination (QE) means to derive a logically equivalent formula which does not involve quantifiers or the quantified variables from the original statement. For example, Real QE would reduce the statement that there exists a real solution x to the quadratic equation $x^2+bx+c = 0$ to the equivalent condition on the discriminant: $b^2 - 4c \geq 0$. Tarski proved Real QE is always possible (with sufficient resources) [7].

A Cylindrical Algebraic Decomposition (CAD) decomposes n-dimension real space into a finite number of semi-algebraic cells, relative to a set of polynomials (or formulae) so that each has constant sign (truth value) on each cell. A CAD for the polynomials in a formula may be used to perform Real QE, by querying a single sample point from each cell and combining cell descriptions into the quantifier free formula [1]. CAD is the backbone of Real QE systems, used in the cases where more efficient algorithms are not applicable. CAD is implemented in a variety of computer algebra systems, as well as dedicated standalone implementations and satisfiability modulo theory solvers. Real QE has been applied to numerous problems throughout engineering, the sciences and even economics.

ISSAC '20, July 20–23, 2020, Kalamata, Greece

ACM ISBN 978-1-4503-7100-1/20/07.
https://doi.org/10.1145/3373207.3403981

Improvements by Machine Learning

Real QE has doubly exponential complexity which limits the scope of its use in practice. It is thus particularly important that implementations are optimised to ensure the best possible performance.

In particular, there are a variety of choices which may need to be made that can dramatically affect the runtime of such algorithms without changing the mathematical correctness of the result produced. These include the variable ordering, the order we process constraints in, and the order we process sub-formulae in.

Rather than having such choices made by the user, or a human-constructed heuristic, we suggest to use Machine Learning (ML), i.e. tools that allow computers to make decisions that are not explicitly programmed, via the analysis of large quantities of data. This is not a simple application of ML since there are few a-priori bounds on the input size, and the only way to know if a choice is good is to evaluate all possibilities and compare. Nevertheless, we have experimented with different classifiers, training methodologies, and feature extract methods to produce classifiers which make much better choices for the variable ordering in our CAD implementation.

In this tutorial we first introduce Real QE via CAD, before reviewing recent work using machine learning to improve the performance of a CAD implementation [2–6]. We hope that the former will be an informative primer, and that the latter has useful lessons for those looking to apply ML to other areas of symbolic computation.

REFERENCES

[1] G.E. Collins. 1975. Quantifier elimination for real closed fields by cylindrical algebraic decomposition. In *Proc. 2nd GI Conference on Automata Theory and Formal Languages.* Springer-Verlag, 134–183. https://doi.org/10.1007/3-540-07407-4_17
[2] M. England and D. Florescu. 2019. Comparing Machine Learning Models to Choose the Variable Ordering for Cylindrical Algebraic Decomposition. In *Intelligent Computer Mathematics (LNCS 11617)*, C. Kaliszyk, E. Brady, A. Kohlhase, and C.C. Sacerdoti (Eds.). Springer, 93–108. https://doi.org/10.1007/978-3-030-23250-4_7
[3] D. Florescu and M. England. 2019. Algorithmically generating new algebraic features of polynomial systems for machine learning. In *Proc. 4th Workshop on Satisfiability Checking and Symbolic Computation (*SC^2 *2019) (CEUR Workshop Proceedings)*, J. Abbott and A. Griggio (Eds.). 12. http://ceur-ws.org/Vol-2460/
[4] D. Florescu and M. England. 2020. Improved Cross-Validation for Classifiers that Make Algorithmic Choices to Minimise Runtime Without Compromising Output Correctness. In *Mathematical Aspects of Computer and Information Sciences (Proc. MACIS '19) (LNCS 11989)*, D. Slamanig, E. Tsigaridas, and Z. Zafeirakopoulos (Eds.). Springer, 341–356. https://doi.org/10.1007/978-3-030-43120-4_27
[5] D. Florescu and M. England. 2020. A machine learning based software pipeline to pick the variable ordering for algorithms with polynomial inputs. In *Mathematical Software (Proc. ICMS '20).* In Press: Springer LNCS.
[6] Z. Huang, M. England, D. Wilson, J. Bridge, J.H. Davenport, and L. Paulson. 2019. Using Machine Learning to Improve Cylindrical Algebraic Decomposition. *Mathematics in Computer Science* 13, 4 (2019), 461–488. https://doi.org/10.1007/s11786-019-00394-8
[7] A. Tarski. 1948. *A Decision Method For Elementary Algebra And Geometry.* RAND Corporation, Santa Monica, CA (reprinted in: https://doi.org/10.1007/978-3-7091-9459-1).

Sub-quadratic Time for Riemann–Roch Spaces

Case of smooth divisors over nodal plane projective curves

Simon Abelard
Laboratoire d'informatique de l'École polytechnique (LIX, UMR 7161), CNRS, École polytechnique, Institut Polytechnique de Paris
Palaiseau, France
simon.abelard@lix.polytechnique.fr

Alain Couvreur
Inria
Laboratoire d'informatique de l'École polytechnique (LIX, UMR 7161), CNRS, École polytechnique, Institut Polytechnique de Paris
Palaiseau, France
alain.couvreur@inria.fr

Grégoire Lecerf
Laboratoire d'informatique de l'École polytechnique (LIX, UMR 7161), CNRS, École polytechnique, Institut Polytechnique de Paris
Palaiseau, France
gregoire.lecerf@lix.polytechnique.fr

ABSTRACT

We revisit the seminal Brill–Noether algorithm in the rather generic situation of smooth divisors over a nodal plane projective curve. Our approach takes advantage of fast algorithms for polynomials and structured matrices. We reach sub-quadratic time for computing a basis of a Riemann–Roch space. This improves upon previously known complexity bounds.

CCS CONCEPTS

• **Computing methodologies** → **Symbolic and algebraic algorithms**.

KEYWORDS

algebraic curves, Riemann–Roch spaces, complexity

ACM Reference Format:
Simon Abelard, Alain Couvreur, and Grégoire Lecerf. 2020. Sub-quadratic Time for Riemann–Roch Spaces: Case of smooth divisors over nodal plane projective curves. In *International Symposium on Symbolic and Algebraic Computation (ISSAC '20), July 20–23, 2020, Kalamata, Greece.* ACM, New York, NY, USA, 8 pages. https://doi.org/10.1145/3373207.3404053

1 INTRODUCTION

Let $\mathbb{K}$ be an effective field and let $\bar{\mathbb{K}}$ denote an algebraic closure of $\mathbb{K}$. Here "effective" means that we can perform arithmetic operations and zero tests in $\mathbb{K}$. The projective space of dimension 2 over $\bar{\mathbb{K}}$ is written $\mathbb{P}^2$. The input projective curve C in $\mathbb{P}^2$ is given by its defining equation $Q(X, Y, Z) = 0$, where $Q \in \mathbb{K}[X, Y, Z]$ is homogeneous of total degree $\delta \geqslant 1$. This paper modifies the variant of the Brill–Noether algorithm proposed in [19] so as to reach sharper complexity bounds.

1.1 Hypotheses

Until the end of the paper, $\mathbb{K}$ is a sufficiently large field with the following restriction:
$\mathbb{K}$-**H** $\mathbb{K}$ is either finite or has characteristic zero, and is therefore a perfect field.

We will assume that the following hypotheses hold for C:
C-$\mathbf{H_1}$ Q is absolutely irreducible, that is irreducible over $\bar{\mathbb{K}}$;
C-$\mathbf{H_2}$ C is nodal: each germ of curve at a singular point splits into two smooth germs with distinct tangent spaces. The number of singular points is written r, and the *nodal divisor*, written E is the symbolic sum of the singular points.

Let us recall that absolute irreducibility can be tested efficiently by means of the algorithms designed in [1]. For the second hypothesis it suffices to check that the Hessian of Q is non-degenerate at each singular point. The restriction on the type of singularities involves simplifications in the Brill–Noether algorithm [7, 17]: basically the desingularization of C is immediate, and the adjoint divisor simply writes from the singular locus. The last hypothesis necessary to our algorithm concerns the input divisor D, for which we want a basis of the Riemann–Roch space, written $\mathcal{L}(D)$:
D-**H** The input divisor D is *smooth* and defined over $\mathbb{K}$, which means that its support is made of regular points of C.

We will decompose a divisor D into $D = D_+ - D_-$, where D_+ and D_- are *positive* (also called *effective*) divisors. When $\deg D_+ < \deg D_-$, $\mathcal{L}(D)$ is (0) so we can freely assume that $\deg D_+ \geqslant \deg D_-$. The above hypotheses are essentially present in [19]: $\mathbb{K}$-H is slightly more restrictive in order to simplify complexity analyses.

1.2 Notation

For complexity analyses we focus on an algebraic model over $\mathbb{K}$ (typically computation trees), so we count the number of arithmetic operations and zero tests performed by the algorithms. Over finite fields, we use Turing machines with sufficiently many but finite number of tapes. In order to simplify the presentation of complexity bounds, we use the *soft-Oh* notation: $f(n) \in \tilde{O}(g(n))$ means that $f(n) = g(n)\log_2^{O(1)}(g(n)+3)$; see [5, chapter 25, section 7]. The vector space of polynomials of degree $< n$ in $\mathbb{K}[X]$ is written $\mathbb{K}[X]_{<n}$. For integer and polynomial arithmetic we content ourselves with softly linear cost bounds [5].

The constant ω denotes a real value between 2 and 3 such that two $n \times n$ matrices over a commutative ring can be multiplied with $O(n^\omega)$ operations; $\omega < 2.3728639$ [18]. The constant ϖ is an other real value between 1.5 and $(\omega+1)/2$ such that the product of a $n \times \sqrt{n}$ matrix by a $\sqrt{n} \times \sqrt{n}$ matrix takes $O(n^\varpi)$ operations; $\varpi < 1.667$ [15, Theorem 10.1].

Given $M \in \mathrm{GL}_3(\mathbb{K})$ and $P \in \mathbb{K}[X, Y, Z]$ we denote by $(P \circ M)(X, Y, Z)$ the polynomial $P(M \cdot (X, Y, Z)^\top)$.

ISSAC '20, July 20–23, 2020, Kalamata, Greece

ACM ISBN 978-1-4503-7100-1/20/07…$15.00
https://doi.org/10.1145/3373207.3404053

1.3 Our contributions

The present paper is essentially based on the variant of the Brill–Noether algorithm designed in [19]. Our first result is the improvement of complexity bounds for the arithmetic of smooth divisors. A second contribution concerns the opportune use of structured linear algebra algorithms: we reformulate the Riemann–Roch problem in terms of modules of relations of rank $\leqslant \delta$, and compute bases thanks to the recent fast algorithm due to Neiger [23].

We represent the Riemann–Roch space

$$\mathcal{L}(D) := \{h \in \mathbb{K}(C) : (h) \geq -D\} \cup \{0\}$$

by a *basis generator*, that is made of $M \in \mathrm{GL}_3(\mathbb{K})$, an integer $l \leqslant \delta$, non-zero homogeneous polynomials $H, G_1, \ldots, G_l$ in $\mathbb{K}[X, Y, Z]$ of respective total degrees d and $d_i \leqslant d$, such that:

- $\deg_Y(Q \circ M) = \delta, \deg_Y H \leqslant \delta - 1, \deg_Y G_i \leqslant \delta - 1$ for $i = 1, \ldots, l$.
- The supports of $M^{-1}(D_+)$, $M^{-1}(D_-)$, $M^{-1}(E)$, and the solutions of $Q \circ M = H = 0$ are in the affine chart $Z = 1$.
- $\left(\frac{X^j Z^{d-d_i-j} G_i}{H}\right) \circ M^{-1}$ with $0 \leqslant j \leqslant d - d_i$ and $1 \leqslant i \leqslant l$ form a basis of $\mathcal{L}(D)$.

The actual vector-space basis of $\mathcal{L}(D)$ can be recovered in softly linear time from the basis generators, the latter being a more compact representation.

Theorem 1.1. *Under hypotheses $\mathbb{K}$-H, C-H_1, C-H_2, with d defined below in* (8)*, given a primitive element representation (see section 3.1) of D satisfying D-H, and assuming $|\mathbb{K}| \geqslant \max(\delta^4, 6(\delta d)^2)$, a basis generator of $\mathcal{L}(D)$ can be computed by a probabilistic algorithm of type Las Vegas with an expected number of $\tilde{O}\left((\delta^2 + \deg D_+)^{\frac{\omega+1}{2}}\right)$ field operations in characteristic zero or $> \max(\delta(\delta-1), \delta d)$, or $\tilde{O}\left((\delta^2 + \deg D_+)^{\frac{\omega+1}{2}} \log q + (\delta^2 + \deg D_+)\log^2 q\right)$ bit operations if $\mathbb{K} = \mathbb{F}_q$.*

Our third contribution, central to this theorem, is a sharp degree bound d for H and the G_i; namely (8). Such a bound is not supplied in [19] when $r > 0$, so when C is not smooth additional assumptions are required in [19, section 2].

1.4 Related work

Riemann–Roch spaces have various applications in applied algebra, number theory and cryptography (*e.g.* arithmetic in Jacobians of curves). Computing bases for these spaces is also pivotal to design geometric codes, where the encoding algorithm consists in evaluating a basis of a certain Riemann–Roch space at points of an algebraic curve. Currently in practice, algebraic curves used in coding theory are mostly limited to cases for which such bases are already known, so for the sake of diversity we aim to handle more general curves and divisors.

Algorithms and implementations for Riemann–Roch spaces have been thoroughly investigated over the past decades. To focus on the most recent contributions, we mention: Hess' algorithm [9] that is implemented within the computer algebra software Magma, and Khuri–Makdisi's approach [16] that is dedicated to group operations in Jacobians of genus-g curves in time $O(g^{\omega+\epsilon})$, where ϵ can be any positive number. More recently, Le Gluher and Spaenlehauer [19] revisited the Brill–Noether approach for smooth divisors D on a nodal curve, and obtained the complexity bound $O(\max(\delta^2, \deg D_+)^\omega)$, yet under the aforementioned restriction on D. For conciseness we refer to [19] for further references.

2 PRELIMINARIES

In order to obtain the aforementioned complexity bound for Riemann–Roch spaces, we rely on structured linear algebra algorithms that will be presented in section 4.1, and on modular composition and elimination, that are the purposes of this section.

2.1 Bivariate modular composition

At present time no algorithm with softly linear time is known for bivariate modular compositions over a general field $\mathbb{K}$. For practical purposes we appeal to a variant of the Paterson–Stockmeyer evaluation scheme designed by Nüsken and Ziegler [25]. We need a slight extension to express the complexity bound in terms of the degree of the modulus.

Algorithm 1
Input: $P \in \mathbb{K}[X, Y]$ of total degree n, $\chi \in \mathbb{K}[Y]$, $u \in \mathbb{K}[Y]_{<\deg \chi}$.
Output: $P(u(Y), Y) \bmod \chi(Y)$.

(1) Let $p := \lfloor \sqrt{n} \rfloor$ and $q := \lceil n/p \rceil$.
(2) For $i = 0, \ldots, p-1$ do:
   (a) Compute $u^i \bmod \chi$ and segment it into $M_{0,i}(Y) + M_{1,i}(Y)Y^n + \cdots + M_{l-1,i}(Y)Y^{(l-1)n}$, with $\deg M_{j,i}(Y) < n$ and $l := \lceil \deg \chi / n \rceil$. This yields an $l \times p$ matrix $M \in \mathbb{K}[Y]^{l \times p}$.
   (b) For $j = 0, \ldots, q-1$, let $N_{i,j}(Y) := P^X_{i+pj}(Y)$, where P^X_{i+pj} represents the coefficient of degree $i + pj$ of P regarded in $\mathbb{K}[Y][X]$. This yields a $p \times q$ matrix $N \in \mathbb{K}[Y]^{p \times q}$ of degree $\leqslant n$.
(3) Compute the matrix product $R := MN$.
(4) For $j = 0, \ldots, q-1$, let $v_j(Y) := R_{0,j}(Y) + R_{1,j}(Y)Y^n + \cdots + R_{l-1,j}(Y)Y^{(l-1)n} \bmod \chi(Y)$.
(5) Return $\sum_{j=0}^{q-1} v_j u^{pj} \bmod \chi$.

Lemma 2.1. *Algorithm 1 is correct and takes $\tilde{O}\left(n^{\frac{\omega-1}{2}}\left(\deg \chi + n^{\frac{3}{2}}\right)\right)$ operations in $\mathbb{K}$.*

Proof. By construction, we have $v_j = P^X_{pj} + P^X_{1+pj} u + \cdots + P^X_{p-1+pj} u^{p-1}$ for $j = 0, \ldots, q-1$, whence

$$P(u(Y), Y) = \sum_{j=0}^{q-1} v_j u^{pj} \bmod \chi(Y).$$

This proves the correctness of the algorithm. Step 2a requires $O(p) = O(\sqrt{n})$ multiplications modulo χ. Step 3 costs

$$\tilde{O}\left(n^{\frac{\omega}{2}+1}\left\lceil \frac{l}{p} \right\rceil\right) = \tilde{O}\left(n^{\frac{\omega}{2}+1}\left(\frac{\frac{\deg \chi}{n} + 1}{n^{\frac{1}{2}}} + 1\right)\right) = \tilde{O}\left(n^{\frac{\omega-1}{2}}\left(\deg \chi + n^{\frac{3}{2}}\right)\right).$$

Step 5 involves $O(q) = O(\sqrt{n})$ multiplications and additions modulo χ, using Horner's method. □

2.2 Primitive element representation

A *primitive element representation* of a set $\mathcal{E}$ of points in the affine plane $\mathbb{A}^2$ is the data of:

- (λ, μ) in $\bar{\mathbb{K}}^2$ such that the linear form $\lambda X + \mu Y$ *separates* the points in $\mathcal{E}$. This means that the form takes different values at different points of $\mathcal{E}$.
- A polynomial θ in $\bar{\mathbb{K}}[S]$ whose roots are the values of $\lambda X + \mu Y$ at the points of $\mathcal{E}$, that is $\theta(S) := \prod_{(x,y)\in\mathcal{E}}(S - (\lambda x + \mu y))$. So θ is monic and separable of degree $|\mathcal{E}|$.

- Polynomials u and v in $\bar{\mathbb{K}}[S]$ of degrees $< |\mathcal{E}|$ such that

$$\mathcal{E} = \{(u(\zeta), v(\zeta)) : \theta(\zeta) = 0\}.$$

Notice that such a representation is uniquely determined by (λ, μ). If $(\lambda, \mu) \in \mathbb{K}^2$ and if $\theta, u, v \in \mathbb{K}[S]$, then the primitive element representation is said to be *defined over* $\mathbb{K}$.

If the annihilator ideal of $\mathcal{E}$ is generated by polynomials with coefficients in $\mathbb{K}$, then a primitive element representation does not necessarily exist over $\mathbb{K}$. However any value of λ/μ outside

$$\left\{ \frac{y_1 - y_2}{x_1 - x_2} : (x_1, y_1) \in \mathcal{E}, (x_2, y_2) \in \mathcal{E}, (x_1, y_1) \neq (x_2, y_2), x_1 \neq x_2 \right\}, \tag{1}$$

yields a primitive element. Therefore, a sufficient condition to ensure that such a primitive element exists is $|\mathbb{K}| > \binom{|\mathcal{E}|}{2}$. Otherwise, λ/μ needs to be taken in an algebraic extension of $\mathbb{K}$. We will not discuss these usual technical details but will make precise the conditions on the cardinality of $\mathbb{K}$ within each sub-algorithm. For the sake of complexity it will be convenient to change the variables X and Y linearly, so we recall the following lemma.

Lemma 2.2. (For instance [12, Proposition 9]) *If $F \in \mathbb{K}[X, Y, Z]$ is homogeneous of degree n, if $|\mathbb{K}| \geqslant n + 1$, and if $M \in \mathrm{GL}_3(\mathbb{K})$, then we can compute $F \circ M$ with $\tilde{O}(n^2)$ operations in $\mathbb{K}$.*

2.3 Change of primitive element

Changing primitive elements mostly reduces to computing characteristic polynomials in $\mathbb{K}[S]/(\theta(S))$. This task has received a lot of attention in computer algebra, but so far no general algorithm is known with nearly linear time; for instance see [6] about the existing literature. For our present purposes it seems reasonable to appeal to the known complexity exponent ω: recall that univariate modular composition in degree n takes $O(n^\omega)$ field operations.

Lemma 2.3. *Given a primitive element representation of $\mathcal{E}$ over $\mathbb{K}$ by $\lambda X + \mu Y$, and given $(\tilde{\lambda}, \tilde{\mu}) \in \mathbb{K}^2$, we can test if $\tilde{\lambda} X + \tilde{\mu} Y$ is primitive for $\mathcal{E}$, and, if so, compute the corresponding representation of $\mathcal{E}$, along with $w(S) \in \mathbb{K}[S]_{<|\mathcal{E}|}$ such that*

$$\begin{array}{rcl} \mathbb{K}[S]/(\theta(S)) & \cong & \mathbb{K}[S]/(\tilde{\theta}(S)) \\ S & \mapsto & w(S) \\ \tilde{\lambda} u(S) + \tilde{\mu} v(S) & \leftarrow\!\shortmid & S, \end{array}$$

is an isomorphism, with $O(|\mathcal{E}|^\omega)$ field operations in characteristic zero or $> |\mathcal{E}|$, or $|\mathcal{E}|^\omega \tilde{O}(\log q) + \tilde{O}(|\mathcal{E}| \log^2 q)$ bit operations if $\mathbb{K} = \mathbb{F}_q$.

Proof. Let Tr denote the trace map of $\mathbb{K}[S]/(\theta(S))$ and let $\tilde{\theta}$ be the characteristic polynomial of the multiplication endomorphism by $\tilde{\lambda} u(S) + \tilde{\mu} v(S)$ in this algebra. Le Verrier's method consists in computing $\mathrm{Tr}((\tilde{\lambda} u + \tilde{\mu} v)^i)$ for $i = 1, \ldots, |\mathcal{E}| - 1$. This task being dual to modular composition, it takes $O(|\mathcal{E}|^\omega)$ operations in $\mathbb{K}$. Then the generating series $\tau(z) := \sum_{i \geqslant 0} \mathrm{Tr}((\tilde{\lambda} u + \tilde{\mu} v)^i)$ satisfies the Newton–Girard formula

$$-\frac{v'(z)}{v(z)} = \tau(z) + O(z^{|\mathcal{E}|}), \tag{2}$$

where $v(z) := z^{|\mathcal{E}|} \tilde{\theta}(1/z)$ is the reciprocal of $\tilde{\theta}$. Therefore v is recovered with $\tilde{O}(|\mathcal{E}|)$ operations in characteristic zero or $> |\mathcal{E}|$. Testing if $\tilde{\lambda} X + \tilde{\mu} Y$ is primitive is equivalent to testing if $\tilde{\theta}$ is squarefree, that takes $\tilde{O}(|\mathcal{E}|)$ field operations in characteristic zero or $> |\mathcal{E}|$, or $\tilde{O}(|\mathcal{E}| \log^2 q)$ bit operations if $\mathbb{K} = \mathbb{F}_q$; for instance see [20]. By a deformation argument we further recover w up to a constant cost factor; see [6, section 2.6].

In positive characteristic, the integration of (2) is more tedious in general, but in the special case $\mathbb{K} = \mathbb{F}_q$, it is possible with $\tilde{O}(|\mathcal{E}| \log q)$ bit operations; see [6, Proposition 3]. □

2.4 Curve intersection

For computing principal divisors on curves we will appeal to the following lemma, based on polynomial resultants. The technique is known so the proof is voluntarily concise. Details can be found in [3, section 4] or [12, section 5].

Lemma 2.4. *Given F of total degree m and G of total degree $\leqslant n$ in $\mathbb{K}[X, Y]$ such that $m \leqslant n$, F has degree m in Y, and the solutions of $F = G = 0$ is a finite set $\mathcal{E}$. We can check if X is primitive for $\mathcal{E}$, and compute a partition of $\mathcal{E} =: \mathcal{E}_1 \cup \cdots \cup \mathcal{E}_s$, where $\mathcal{E}_i$ contains points with the same known intersection multiplicity m_i, with $\tilde{O}(nm^2 + n \deg_Y G)$ field operations in characteristic zero or $> mn$, or $\tilde{O}(((mn)^\omega + n \deg_Y G) \log q + mn \log^2 q)$ bit operations if $\mathbb{K} = \mathbb{F}_q$.*

Proof. The remainder H of G by F regarded in $\mathbb{K}[X][Y]$ can be computed with $\tilde{O}(n \deg_Y G)$ operations in $\mathbb{K}$, so we obtain $\chi(X) := \mathrm{Res}_Y(F(X, Y), H(X, Y))$ with cost $\tilde{O}(nm^2)$ by [21, Corollary 31]. Since χ has degree $\leqslant mn$, the squarefree decomposition $\chi =: \theta_1^{m_1} \cdots \theta_s^{m_s}$ contributes to $\tilde{O}(mn)$ field operations in characteristic zero or $> mn$, or to $\tilde{O}(mn \log^2 q)$ bit operations over $\mathbb{F}_q$.

After fast multi-remaindering of F and H by $\theta_1, \ldots, \theta_s$ [5, chapter 10], the directed evaluation paradigm [2, 11] yields a decomposition $\theta_i =: \theta_{i,1} \cdots \theta_{i,s_i}$, and bivariate polynomials $Q_{i,j}(X, Y)$ such that

$$Q_{i,j}(\zeta, Y) = \gcd(F(\zeta, Y), H(\zeta, Y)) \tag{3}$$

with $\deg_X Q_{i,j} < \deg \theta_{i,j}$, for all $\theta_{i,j}(\zeta) = 0$, $j = 1, \ldots, s_i$, $i = 1, \ldots, s$. It turns out that X is a primitive element of $\mathcal{E}$ if, and only if, each $Q_{i,j}(\zeta, Y)$ is a power of a degree 1 polynomial $Y - v_{i,j}(\zeta)$ with $\deg v_{i,j} < \deg \theta_{i,j}$. In this case, the representation $\theta_i(X) = Y - v_j(X) = 0$ of $\mathcal{E}_i$ is deduced in softly linear time by Chinese remaindering; details will be given in Lemma 3.5 below for a slightly more general situation. This takes $\tilde{O}(nm^2)$ operations in $\mathbb{K}$ when the characteristic p is zero or $> mn$. Otherwise when $p > 0$, the gcd (3) is required to be a power (coprime to p) of $Y^{p^{t_{i,j}}} - w_{i,j}(\zeta)$. We compute $A := X^{p^{t_{i,j}}} \bmod \theta_{i,j}(X)$ with bit cost $\tilde{O}(\deg \theta_{i,j} \log m \log q)$ since $p^{t_{i,j}} = O(\log m)$. Computing the characteristic polynomial $\tilde{\theta}_{i,j}$ of A and the expression $X = B(A)$ as in the proof of Lemma 2.3 involves bit cost $\tilde{O}((\deg \theta_{i,j})^\omega \log q + \deg \theta_{i,j} \log^2 q)$. By modular composition, we deduce $Y^{p^{t_{i,j}}} - (w_{i,j} \circ B) \left(\zeta^{p^{t_{i,j}}} \right)$ with $\tilde{O}((\deg \theta_{i,j})^\omega \log q)$ bit cost. After the extraction of p-th roots, the latter expression finally becomes $(Y - v_{i,j}(\zeta))^{p^{t_{i,j}}}$, with further $\tilde{O}(\deg \theta_{i,j} \log^2 q)$ bit operations.

By [3, Proposition 2.7], $\chi(X)$ is the characteristic polynomial of X in $\mathbb{K}[X, Y]/(F(X, Y), G(X, Y))$, so m_i is the intersection multiplicity of the points represented by $\theta_i(X) = Y - v_i(X) = 0$. □

3 DIVISOR

This section gathers complexity results for basic operations on smooth divisors of C.

3.1 Primitive element representation

A *smooth positive* divisor D of C is a multi-set of smooth points of C. The underlying set of points $\mathcal{E} = \{P_1, \dots, P_s\}$ is called the *support* of the divisor, and it is customary to write D as the formal sum $D = m_1P_1 + \cdots + m_sP_s$, where $m_i > 0$ is the *multiplicity* of P_i in D. Up to a linear change of variables we may assume that the support of D is in the affine chart $Z = 1$. In this case, a *primitive element* of D is a linear form $\lambda X + \mu Y$ that separates its support and satisfies the additional conditions:

$$\begin{vmatrix} \frac{\partial Q}{\partial X}(P_i) & \frac{\partial Q}{\partial Y}(P_i) \\ \lambda & \mu \end{vmatrix} \neq 0 \text{ for } i = 1, \dots, s. \tag{4}$$

Since the P_i are smooth on C, if $\mu \neq 0$, the latter condition is equivalent to requiring that λ/μ is outside the set

$$\left\{ \frac{\frac{\partial Q}{\partial X}(u(\zeta), v(\zeta), 1)}{\frac{\partial Q}{\partial Y}(u(\zeta), v(\zeta), 1)} : \chi(\zeta) = 0, \frac{\partial Q}{\partial Y}(u(\zeta), v(\zeta), 1) \neq 0 \right\}. \tag{5}$$

Geometrically speaking, this means that the line $\lambda X + \mu Y = 0$ is neither vertical nor tangent to the curve C at any point of D. If λ/μ is outside the sets (1) and (5) then it is primitive for D. The sum of the cardinalities of these two sets is $\leqslant \binom{\deg D+1}{2}$, where $\deg D := m_1 + \cdots + m_s$. Consequently, as soon as $|\mathbb{K}| > \binom{\deg D+1}{2}$, primitive elements can be found in $\mathbb{K}$.

Proposition 3.1. *Given a smooth positive divisor $D = m_1P_1 + \cdots + m_sP_s$ whose support is in the affine chart $Z = 1$, and given a primitive element $\lambda X + \mu Y$ for D, there exist unique polynomials χ, u, and v in $\bar{\mathbb{K}}[S]$ with the following properties:*

Div-H$_0$ *χ is monic of degree $\deg D$, and u, v have degrees $< \deg D$,*
Div-H$_1$ *$Q(u(S), v(S), 1) = 0 \bmod \chi(S)$,*
Div-H$_2$ *$\lambda u(S) + \mu v(S) = S$,*
Div-H$_3$ *$\mu \frac{\partial Q}{\partial X}(u(S), v(S), 1) - \lambda \frac{\partial Q}{\partial Y}(u(S), v(S), 1)$ is coprime to $\chi(S)$.*

Proof. We write χ_0, u_0, and v_0 for the primitive element representation of the support $\mathcal{E}$, so we have $Q(u_0(S), v_0(S), 1) = 0 \bmod \chi_0(S)$ and $\lambda u_0(S) + \mu v_0(S) = S$. The hypothesis (4) means that $(u_0(S), v_0(S))$ is a regular root modulo $\chi_0(S)$ of the map $\Xi : \mathbb{K}[S]^2 \to \mathbb{K}[S]^2$ defined by

$$\Xi : \begin{pmatrix} X \\ Y \end{pmatrix} \mapsto \begin{pmatrix} Q(X, Y, 1) \\ \lambda X + \mu Y - S \end{pmatrix}.$$

For $n \geqslant 0$ we appeal to the following Newton iteration based on Ξ:

$$\begin{pmatrix} u_{n+1} \\ v_{n+1} \end{pmatrix} := \begin{pmatrix} u_n \\ v_n \end{pmatrix} - D\Xi(u_n, v_n)^{-1} \Xi(u_n, v_n) \bmod \chi_0^{2^{n+1}}.$$

It follows that (u_n, v_n) is the unique root of Ξ modulo $\chi_0^{2^n}$ that coincides to (u_0, v_0) modulo χ_0. When 2^n is strictly larger than the largest multiplicity in D, we set

$$\chi(S) := (S - \lambda x_1 - \mu y_1)^{m_1} \cdots (S - \lambda x_r - \mu y_r)^{m_r},$$

where (x_i, y_i) denotes the coordinates of P_i, and then $u := u_n \bmod \chi$ and $v := v_n \bmod \chi$. Since χ divides $\chi_0^{2^n}$ the required properties are satisfied.

The uniqueness follows from the one of the lifted roots of Ξ, since conditions Div-H$_0$ to Div-H$_3$ imply that χ_0, $u \operatorname{rem} \chi_0$ and $v \operatorname{rem} \chi_0$ constitute the primitive element representation of $\mathcal{E}$; that means $u_0 = u \operatorname{rem} \chi_0$ and $v_0 = v \operatorname{rem} \chi_0$. □

A smooth positive divisor D as above will be represented by λ, μ, χ, u, v along with $\nabla Q(u, v, 1) \bmod \theta$, where θ denotes the square-free part of χ, and ∇Q represents the *gradient* of Q in the variables X and Y.

3.2 Lifting a divisor

We analyze the complexity of the Newton iteration seen in the proof of Proposition 3.1.

Lemma 3.2. *Let D be a smooth positive divisor parametrized by $\lambda X + \mu Y$. The representation of $2D$ by $\lambda X + \mu Y$ can be obtained with $\tilde{O}\left(\delta^{\frac{\omega}{2}+1} + (\deg D)^{\frac{\omega+2}{3}}\right)$ operations in $\mathbb{K}$.*

Proof. Let χ, u, v represent D, so $\Xi(u(S), v(S)) = 0 \bmod \chi(S)$. We can use the Newton iteration to obtain

$$\begin{pmatrix} \tilde{u}(S) \\ \tilde{v}(S) \end{pmatrix} := \begin{pmatrix} u(S) \\ v(S) \end{pmatrix} - D\Xi(u(S), v(S))^{-1} \cdot \Xi(u(S), v(S)) \bmod \chi(S)^2,$$

that yields $\Xi(\tilde{u}(S), \tilde{v}(S)) = 0 \bmod \chi(S)^2$. The evaluations of Q and of its partial derivatives at $(u(S), v(S), 1)$ modulo $\chi(S)^2$ take

$$\tilde{O}\left(\delta^{\frac{\omega}{2}+1} + (\deg \chi)^{\frac{\omega+2}{3}}\right)$$

operations in $\mathbb{K}$ by Lemma 2.1. The inverse of the determinant of $D\Xi(u(S), v(S))$ contributes to $\tilde{O}(\deg \chi)$. □

3.3 Nodal divisor

The *nodal divisor* of C, written E, will be given by a primitive element representation $\lambda_E, \mu_E, \chi_E, u_E, v_E$ of the set of singular points of C. In the terminology of the Brill–Noether algorithm, E plays the role of the *adjoint divisor* of C. Since E only depends on C, it might be regarded as a precomputation. Yet for the computation of a single Riemann–Roch space it is fair to take its cost into account. A probabilistic method is summarized in the next proposition. The hypothesis on $|\mathbb{K}|$ is flexible: in fact throughout the paper we have given priority to simple bounds that ensure (conditional) probabilities of success roughly about $1/2$ in the randomized sub-algorithms.

Proposition 3.3. *Assume $|\mathbb{K}| \geqslant \delta^4$. Given Q satisfying C-H$_1$, we can check if C-H$_2$ holds, compute $M \in \mathrm{GL}_3(\mathbb{K})$ such that $Q \circ M$ has degree δ in Y and its singular locus lies in the chart $Z = 1$, and get a primitive element representation of $M^{-1}(E)$, with a probabilistic algorithm of type Las Vegas that takes an expected number of $\tilde{O}(\delta^3)$ field operations in characteristic zero or $> \delta(\delta - 1)$, or $\tilde{O}(\delta^{2\omega} \log q + \delta^2 \log^2 q)$ bit operations if $\mathbb{K} = \mathbb{F}_q$.*

Proof. By taking α, β at random we easily find values in $\mathbb{K}$ such that $Q(X + \alpha Y, Y, Z + \beta Y)/Q(\alpha, 1, \beta)$ is monic in Y. The running time is $\tilde{O}(\delta^2)$ when using Lemma 2.2, since it suffices to ensure $Q(\alpha, 1, \beta) \neq 0$, and thanks to the Schwartz–Zippel lemma [5, Lemma 6.44] the expected number of trials is $O(1)$. From now we assume that Q is monic in Y.

The resultant $R(X, Z) := \mathrm{Res}_Y\left(Q, \frac{\partial Q}{\partial Y}\right)$ is homogeneous of degree $\delta(\delta - 1)$. So up to replacing Z by $Z + \gamma X$ in Q, we can further assume that R has degree $\delta(\delta - 1)$ in X with high probability. In particular the solution set $\mathcal{E}$ of $Q = \frac{\partial Q}{\partial Y} = 0$ lies in the chart $Z = 1$. Let $X + \mu Y$ be a candidate primitive element for it. Then, we replace X by $X - \mu Y$ in Q, so X is finally expected to be primitive. Lemma 2.4 applies to $\frac{\partial Q}{\partial Y}$ and Q: γ and μ are suitable if, and only

if, $R(X,1)$ has degree $\delta(\delta-1)$, that equals the number of solutions counted with multiplicities. Then we recover a parametrization $\theta(X) = Y - v(X) = 0$ of $\mathcal{E}$ via usual Chinese remaindering. The parametrization of E is deduced from

$$\begin{aligned} \theta_E(X) &:= \theta(X)/\gcd\left(\frac{\partial Q}{\partial X}(X, v(X), 1), \theta(X)\right) \\ v_E(E) &:= \theta(X) \operatorname{rem} \theta_E(X). \end{aligned}$$

C-H_2 holds if and only if the Hessian of Q has full rank at the singular points, that can be checked with further $\tilde{O}\left(\delta^{\frac{\omega+3}{2}}\right)$ operations in $\mathbb{K}$ thanks to Lemma 2.1. □

3.4 Decomposition of a divisor

For performing arithmetic operations on divisors efficiently we decompose them, operate on components, and recompose them. For a divisor D defined over $\mathbb{K}$ there exists a unique *equal multiplicity decomposition* written $\sum_{i=1}^{s} m_i D_i$, where:

- the D_i are positive, defined over $\mathbb{K}$, and made of simple points,
- and the m_i are pairwise distinct.

Decompositions and recompositions can be computed fast, as summarized in the following lemmas.

Lemma 3.4. *The equal multiplicity decomposition of a smooth positive divisor D over $\mathbb{K}$ takes $\tilde{O}(\deg D)$ fields operations in characteristic zero or $> \deg D$, or $\tilde{O}(\deg D \log^2 q)$ bit operations if $\mathbb{K} = \mathbb{F}_q$.*

Proof. Let $\lambda X+\mu Y, \chi, u, v$ represent D as above. We compute the squarefree factorization of χ into $\theta_1^{m_1} \cdots \theta_s^{m_s}$, with $\tilde{O}(\deg D)$ field operations in characteristic zero or $> \deg D$, and with $\tilde{O}(\deg D \log^2 q)$ bit operations if $\mathbb{K} = \mathbb{F}_q$. So D writes as $m_1 D_1 + \cdots + m_s D_s$, where D_i is parametrized by $\lambda X + \mu Y$, $\chi_i := \theta_i^{m_i}$, $u_i := u \operatorname{rem} \chi_i$, and $v_i := v \operatorname{rem} \chi_i$ and $\nabla Q(u_i, v_i, 1) \bmod \theta_i := \nabla Q(u, v, 1) \bmod \theta_i$. Using fast multi-remaindering, this takes $\tilde{O}(\deg D)$ operations in $\mathbb{K}$; see [5, chapter 10]. □

Lemma 3.5. *Let $D_1, \ldots, D_s$ be smooth positive divisors over $\mathbb{K}$, with disjoint supports, and parametrized by the same primitive element $\lambda X + \mu Y$. If $\lambda X + \mu Y$ is primitive for the sum $D := D_1 + \cdots + D_s$, then its representation can be obtained with $\tilde{O}(\deg D)$ operations in $\mathbb{K}$.*

Proof. Let χ_i, u_i, v_i represent D_i, and let θ_i denote the squarefree part of χ_i. By assumption, the χ_i are pairwise coprime. Then $\chi := \chi_1 \cdots \chi_s$ can be computed with $\tilde{O}(\deg D)$ operations in $\mathbb{K}$; see [5, chapter 10]. Since u, v and $\nabla Q(u, v, 1) \bmod \theta$ satisfy $u_i = u \operatorname{rem} \chi_i$, $v_i = v \operatorname{rem} \chi_i$, $\nabla Q(u_i, v_i, 1) = \nabla Q(u, v, 1) \operatorname{rem} \theta_i$ for $i = 1, \ldots, s$, they can be obtained via Chinese remaindering with $\tilde{O}(\deg D)$ operations in $\mathbb{K}$. □

3.5 Change of primitive element

Assume that $D := m(P_1 + \cdots + P_s)$, so $\chi = \theta^m$ with θ separable of degree s. Consider

$$\begin{aligned} \Gamma: \quad \mathbb{K}[S]/(\chi(S)) &\cong (\mathbb{K}[Z]/(\theta(Z)))[[T-Z]]/(T-Z)^m \quad (6) \\ S &\mapsto T. \end{aligned}$$

Both directions of this isomorphism can be computed in softly linear time, namely $\tilde{O}(\deg D)$; see [10, section 4.2]. In fact, $(\Gamma(u), \Gamma(v))$ can be regarded as the simultaneous power series expansions of Q at $P_1, \ldots, P_s$ with precision m. In order to change the primitive element for D, we first examine what happens to the underlying support, and then change the representations in the power series expansions.

Lemma 3.6. *Given $D = m(P_1 + \cdots + P_s)$ over $\mathbb{K}$ parametrized by $\lambda X + \mu Y$, and given $(\tilde{\lambda}, \tilde{\mu}) \in \mathbb{K}^2$, we can test if $\tilde{\lambda} X + \tilde{\mu} Y$ is primitive for D, and, if so, compute the corresponding representation, with $O((\deg D)^\omega)$ field operations in characteristic zero or $> \deg D$, or $(\deg D)^\omega \tilde{O}(\log q) + \tilde{O}(\deg D \log^2 q)$ bit operations if $\mathbb{K} = \mathbb{F}_q$.*

Proof. First it is checked that $\begin{vmatrix} \frac{\partial Q}{\partial X}(u, v, 1) & \frac{\partial Q}{\partial Y}(u, v, 1) \\ \tilde{\lambda} & \tilde{\mu} \end{vmatrix}$ is invertible modulo θ. If so, we change the primitive element for the support of D by means of Lemma 2.3:

$$\begin{aligned} \Phi: \quad \mathbb{K}[S]/(\theta(S)) &\cong \mathbb{K}[S]/(\tilde{\theta}(S)) \\ S &\mapsto w(S). \end{aligned}$$

That takes $O((\deg \theta)^\omega)$ operations in $\mathbb{K}$ in characteristic zero or $> \deg \theta$, or $(\deg \theta)^\omega \tilde{O}(\log q) + \tilde{O}(\deg \theta \log^2 q)$ bit operations if $\mathbb{K} = \mathbb{F}_q$. We convert D to local representation and get the following diagram:

$$\begin{array}{lccc} \Gamma: & \mathbb{K}[S]/(\chi(S)) & \to & (\mathbb{K}[Z]/(\theta(Z)))[[T-Z]]/(T-Z)^m \\ & & & \downarrow \text{ coefficient-wise extension of } \Phi \\ \tilde{\Gamma}: & \mathbb{K}[S]/(\tilde{\chi}(S)) & \to & (\mathbb{K}[Z]/(\tilde{\theta}(Z)))[[T-Z]]/(T-Z)^m, \end{array}$$

where $\tilde{\chi}(S) := \tilde{\theta}(S)^m$. The parametrization of D in terms of $\tilde{\lambda} X + \tilde{\mu} Y$ is $\tilde{u}(S) := \tilde{\Gamma}^{-1}(\Phi(\Gamma(u(S))))$ and $\tilde{v}(S) := \tilde{\Gamma}^{-1}(\Phi(\Gamma(v(S))))$. That incurs $O(m)$ compositions modulo $\tilde{\theta}$, that is $O(m(\deg \tilde{\theta})^\omega) = O((\deg D)^\omega)$. Finally $\nabla Q(\tilde{u}, \tilde{v}, 1) \bmod \tilde{\theta}$ involves two compositions modulo $\tilde{\theta}$. □

Proposition 3.7. *Given a smooth positive divisor D parametrized by $\lambda X + \mu Y$, and given $(\tilde{\lambda}, \tilde{\mu}) \in \mathbb{K}^2$, we can test if $\tilde{\lambda} X + \tilde{\mu} Y$ is primitive for D, and, if so, compute the corresponding representation with $O((\deg D)^\omega)$ field operations in characteristic zero or $> \deg D$, or $(\deg D)^\omega \tilde{O}(\log q) + \tilde{O}(\deg D \log^2 q)$ bit operations if $\mathbb{K} = \mathbb{F}_q$.*

Proof. We compute the equal multiplicity decomposition of $D = m_1 D_1 + \cdots + m_s D_s$ as in Lemma 3.4. For each separable factor D_i of multiplicity m_i, we try to compute the primitive element representation $\tilde{\chi}_i, \tilde{u}_i, \tilde{v}_i$ of the support of D_i for $\tilde{\lambda} X + \tilde{\mu} Y$, by Lemma 3.6. If it fails then $\tilde{\lambda} X + \tilde{\mu} Y$ cannot be primitive for D. In order to check that $\tilde{\lambda} X + \tilde{\mu} Y$ is finally primitive for D it remains to verify that the squarefree parts of the $\tilde{\chi}_i$ are coprime. Then we may glue the representations of the $m_i D_i$ via Lemma 3.5. □

3.6 Sum of divisors

Gathering tools presented above we obtain efficient sums and subtractions for divisors.

Proposition 3.8. *Given two smooth positive divisors D_1 and D_2 such that $|\mathbb{K}| \geqslant (\deg D_1 + \deg D_2)^2$, the sum $D := D_1 + D_2$ can be computed with a probabilistic algorithm of type Las Vegas that takes an expected $\tilde{O}\left(\delta^{\frac{\omega}{2}+1} + (\deg D)^{\frac{\omega+1}{2}}\right)$ field operations in characteristic zero or $> \deg D$, and $\tilde{O}\left(\left(\delta^{\frac{\omega}{2}+1} + (\deg D)^{\frac{\omega+1}{2}}\right) \log q + \deg D \log^2 q\right)$ bit operations if $\mathbb{K} = \mathbb{F}_q$.*

Proof. First, a common primitive element $\lambda X + \mu Y$ is found at random for D_1 and D_2 with an expected $O(\deg D^\omega)$ field operations in characteristic zero or $> \deg D$, or $(\deg D)^\omega \tilde{O}(\log q) + \tilde{O}(\deg D \log^2 q)$ bit operations if $\mathbb{K} = \mathbb{F}_q$, by Proposition 3.7. The number of trials is $O(1)$ thanks to the assumption on $|\mathbb{K}|$.

We split D_i into $\check{D}_i + \hat{D}_i$ for $i = 1, 2$ such that $\hat{D}_1$ and $\hat{D}_2$ have the same support $\mathcal{E}$, itself disjoint from $\check{D}_1$ and $\check{D}_2$. Let $\hat{\chi}_i, \hat{u}_i, \hat{v}_i$ denote the parametrization of $\hat{D}_i$ for $i = 1, 2$. Let w_1 and w_2 be the cofactors in the Bézout relation $\gcd(\hat{\chi}_1, \hat{\chi}_2) = w_1\hat{\chi}_1 + w_2\hat{\chi}_2$, then

$$\begin{aligned}\tilde{\chi}_3 &:= \operatorname{lcm}(\hat{\chi}_1, \hat{\chi}_2),\\ \tilde{u}_3 &:= \hat{u}_1 w_2(\hat{\chi}_2/\gcd(\hat{\chi}_1, \hat{\chi}_2)) + \hat{u}_2 w_1(\hat{\chi}_1/\gcd(\hat{\chi}_1, \hat{\chi}_2)) \operatorname{rem} \tilde{\chi}_3,\\ \tilde{v}_3 &:= \hat{v}_1 w_2(\hat{\chi}_2/\gcd(\hat{\chi}_1, \hat{\chi}_2)) + \hat{v}_2 w_1(\hat{\chi}_1/\gcd(\hat{\chi}_1, \hat{\chi}_2)) \operatorname{rem} \tilde{\chi}_3,\end{aligned}$$

is the parametrization of the divisor of support $\mathcal{E}$ where the multiplicity of a point P in it is the maximum of the multiplicities of P in $\hat{D}_1$ and $\hat{D}_2$. Therefore the parametrization of $D_3 := \hat{D}_1 + \hat{D}_2$ is deduced by means of a single lifting step, that costs

$$\tilde{O}\left(\delta^{\frac{\omega}{2}+1} + (\deg D_3)^{\frac{\omega+2}{3}}\right) \text{ by Lemma 3.2.}$$

Glueing $\check{D}_1 + \check{D}_2 + D_3$ takes softly linear time by Lemma 3.5. □

Proposition 3.9. *Given two smooth positive divisors D_1 and D_2 by their primitive element representations, and such that $|\mathbb{K}| \geqslant (\deg D_1 + \deg D_2)^2$, a representation of $[D_1 - D_2]_+$ can be computed with a probabilistic algorithm of type Las Vegas that takes an expected number of $\tilde{O}((\deg D_1)^\omega + (\deg D_2)^\omega)$ field operations in characteristic zero or $> \deg D_1 + \deg D_2$, or*

$$\tilde{O}(((\deg D_1)^\omega + (\deg D_2)^\omega)\log q + (\deg D_1 + \deg D_2)\log^2 q)$$

bit operations if $\mathbb{K} = \mathbb{F}_q$.

Proof. First, a common primitive element $\lambda X + \mu Y$ is found for D_1 and D_2 as is the proof of the latter proposition, and let χ_i, u_i, v_i denote the parametrization of D_i for $i = 1, 2$. The parametrization of $[D_1 - D_2]_+$ is $\chi := \chi_1/\gcd(\chi_1, \chi_2)$, $u = u_1 \operatorname{rem} \chi$, $v := v_1 \operatorname{rem} \chi$. □

4 RIEMANN–ROCH SPACE

We are now ready to revisit the Brill–Noether strategy. For the mathematical aspects of the proofs below, we refer the reader to [19]. The main improvements upon [19] concern fast structured linear algebra, and the extension to any smooth input divisor D.

4.1 Shifted Popov form

Let M denote a $m \times n$ matrix with entries in $\mathbb{K}[X]$, and let us consider a vector $\boldsymbol{s} := (s_1, \ldots, s_n) \in \mathbb{Z}^n$ called a *shift* for the degrees. The $\boldsymbol{s}$-degree of a row vector $\boldsymbol{a} = (a_1, \ldots, a_n)$ in $\mathbb{K}[X]^n$ is defined as $\deg_{\boldsymbol{s}} \boldsymbol{a} := \max(a_1 + s_1, \ldots, a_n + s_n)$. If $\boldsymbol{a}$ is non-zero then the *pivot index* of $\boldsymbol{a}$ is the largest index i where the latter maximum is attained. The entry a_i is called the *pivot*, and its degree is the *pivot degree*. If $\boldsymbol{a}$ is zero then its pivot index is set to zero. The matrix M is in *Popov form* if the following properties are satisfied:

- The positive pivot indices of the rows of M are in increasing order;
- The pivots of the rows of M are monic;
- The pivots of M have a degree strictly larger than the other elements in their column.

When $m = n$ and M is nonsingular then its pivots are the diagonal elements. In this case, M satisfies the "predictable degree" property:

Lemma 4.1. *If $\boldsymbol{b} = (b_1, \ldots, b_n) := \boldsymbol{a}M$, then $\deg b_i + s_i = d_i + \deg a_i$ for $i = 1, \ldots, n$, where d_i denotes the $\boldsymbol{s}$-degree of the i-th row of M.*

The "naive algorithm" for Popov forms takes $\tilde{O}(mnr(\deg M)^2)$ operations in $\mathbb{K}$, where r is the rank of M and when $\boldsymbol{s}$ is zero [22, Theorem 7.1]. The current best bounds are $\tilde{O}(m^{\omega-1}nd)$ for a $m \times n$ matrix with $m \leqslant n$ [23, 24].

Proposition 4.2. *Assume that M is square, nonsingular, and in Popov form as above. Then, the elements of $\boldsymbol{s}$-degree $\leqslant d$ in the $\mathbb{K}[X]$-module generated by the rows $M_1, \ldots, M_n$ of M form a $\mathbb{K}$-vector space of basis $X^j M_i$, $j = 0, \ldots, d - d_i$, $i = 1, \ldots, n$.*

Proof. A $\mathbb{K}$-relation between the elements of the candidate basis leads to a $\mathbb{K}[X]$-relation between the rows of M. According to the assumptions, such a proper relation cannot exist so the candidate basis is free over $\mathbb{K}$. An element $X^j M_i$ with $j = 0, \ldots, d - d_i$ and $i = 1, \ldots, n$ satisfies $\boldsymbol{s}\text{-}\deg(X^j M_i) \leqslant j + d_i \leqslant d$. Conversely let $\boldsymbol{b}$ be a $\mathbb{K}[X]$-combination $\boldsymbol{a}M$ of the rows of M of $\boldsymbol{s}$-degree $\leqslant d$. Lemma 4.1 implies that $\deg a_i = \deg b_i + s_i - d_i \leqslant d - d_i$. □

4.2 Bivariate interpolation

Considering Y as the "main variable", the subset of polynomials in $\mathbb{K}[X][Y]$ that vanish at a given set of points is a free $\mathbb{K}[X]$-module. This motivates the definition of *basis generator* of a vector subspace of polynomials of degree $\leqslant d$ in $\mathbb{K}[X, Y]$: this is a set $F_1, \ldots, F_n$ of polynomials such that $X^j F_i$, $j = 0, \ldots, d - \deg F_i$, $i = 1, \ldots, n$ form a vector basis; n is called the *rank*.

Proposition 4.3. *Let D be a smooth positive divisor and let $d \geqslant 1$. Assume that D and E are parametrized by X and write $\chi(X) = Y - v(X) = 0$ for the corresponding parametrization of D. Then, there exists a basis generator of rank $\leqslant \delta$ of polynomials $F \in \mathbb{K}[X, Y]$ of degree $\leqslant d$ such that $F(X, v(X)) = 0 \bmod \chi(X)$, $F(X, v_E(X)) = 0 \bmod \chi_E(X)$, and $\deg_Y F < \delta$. It takes $\tilde{O}(\min(d, \delta)^{\omega-1}(r + \deg D))$ operations in $\mathbb{K}$; recall that $r = \deg E$.*

Proof. Let $n := \min(d, \delta - 1)$. The parametrization of E in this context is $\chi_E(X) = Y - v_E(X) = 0$. In softly linear time we compute $a_i = v^i \operatorname{rem} \chi$ and $b_i = v_E^i \operatorname{rem} \chi_E$, for $i = 0, \ldots, n$ and then consider the $\mathbb{K}[X]$-module

$$\mathcal{M} := \{(f_0, \ldots, f_n) \in \mathbb{K}[X]^{n+1} : f_0 a_0 + \cdots + f_n a_n = 0 \bmod \chi \text{ and } f_0 b_0 + \cdots + f_n b_n = 0 \bmod \chi_E\}. \quad (7)$$

Since $\mathbb{K}[X]$ is principal, $\mathcal{M}$ is a free module of rank $n + 1$, because it contains $(0, \ldots, 0, \chi, 0, \ldots, 0)$ with χ at position i, for all $i = 0, \ldots, n$.

Using [23, Theorem 1.4] with $\boldsymbol{s} := (d, d - 1, \ldots, d - n)$, the nonsingular matrix in $\boldsymbol{s}$-Popov form whose rows are a basis of $\mathcal{M}$ can be computed with $\tilde{O}(n^{\omega-1}(\deg \chi + \deg \chi_E))$ operations in $\mathbb{K}$. □

4.3 Denominator

Let $Q_1, \ldots, Q_r$ denote the singular points of C, let $C' \to C$ be the desingularization map for C, and for any $i \in \{1, \ldots, r\}$ let $Q_{i,1}$ and $Q_{i,2}$ represent the points of C' above Q_i. In the sequel we set

$$d := \left\lceil \frac{(\delta - 1)(\delta - 2) + \deg D_+}{\delta} \right\rceil. \quad (8)$$

Lemma 4.4. *There exists a non-zero homogeneous polynomial H in $\bar{\mathbb{K}}[X,Y,Z]$ of degree $\leqslant d$ such that Q does not divide H, $(H)_0 \geqslant D_+$, $(H)_0 \geqslant E$, and the intersection multiplicities of H at the singular points of C are 2 (recall that $(H)_0 \geqslant E$ means "H is adjoint to C").*

Proof. Let us fix a homogeneous polynomial $L \in \mathbb{K}[X,Y,Z]$ of degree 1, and set $\bar{D} := D_+ + \sum_{j=1}^{r}(Q_{i,1}+Q_{i,2})$. Since C has degree δ with only ordinary singularities, its genus is $g = \frac{(\delta-1)(\delta-2)}{2} - r$, and we have $\deg(L)_0 = \delta$, so the hypothesis on d means $\deg(d(L)_0 - \bar{D}) \geqslant 2g$. The Riemann–Roch theorem [4, Corollary 3] thus implies that

$$\dim(\mathcal{L}(d(L)_0 - \bar{D} - Q_{i,j})) = \dim(\mathcal{L}(d(L)_0 - \bar{D})) - 1$$

holds for all $(i,j) \in \{1,\dots,r\} \times \{1,2\}$.

So far we have proved that for any $(i,j) \in \{1,\dots,r\} \times \{1,2\}$ there exists a function $h_{i,j} \in \mathcal{L}(d(L)_0 - \bar{D})$ that is not contained in $\mathcal{L}(d(L)_0 - \bar{D} - Q_{i,j})$. By [7, Théorème 2.7.1] (or [8, Théorème 2.5]), $h_{i,j}$ has a rational function representation of the form $\frac{H_{i,j}}{L^d}$, where $H_{i,j} \in \bar{\mathbb{K}}[X,Y,Z]$ is homogeneous of degree d and is not divisible by Q. In other words $(H_{i,j})_0 \geqslant D_+$, $(H_{i,j})_0 \geqslant E$, and the intersection multiplicity of $H_{i,j}$ at $Q_{i,j}$ is 2.

Let $\alpha_{i,j}$ for $i = 1,\dots,r$ and $j = 1,2$ be parameters in $\bar{\mathbb{K}}$, and consider $H := \sum_{i=1}^{r}\sum_{j=1}^{2} \alpha_{i,j} H_{i,j}$. By construction $(H)_0 \geqslant D_+$ and $(H)_0 \geqslant E$ hold, and the resultant $\mathrm{Res}_Y(Q,H)$ is a non-zero polynomial in all the $\alpha_{i,j}$ when regarded in $\mathbb{K}[\alpha_{1,1},\dots,\alpha_{r,2}][X]$. Let $T_{i,j}$ denote a tangent vector at the image in C of the germ of curve of C' at $Q_{i,j}$. Regarded in $\bar{\mathbb{K}}[\alpha_{1,1},\dots,\alpha_{r,2}]$ the polynomial $\prod_{i=1}^{r}\prod_{j=1}^{2}(T_{i,j} \cdot \nabla H(Q_i))$ is non-zero. Consequently almost all choices of the $\alpha_{i,j}$ yield H with the required properties. □

Algorithm 2

Input: $Q \in \mathbb{K}[X,Y,Z]$, E, a smooth divisor D on C.

Output: $M \in \mathrm{GL}_3(\mathbb{K})$, $H \in \mathbb{K}[X,Y]$ of degree $\leqslant d$ such that $\deg_Y H < \delta$, $(H)_0 \geqslant M^{-1}(D_+)$, $(H)_0 \geqslant M^{-1}(E)$, and with intersection multiplicities exactly 2 at the singular points of $M^{-1}(C)$; and $(H)_0 - 2M^{-1}(E)$ for which X is primitive.

Assumptions: C-H_1, C-H_2, $\deg_Y Q = \delta$; the supports of E and D are in the chart $Z = 1$.

(1) Take α, β at random in $\mathbb{K}$ and set
$$M := \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ \beta & 0 & 1 \end{pmatrix}^{-1} \begin{pmatrix} 1 & \alpha & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}^{-1}.$$
(2) If $\deg_Y(Q \circ M) \neq \delta$ then go to step 1.
(3) If the supports of $M^{-1}(E)$ and $M^{-1}(D)$ are not in the chart $Z = 1$ then go to step 1.
(4) If X is primitive for $M^{-1}(E)$ and $M^{-1}(D_+ + D_-)$, then compute its primitive element representation as in section 2.2. Otherwise go to step 1.
(5) Let d be as in (8). Compute a basis generator $H_1,\dots,H_l$ of the polynomials H satisfying $\deg H \leqslant d$, $\deg_Y H < \delta$, $(H)_0 \geqslant M^{-1}(D_+)$, $(H)_0 \geqslant M^{-1}(E)$.
(6) Set $H(X,Y) := \sum_{i=1}^{l} \alpha_i(X) H_i(X,Y)$ with $\alpha_i(X) \in \mathbb{K}[X]_{\leqslant d - \deg H_i}$ taken at random.
(7) Compute the intersection of $H(X,Y) = 0$ and $(Q \circ M)(X,Y,1) = 0$. If the cardinality of the solution set is not δd counting multiplicities, or does not admit X as a primitive element then go to step 1.
(8) If the multiplicities of H at the singular points of $Q \circ M$ are not 2, then go to step 6.
(9) Return M, H and $(H)_0 - 2M^{-1}(E)$.

Proposition 4.5. *Assume $|\mathbb{K}| > 6(\delta d)^2$. Algorithm 2 is correct and takes an expected $\tilde{O}\left((\delta^2 + \deg D_+)^{\frac{\omega+1}{2}}\right)$ operations in characteristic 0 or $> \delta d$, or $\tilde{O}\left((\delta^2 + \deg D_+)^{\frac{\omega+1}{2}} \log q + (\delta^2 + \deg D_+)\log^2 q\right)$ bit operations when $\mathbb{K} = \mathbb{F}_q$.*

Proof. By Lemma 4.4 there exists $\bar{H} \in \bar{\mathbb{K}}[X,Y,Z]$ of degree d not divisible by Q, such that $(\bar{H})_0 \geqslant D_+$, $(\bar{H})_0 \geqslant E$, and with intersection multiplicities exactly 2 at the singular points of C. Let $\mathcal{E}$ denote the set of the zeros of $\bar{H}$ on C. Since Q is monic in Y, $(0:0:1) \notin \mathcal{E}$. Consequently for all but finite number of values of β the set $M(\mathcal{E})$ is in the chart $Z = 1$. On the other hand for almost all values of α, the form X is primitive for $M^{-1}(\mathcal{E})$ and $\deg_Y(Q \circ M) = \delta$ holds. Then, $(\bar{H} \circ M) \bmod_Y (Q \circ M)$ belongs to the $\bar{\mathbb{K}}$ extension of the polynomial space computed in step 5. Consequently the algorithm finishes with a correct output for almost all values of $\alpha, \beta, \alpha_1, \dots, \alpha_n$ over $\bar{\mathbb{K}}$.

Let us now estimate the probabilities involved by random choices over $\mathbb{K}$. The coefficient of Y^δ in $Q \circ M$ is a non-zero polynomial of degree $\leqslant 2\delta$ in α, β. By the Schwartz-Zippel lemma, $\deg_Y(Q \circ M) = \delta$ fails with probability $\leqslant \frac{2\delta}{6(d\delta)^2} < \frac{1}{2}$. Assuming that step 2 succeeds, then the probability of step 3 failing is

$$\leqslant \frac{r + \deg D_+ + \deg D_-}{6(d\delta)^2} \leqslant \frac{(\delta-1)(\delta-2) + 4\deg D_+}{6(d\delta)^2} \leqslant \frac{2\delta d}{6(d\delta)^2} \leqslant \frac{1}{2}.$$

Once step 3 has succeeded then β is properly fixed, step 4 requires that X be primitive. Using (1), this fails with probability

$$\leqslant \frac{\binom{r+\deg D_+ + \deg D_- + 1}{2}}{6(d\delta)^2} \leqslant \frac{2(\delta d)^2 + \delta d}{6(d\delta)^2} \leqslant \frac{1}{2}.$$

The coefficient of $X^{\delta d}$ in $R(X,Z) := \mathrm{Res}_Y(Q \circ M, Z^d H(X/Z, Y/Z))$ is a non-zero homogeneous polynomial of degree $\leqslant \delta$ in the coefficients of $\alpha_1,\dots,\alpha_n$. In addition the discriminant of the separable part of $R(X,1)$ is non-zero of degree $\leqslant 2\delta^2 d$ in the coefficients of $\alpha_1,\dots,\alpha_n$. Thus, the probability that step 7 fails is $\leqslant \frac{2\delta^2 d + \delta}{6(d\delta)^2} \leqslant \frac{1}{2}$.

Let $T_{i,j}$ denote a tangent vector at the image in C of the germ of curve of C' at $Q_{i,j}$. The polynomial $\prod_{i=1}^{r}\prod_{j=1}^{2}(T_{i,j} \cdot \nabla H(Q_i))$ is non-zero of total degree $2r$ in the coefficients of $\alpha_1,\dots,\alpha_n$. By the Schwartz–Zippel lemma, the probability that step 8 fails given that all the previous steps succeeded is $\leqslant \frac{2r}{6(d\delta)^2} \leqslant \frac{1}{2}$. Consequently, the expected number of times the algorithm returns to step 1 or step 6 is $O(1)$.

Assume that $\mathbb{K}$ has characteristic 0 or $> d\delta$. Step 2 takes softly linear time by Lemma 2.2. Step 4 contributes to $\tilde{O}(r^\omega + (\deg D_+)^\omega)$ by Proposition 3.7. Step 5 takes $\tilde{O}(\min(d,\delta)^{\omega-1}(r + \deg D_+)) = \tilde{O}(\delta^{\omega-1}(r + \deg D_+))$ by Proposition 4.3. Step 6 contributes to $\tilde{O}(\delta d)$. Step 7 is done via Lemma 2.4 with $\tilde{O}(d\delta^2)$ operations in $\mathbb{K}$.

In step 8 since X is primitive for $(H)_0$ and E and since the intersection multiplicities in $(H)_0$ are known, we can conveniently check whether the points in E have intersection multiplicity 2. And if so we deduce a primitive element representation of $(H)_0 - 2M^{-1}(E)$ in softly linear time. The total complexity bound is obtained by summing the cost of each step, thanks to $r = O(\delta^2)$ and $\delta^{\omega-1}(\delta^2 +$

$\deg D_+) = O\left((\delta^2 + \deg D_+)^{\frac{\omega+1}{2}}\right)$. The same kind of analysis applies over $\mathbb{F}_q$, and is left to the reader. □

The value of d defined in (8) guarantees that a denominator H can be found in degree $\leqslant d$. In favorable cases, smaller values for d are possible: in fact, when $\deg D_+ = O(\delta^2)$ and $r = 0$ the degree bound d used in [19] is sharper.

4.4 Riemann–Roch space

Once we have obtained a common denominator H as above for $\mathcal{L}(D)$, we focus on the numerators, as follows.

Algorithm 3
Input: $Q \in \mathbb{K}[X, Y, Z]$ of degree δ, E, and a smooth divisor D on C.
Output: a basis generator of rank $\leqslant \delta$ of $\mathcal{L}(D)$.
Assumptions: C-H_1, C-H_2, $\deg_Y Q = \delta$; the supports of E and D are in the chart $Z = 1$.
(1) Compute M, H and $D_{\mathrm{res}} := (H)_0 - 2M^{-1}(E)$ by means of Algorithm 2.
(2) Compute $D_{\mathrm{num}} = M^{-1}(D_-) + (D_{\mathrm{res}} - M^{-1}(D_+))$.
(3) Compute a basis generator $G_1, \ldots, G_l$ of the vector space of polynomials in $\mathbb{K}[X, Y]$ of degree $\leqslant d$ such that $(G)_0 \geqslant D_{\mathrm{num}}$ and $(G)_0 \geqslant M^{-1}(E)$.
(4) Return M and $G_1, \ldots, G_l$.

Proposition 4.6. *Assume that $|K| \geqslant 6(\delta d)^2$. Then, Algorithm 3 is correct and takes an expected $\tilde{O}\left((\delta^2 + \deg D_+)^{\frac{\omega+1}{2}}\right)$ field operations in characteristic 0 or $> \delta d$, or*

$$\tilde{O}\left((\delta^2 + \deg D_+)^{\frac{\omega+1}{2}} \log q + (\delta^2 + \deg D_+) \log^2 q\right)$$

bit operations when $\mathbb{K} = \mathbb{F}_q$.

Proof. Combination of Propositions 3.8, 3.9, 4.3 and 4.5. The correctness comes from the more general Brill–Noether framework, we refer to [7, 17, 19] for detailed proofs. The only significant difference is our choice of H. Since $H \geq E$ by definition, H is a suitable denominator by [7, Théorème 2.7.1]. □

Proof. *(Proof of Theorem 1.1)* First we use Proposition 3.3. Once the resulting change of variables is applied to Q and E, Proposition 4.6 yields the claimed complexity bounds. □

In the case $\mathbb{K} = \mathbb{F}_q$, most of the auxiliary routines get closer to optimality in theory: for bivariate composition, bounds *à la* Kedlaya–Umans are quasi-linear. Unfortunately they have not led to efficient practical implementations so far; see [14, section 8]. For curve intersections, better complexity bounds also exist, but under genericity assumptions; see [13, 26]. If assumptions could be dropped then the complexity bound of Theorem 1.1 would become $\tilde{O}(\delta^{\omega-1}(r + \deg D_+) \log q + (\delta^2 + \deg D_+)^{1+\epsilon} \log q + (\delta^2 + \deg D_+) \log^2 q)$ bit operations, for any fixed $\epsilon > 0$. The bottleneck would be structured linear algebra underlying Proposition 4.3. If ω were further proved to be close to 2, then our algorithm would be close to optimal in terms of the size of the output whenever $r = O(\deg D_+)$.

ACKNOWLEDGMENTS

This paper is part of a project that has received funding from the French "Agence de l'Innovation de Défense". We are grateful to Vincent Neiger for helpful discussions.

REFERENCES

[1] G. Chèze and G. Lecerf. Lifting and recombination techniques for absolute factorization. *J. Complexity*, 23(3):380–420, 2007.
[2] X. Dahan, M. Moreno Maza, É. Schost, and Yuzhen Xie. On the complexity of the D5 principle. In J.-G. Dumas, editor, *Proceedings of Transgressive Computing 2006: a conference in honor of Jean Della Dora*, pages 149–168. U. J. Fourier, Grenoble, France, 2006.
[3] C. Durvye and G. Lecerf. A concise proof of the Kronecker polynomial system solver from scratch. *Expo. Math.*, 26(2), 2007.
[4] W. Fulton. *Algebraic Curves – An Introduction to Algebraic Geometry.* Addison-Wesley, 1989.
[5] J. von zur Gathen and J. Gerhard. *Modern computer algebra.* Cambridge University Press, New York, 3rd edition, 2013.
[6] B. Grenet, J. van der Hoeven, and G. Lecerf. Deterministic root finding over finite fields using Graeffe transforms. *Appl. Algebra Engrg. Comm. Comput.*, 27(3):237–257, 2016.
[7] G. Haché. *Construction Effective des Codes Géométriques.* PhD thesis, Université Paris 6, 1996.
[8] G. Haché. *L'algorithme de Brill-Noether appliqué aux courbes réduites.* Rapport de recherche n° 1998-01, Laboratoire d'Arithmétique, de Calcul formel et d'Optimisation ESA - CNRS 6090, Université de Limoges, France, 1998. https://www.unilim.fr/laco/rapports/1998/R1998_01.pdf.
[9] F. Hess. Computing Riemann–Roch spaces in algebraic function fields and related topics. *J. Symbolic Comput.*, 33(4):425–445, 2002.
[10] J. van der Hoeven and G. Lecerf. Composition modulo powers of polynomials. In *Proceedings of the 2017 ACM on International Symposium on Symbolic and Algebraic Computation*, ISSAC '17, pages 445–452. New York, NY, USA, 2017. ACM.
[11] J. van der Hoeven and G. Lecerf. Directed evaluation. Technical Report, HAL, 2018. https://hal.archives-ouvertes.fr/hal-01966428.
[12] J. van der Hoeven and G. Lecerf. On the complexity exponent of polynomial system solving. *Found. Comput. Math.*, 2020. https://doi.org/10.1007/s10208-020-09453-0.
[13] J. van der Hoeven and G. Lecerf. Fast computation of generic bivariate resultants. Technical Report, HAL, 2019. https://hal.archives-ouvertes.fr/hal-02080426.
[14] J. van der Hoeven and G. Lecerf. Fast multivariate multi-point evaluation revisited. *J. Complexity*, 56:101405, 2020.
[15] Xiaohan Huang and V. Y. Pan. Fast rectangular matrix multiplication and applications. *J. Complexity*, 14(2):257–299, 1998.
[16] K. Khuri-Makdisi. Asymptotically fast group operations on Jacobians of general curves. *Math. Comp.*, 76(260):2213–2239, 2007.
[17] D. Le Brigand and J.-J. Risler. Algorithme de Brill–Noether et codes de Goppa. *Bulletin de la société mathématique de France*, 116(2):231–253, 1988.
[18] F. Le Gall. Powers of tensors and fast matrix multiplication. In K. Nabeshima, editor, *ISSAC'14: International Symposium on Symbolic and Algebraic Computation*, pages 296–303. New York, NY, USA, 2014. ACM.
[19] A. Le Gluher and P.-J. Spaenlehauer. A fast randomized geometric algorithm for computing Riemann–Roch spaces. *Math. Comp.*, 2019. https://doi.org/10.1090/mcom/3517.
[20] G. Lecerf. Fast separable factorization and applications. *Appl. Algebra Engrg. Comm. Comput.*, 19(2):135–160, 2008.
[21] G. Lecerf. On the complexity of the Lickteig–Roy subresultant algorithm. *J. Symbolic Comput.*, 92:243–268, 2019.
[22] T. Mulders and A. Storjohann. On lattice reduction for polynomial matrices. *J. Symbolic Comput.*, 35(4):377–401, 2003.
[23] V. Neiger. Fast computation of shifted Popov forms of polynomial matrices via systems of modular polynomial equations. In *Proceedings of the ACM on International Symposium on Symbolic and Algebraic Computation*, ISSAC '16, pages 365–372. New York, NY, USA, 2016. ACM.
[24] V. Neiger, J. Rosenkilde, and G. Solomatov. Computing Popov and Hermite forms of rectangular polynomial matrices. In *Proceedings of the 2018 ACM International Symposium on Symbolic and Algebraic Computation*, ISSAC '18, pages 295–302. New York, NY, USA, 2018. ACM.
[25] M. Nüsken and M. Ziegler. Fast multipoint evaluation of bivariate polynomials. In S. Albers and T. Radzik, editors, *Algorithms – ESA 2004. 12th Annual European Symposium, Bergen, Norway, September 14-17, 2004*, volume 3221 of *Lect. Notes Comput. Sci.*, pages 544–555. Springer Berlin Heidelberg, 2004.
[26] G. Villard. On computing the resultant of generic bivariate polynomials. In *Proceedings of the 2018 ACM International Symposium on Symbolic and Algebraic Computation*, ISSAC '18, pages 391–398. New York, NY, USA, 2018. ACM.

On the Parallelization of Triangular Decompositions

Mohammadali Asadi
University of Western Ontario
London, Canada
masadi4@uwo.ca

Alexander Brandt
University of Western Ontario
London, Canada
abrandt5@uwo.ca

Robert H. C. Moir
University of Western Ontario
London, Canada
rmoir3@uwo.ca

Marc Moreno Maza
University of Western Ontario
London, Canada
moreno@csd.uwo.ca

Yuzhen Xie
University of Western Ontario
London, Canada
yuzhenxie@yahoo.ca

ABSTRACT

We discuss the parallelization of algorithms for solving polynomial systems by way of triangular decomposition. The Triangularize algorithm proceeds through incremental intersections of polynomials to produce different components (points, curves, surfaces, etc.) of the solution set. Independent components imply the opportunity for concurrency. This "component-level" parallelization of triangular decompositions, our focus here, belongs to the class of dynamic irregular parallelism. Potential parallel speed-up depends only on geometrical properties of the solution set (number of components, their dimensions and degrees); these algorithms do not scale with the number of processors. To manage the irregularities of component-level parallelization we combine different concurrency patterns, namely, workpile, producer-consumer, and fork/join. We report on our implementation in the freely available BPAS library. Experimentation with thousands of polynomial systems yield examples with up to 9.5× speed-up on a 12-core machine.

CCS CONCEPTS

• **Computing methodologies** → **Symbolic and algebraic manipulation**; **Parallel algorithms**; • **Mathematics of computing** → **Solvers**; **Mathematical software performance**.

KEYWORDS

polynomial system solving, parallel processing, triangular decomposition, fork-join model, producer-consumer problem, regular chains, dynamic irregular parallel applications

ACM Reference Format:
Mohammadali Asadi, Alexander Brandt, Robert H. C. Moir, Marc Moreno Maza, and Yuzhen Xie. 2020. On the Parallelization of Triangular Decompositions. In *International Symposium on Symbolic and Algebraic Computation (ISSAC '20), July 20–23, 2020, Kalamata, Greece.* ACM, New York, NY, USA, 8 pages. https://doi.org/10.1145/3373207.3404065

ISSAC '20, July 20–23, 2020, Kalamata, Greece

ACM ISBN 978-1-4503-7100-1/20/07…$15.00
https://doi.org/10.1145/3373207.3404065

1 INTRODUCTION

Solving a polynomial system by means of triangular decomposition entails computing a collection of regular chains which together encode the zero set of the input system. Where triangular decomposition proceeds incrementally, that is, by solving one equation after the other, a splitting of the quasi-component of a regular chain may be discovered when intersecting the next polynomial of the input system and the current partial solution. Concurrency is possible as the decomposition proceeds independently on each branch.

Parallelization of high-level procedures for algebraic and geometric computation is not new, receiving much attention in the '80s and '90s, for example see [2, 8, 9, 12, 22]. In recent years parallelization has again seen attention but in low-level operations like polynomial arithmetic [5, 13, 19], GCDs [14], and Factorization [20]. Parallelization in these low-level routines is more natural, being known as *regular parallelism* [18], since the task decomposes in a static way into consistently sized units of work. Taking advantage of the *irregular parallelism* in high-level geometric computations is more challenging, where splitting, and thus parallelism, is dependent only on the geometry of the input system, and must be found dynamically. For example, in the normalization algorithm of [6], components are first found serially and then processed by a *parallel map* over the components. In our proposed technique, we both discover components and process them in parallel. Indeed, finding splittings in the geometry is as difficult as solving the system itself.

Parallel triangular decomposition was first addressed in [21]. There, parallelism was facilitated by multi-processor shared memory and inter-process communication. The overhead associated with this parallel implementation is drastic and only suited for extremely large problems. It also relied on solving systems modulo a prime in order to generate extra splittings and provide opportunities for parallelism. Solving instead over the rationals provides less opportunity for parallelism but is of more practical importance.

Despite these challenges, we investigate opportunities for thread-level parallelism in triangular decomposition algorithms over the rational numbers. In particular, we discuss three different categories of concurrency to be exploited: (1) high-level parallelism via independent intersection tasks, (2) finer-grained parallelism by means of *asynchronous generators* between subroutines, and (3) a divide-and-conquer scheme for the removal of redundant components. The parallel schemes are independent but their implementations are designed to work cooperatively if needed. This is particularly

important to combat the work imbalance and inherent irregular parallelism of triangular decomposition algorithms. As we will discuss, we find that the use of generators in addition to a top-level parallelization scheme is an effective sort of dynamic load-balancing.

Our implementation is extensive, leading to 12 possible configurations of the Triangularize algorithm. This includes solving in the sense of Kalkbrener or Lazard and Wu, two organizations of the top-level Triangularize algorithm, and three different levels of parallelization. Our algorithms have been implemented in the C/C++ language and extensively evaluated using a collection of over 3000 polynomial systems. The results are encouraging, yielding up to 9.5× parallel speed-up on a 12-core machine.

We begin in Section 2 with a brief review of regular chain theory, the Triangularize algorithm, and parallel patterns. Section 3 examines opportunities for parallelism in Triangularize via those parallel patterns. We report on our implementation in Section 4. Finally, we conclude in Section 5 with discussion on experimental results, the effectiveness of our techniques, and areas for future work.

2 PRELIMINARIES

This section is a short review of concepts and algorithms for triangular decomposition and parallel programming. The first two sections deal with the former, for which details can be found in [10]. Throughout this paper, let $\mathbf{k}$ be a perfect field, $\mathbf{K}$ be its algebraic closure, and $\mathbf{k}[\underline{X}]$ be the polynomial ring with $\underline{X} = X_1 < \cdots < X_n$.

2.1 Regular chain theory

Let $p \in \mathbf{k}[\underline{X}]$. Assume that $p \notin \mathbf{k}$ holds. Denote by $\mathrm{mvar}(p)$, $\mathrm{init}(p)$, $\mathrm{mdeg}(p)$, and $\mathrm{tail}(p)$, respectively, the greatest variable appearing in p (called the *main variable* of p), the leading coefficient of p w.r.t. $\mathrm{mvar}(p)$ (called the *initial* of p), the degree of p w.r.t. $\mathrm{mvar}(p)$ (called the *main degree* of p) and the reductum of p w.r.t. $\mathrm{mvar}(p)$ (called the *tail* of p). For $F \subseteq \mathbf{k}[\underline{X}]$, we denote by $\langle F \rangle$ and $V(F)$ the ideal generated by F in $\mathbf{k}[\underline{X}]$ and the algebraic set of $\mathbf{K}^n$ consisting of the common roots of the polynomials of F, respectively.

Triangular set. Let $T \subseteq \mathbf{k}[\underline{X}]$ be a *triangular set*, that is, a set of non-constant polynomials with pairwise distinct main variables. Denote by $\mathrm{mvar}(T)$ the set of main variables of the polynomials in T. A variable $v \in \underline{X}$ is called *algebraic* w.r.t. T if $v \in \mathrm{mvar}(T)$, otherwise it is said *free* w.r.t. T. For $v \in \mathrm{mvar}(T)$, we denote by T_v and T_v^- (resp. T_v^+) the polynomial $f \in T$ with $\mathrm{mvar}(f) = v$ and the polynomials $f \in T$ with $\mathrm{mvar}(f) < v$ (resp. $\mathrm{mvar}(f) > v$). Let h_T be the product of the initials of the polynomials of T. We denote by $\mathrm{sat}(T)$ the *saturated ideal* of T: if $T = \emptyset$ holds, then $\mathrm{sat}(T)$ is defined as the trivial ideal $\langle 0 \rangle$, otherwise it is the ideal $\langle T \rangle : h_T^\infty$. The *quasi-component* $W(T)$ of T is defined as $V(T) \setminus V(h_T)$. For $f \in \mathbf{k}[\underline{X}]$, we define $Z(f, T) := V(f) \cap W(T)$. The Zariski closure of $W(T)$ in $\mathbf{K}^n$, denoted by $\overline{W(T)}$, is the intersection of all algebraic sets $V \subseteq \mathbf{K}^n$ such that $W(T) \subseteq V$ holds; moreover we have $\overline{W(T)} = V(\mathrm{sat}(T))$.

Regular chain. A triangular set $T \subseteq \mathbf{k}[\underline{X}]$ is a *regular chain* if either T is empty, or letting v be the largest variable occurring in T, the set T_v^- is a regular chain, and the initial of T_v is regular (that is, neither zero nor zero divisor) modulo $\mathrm{sat}(T_v^-)$. The *dimension* of T, denoted by $\dim(T)$, is by definition the dimension of its saturated ideal and, as a property, equals $n - |T|$, where $|T|$ is the number of elements of T. The saturated ideal $\mathrm{sat}(T)$ of the regular chain T enjoys important properties, in particular the following, proved in [7]. Let $u_1, \ldots, u_d$ be all the free variables of T. Then $\mathrm{sat}(T)$ is unmixed of dimension d. Moreover, we have $\mathrm{sat}(T) \cap \mathbf{k}[u_1, \ldots, u_d] = \langle 0 \rangle$. Another property is the fact that a polynomial p belongs to $\mathrm{sat}(T)$ if and only if p reduces to 0 by pseudo-division w.r.t T, see [3].

Regular GCD. Let i be an integer with $1 \leq i \leq n$, $T \subseteq \mathbf{k}[\underline{X}]$ be a regular chain, $p, t \in \mathbf{k}[\underline{X}] \setminus \mathbf{k}$ be polynomials with the same main variable X_i, and $g \in \mathbf{k}$ or $g \in \mathbf{k}[\underline{X}]$ with $\mathrm{mvar}(g) \leq X_i$. Assume that

(1) $X_i > X_j$ holds for all $X_j \in \mathrm{mvar}(T)$, and
(2) both $\mathrm{init}(p)$ and $\mathrm{init}(t)$ are regular w.r.t. $\mathrm{sat}(T)$.

Denote by $\mathbb{A}$ the total ring of fractions of the residue class ring $\mathbf{k}[X_1, \ldots, X_{i-1}]/\sqrt{\mathrm{sat}(T)}$. Note that $\mathbb{A}$ is isomorphic to a direct product of fields. We say that g is a *regular GCD* of p, t w.r.t. T whenever:

(G_1) the leading coefficient of g in X_i is a regular element of $\mathbb{A}$;
(G_2) g belongs to the ideal generated by p and t in $\mathbb{A}[X_i]$; and
(G_3) if $\deg(g, X_i) > 0$, then g pseudo-divides both p and t in $\mathbb{A}[X_i]$, that is, both $\mathrm{prem}(p, g)$ and $\mathrm{prem}(t, g)$ belong to $\sqrt{\mathrm{sat}(T)}$.

When Conditions (G_1), (G_2), (G_3) and $\deg(g, X_i) > 0$ hold, we have:

(G_4) if $\mathrm{mdeg}(g) = \mathrm{mdeg}(t)$, then $\sqrt{\mathrm{sat}(T \cup t)} = \sqrt{\mathrm{sat}(T \cup g)}$ and $W(T \cup t) \subseteq Z(h_g, T \cup t) \cup W(T \cup g) \subseteq \overline{W(T \cup t)}$ both hold,
(G_5) if $\mathrm{mdeg}(g) < \mathrm{mdeg}(t)$, let $q = \mathrm{pquo}(t, g)$, then $T \cup q$ is a regular chain and the following two relations hold:
  (a) $\sqrt{\mathrm{sat}(T \cup t)} = \sqrt{\mathrm{sat}(T \cup g)} \cap \sqrt{\mathrm{sat}(T \cup q)}$,
  (b) $W(T \cup t) \subseteq Z(h_g, T \cup t) \cup W(T \cup g) \cup W(T \cup q) \subseteq \overline{W(T \cup t)}$,
(G_6) $W(T \cup g) \subseteq V(p)$,
(G_7) $V(p) \cap \overline{W(T \cup t)} \subseteq W(T \cup g) \cup V(p, h_g) \cap W(T \cup t) \subseteq V(p) \cap \overline{W(T \cup t)}$.

Triangular decomposition. Let $F \subseteq \mathbf{k}[\underline{X}]$. Regular chains $T_1, \ldots, T_e$ of $\mathbf{k}[\underline{X}]$ form a *triangular decomposition* of $V(F)$ in the sense of Kalkbrener (resp. Wu and Lazard) whenever we have $V(F) = \cup_{i=1}^e \overline{W(T_i)}$ (resp. $V(F) = \cup_{i=1}^e W(T_i)$). Hence, a triangular decomposition of $V(F)$ in the sense of Wu and Lazard is necessarily a triangular decomposition of $V(F)$ in the sense of Kalkbrener, while the converse is not true. One important issue in the implementation of algorithms decomposing polynomial ideals and algebraic sets is the removal of redundant components. In the context of triangular decompositions, this issue implies being able to decide whether $W(T_i) \subseteq W(T_j)$ holds or not, for any two regular chains $T_i, T_j \subseteq \mathbf{k}[\underline{X}]$.

2.2 Specification of the main algorithms

Triangularize. Let $F \subseteq \mathbf{k}[\underline{X}]$. The function call Triangularize(F) computes regular chains $T_1, \ldots, T_e \subseteq \mathbf{k}[\underline{X}]$ forming a triangular decomposition of $V(F)$ in the sense of either Kalkbrener, or Wu and Lazard. An algorithm for Triangularize(F) is presented in [10].

Regularize. For $p \in \mathbf{k}[\underline{X}]$ and $T \subseteq \mathbf{k}[\underline{X}]$ a regular chain, Regularize(p, T) computes regular chains $T_1, \ldots, T_e \subseteq \mathbf{k}[\underline{X}]$ such that:

(R_1) for $i = 1, \ldots, e$, either $p \in \mathrm{sat}(T_i)$ or p is regular w.r.t. $\mathrm{sat}(T_i)$,
(R_2) we have $W(T) \subseteq W(T_1) \cup \cdots \cup W(T_e) \subseteq \overline{W(T)}$.

RegularGcd. Let i, T, p, t, g be as above in the definition of a regular GCD. The function call RegularGcd(p, t, X_i, T) computes a set of pairs $\{(g_1, T_1), \ldots, (g_e, T_e)\}$ such that:

(1) for $i = 1, \ldots, e$, if $\dim(T_i) = \dim(T)$ holds, then g_i is a regular GCD of p, t w.r.t. T_i,
(2) we have $W(T) \subseteq W(T_1) \cup \cdots \cup W(T_e) \subseteq \overline{W(T)}$.

Intersect. Let $p \in \mathbf{k}[\underline{X}]$ and let $T \subseteq \mathbf{k}[\underline{X}]$ be a regular chain. The function call Intersect(p, T) computes regular chains $T_1, \ldots, T_e \subseteq \mathbf{k}[\underline{X}]$ such that: $V(p) \cap W(T) \subseteq W(T_1) \cup \cdots \cup W(T_e) \subseteq V(p) \cap \overline{W(T)}$.

2.3 Parallel Programming Patterns

The algorithms shown in the previous subsection already hint at their parallel opportunities where each either take a list as an argument or return a list. These opportunities are explained in detail in Section 3, while here we review some parallel programming patterns which will be employed by those opportunities.

The first key observation is that the Triangularize algorithm itself only presents parallelism when the solution set can be separated into multiple components. The existence of such components is not an algorithmic property but rather one subject to the system being solved. Even if computations do split, the work is likely to be unbalanced. This describes *irregular parallelism*. In contrast, *regular parallelism* exists where problems algorithmically decompose into sub-problems of roughly equal size. Despite these challenges, parallel patterns can be employed for irregular parallelism [18].

We are concerned with *thread parallelism*, and thus with minimizing *parallel overheads*, as well as effectively managing inter-thread dependencies and communication. The former deals with the cost of spawning threads, and *over-subscription*—where software threads outnumber hardware resources to drastically reduce performance. The latter can be addressed through parallel design patterns [18].

Parallel Map and Workpile. The *map* pattern maps a function to each item in a collection, simultaneously executing the function on each independent data, scaling well with increasing data and threads. But, threads must operate in lockstep, and are thus limited by the slowest in the group, working best with regular parallelism.

The *workpile* pattern generalizes map to handle both irregular amounts of work and an unknown number of tasks for *load-balancing*. Tasks are collected into a pile (or queue) and one thread executes one task from the pile in parallel, repeating until the pile is empty. This pattern allows in-flight tasks to add additional tasks to the pile, allowing new tasks to be launched immediately by an idle thread. Threads are thus uncoupled, making load balancing possible. Tasks in the pile may also be ordered so that tasks can create new tasks earlier in the computation to exploit further parallelism.

Asynchronous Generators, Producer-Consumer, and Pipeline. A *generator* function is one yielding data items one a time rather than many as a collection. Concurrency arises if items are generated asynchronously while the caller processes a generated item; hence an *asynchronous generator*. This yields the classic *producer-consumer problem* (see [4, Ch. 6]). Using a collection of producer-consumer pairs in a sequence (or directed acyclic graph), where interior nodes are both producers and consumers, is one way of describing the *pipeline* pattern. Pipeline's greatest asset is its ability to begin processing before all input data items are ready (cf. the map pattern). If the producer-consumer pairs are implemented using generators, one can construct a tree—rather than a strict pipeline— which dynamically grows and shrinks as functions are called and return. A tree arises where a producer consumes multiple generators.

Divide-and-Conquer and Fork-Join. Divide-and-Conquer (DnC) is a ubiquitous algorithmic technique based on recursion. A problem is divided into sub-problems, each solved (conquered) recursively, and then sub-solutions are combined to provide a solution to the original problem. Where there are multiple recursive calls per level, the *fork-join* pattern can be employed where each recursive branch is executed in parallel (forked) and then joined together before returning. In a parallel DnC it is important to avoid too many parallel recursive calls to reduce parallel overheads and over-subscription.

3 CONCURRENCY OPPORTUNITIES

In this section, we highlight the opportunities for concurrent execution offered by the algorithms for computing triangular decompositions presented in [10]. To do so, we review the key ideas underlying those algorithms and show how concurrency can be exposed.

3.1 Parallel Map and the Triangularize procedure

Algorithm 1 states a simple procedure implementing the Triangularize procedure. Lines 1 to 5 in Algorithm 1 compute a triangular decomposition of $V(F)$ in the sense of Wu and Lazard; this follows easily from the specification of the Intersect algorithm given in Section 2. Line 5 ensures the decomposition is free of redundant components; we shall discuss this step in detail in Section 3.3.

One can organize the regular chains computed in the loop of Algorithm 1 as a tree with an edge going from node T to node T' if T' is returned by Intersect(p, T) for some $p \in F$, (e.g., see Figure 2 in Section 5). Let (T, T') be such an edge: observe that we have $|T| \leq |T'|$. Algorithm 1 traverses this tree in a breadth-first search manner. Using this algorithm, a Kalkbrener decomposition can be computed by simply pruning branches of the tree, for which the *height* of a regular chain, i.e., the number of polynomials in the chain, exceeds the number of input polynomials in F. This is a consequence of *Krull's height theorem*, see [10], for details.

Algorithm 1 Triangularize(F)

Input: a finite set $F \subseteq \mathbf{k}[\underline{X}]$
Output: regular chains $T_1, \ldots, T_e \subseteq \mathbf{k}[\underline{X}]$ such that $V(F) = W(T_1) \cup \cdots \cup W(T_e)$
1: $\mathcal{T} := \{\emptyset\}$
2: **for** $p \in F$ **do**
3: $\quad \mathcal{T}' := \bigcup_{T \in \mathcal{T}}$ Intersect(p, T)
4: $\quad \mathcal{T} := \mathcal{T}'$
5: Remove from $\mathcal{T}$ any T_1 where there exists $T_2 \in \mathcal{T}$ such that $W(T_1) \subseteq W(T_2)$ and $T_1 \neq T_2$ both hold.
6: **return** $\mathcal{T}$

It follows from Algorithm 1 that whenever Intersect(p, T) returns more than one regular chain, there is an opportunity for concurrent execution. Indeed, the branches of the breadth-first search are independent and can be continued concurrently. Referring to the celebrated parallel map pattern [18, Ch. 4], one can see Line 3 as a map-step where Intersect maps each current regular chain. Moreover, this can be seen as coarse-grained parallelism as each Intersect call represents substantial work. We now turn our attention to parallel opportunities in the core subroutines of Triangularize.

3.2 Asynchronous generators with Intersect, RegularGcd and Regularize

Let $p \in \mathbf{k}[\underline{X}]$ and $T \subseteq \mathbf{k}[\underline{X}]$ be a regular chain. The operation Intersect(p, T) is quite complicated in general. Yet, for the purpose

of discussing opportunities concurrency, it is sufficient to consider the most common scenario. Let us assume $p \notin \mathbf{k}$, $v = \text{mvar}(p)$, init(p) is regular w.r.t sat(T) (calling Regularize can assure this), and that T_v^+ is empty (by proceeding by induction on the number of variables). Algorithm 2 implements Intersect(p, T) under these conditions. It follows from the applications of Formulas (G_6) and (G_7) from Section 2 together with induction on $\dim(T_v^-)$.

Algorithm 2 Intersect(p, T)

Input: $p \in \mathbf{k}[\underline{X}]$, $p \notin \mathbf{k}$, $v := \text{mvar}(p)$, a regular chain $T \subseteq \mathbf{k}[\underline{X}]$ such that init(p) regular w.r.t. sat(T) and $T_v^+ = \emptyset$.
Output: regular chains $T_1, \ldots, T_e \subseteq \mathbf{k}[\underline{X}]$ such that $V(p) \cap W(T) \subseteq W(T_1) \cup \cdots \cup W(T_e) \subseteq V(p) \cap \overline{W(T)}$.

```
1:  if v ∉ mvar(T) then
2:      yield T ∪ {p}
3:      for S in Intersect(init(p), T) do
4:          for U in Intersect(tail(p), S) do
5:              yield U
6:  else
7:      for (g_i, T_i) ∈ RegularGcd(p, T_v, v, T_v^-) do
8:          if dim(T_i) ≠ dim(T_v^-) then
9:              for T_{i,j} ∈ Intersect(p, T_i) do
10:                 yield T_{i,j}
11:         else
12:             if g_i ∉ k and mvar(g_i) = v then
13:                 yield T_i ∪ {g_i}
14:             for T_{i,j} ∈ Intersect(lc(g_i, v), T_i) do
15:                 for T_{i,j,k} ∈ Intersect(p, T_{i,j}) do
16:                     yield T_{i,j,k}
```

Note that Algorithm 2 is a *generator function*, also called an *iterator*, a special type of co-routine, see Chapter 8 in [23]. In our pseudo-code, the keyword **yield** outputs a value to the generator's caller and then resumes execution. In contrast, **return** is used to return a value and terminate the function. Each yield in Intersect is an opportunity for concurrency where the caller may execute in parallel with the one yielded regular chain, meanwhile Intersect continues. Hence, Intersect may be implemented as a so-called *asynchronous generator*, a concept described in Section 2.3.

Observe now that the function call RegularGcd(p, T_v, v, T_v^-), when it returns more than one pair, provides additional opportunities for concurrency. Let us actually see how this latter function call is performed in [10]. The subresultant chain S of p and T_v, regarded as univariate in v, is computed. Let $\lambda = \min(\text{mdeg}(p), \text{mdeg}(T_v))$ and let i be in the range $0, \ldots, \lambda + 1$. Denote by S_i the subresultant (from S) of index i and by s_i the principal subresultant coefficient of S_i. Recall that we have $S_{\lambda+1} = p$, $S_\lambda = T_v$, and that S_0 is simply the resultant of p and T_v in v; moreover, we have $S_0 = s_0$. Let j be an integer, with $1 \leq j \leq \lambda + 1$, such that s_j is a regular modulo sat(T_v^-) and such that for any $0 \leq i < j$, we have $s_i \in \text{sat}(T_v^-)$. Then S_j is a regular GCD of p and T_v w.r.t. T_v^-. By calling Regularize on s_k, $k = 0, \ldots, j$ it is always possible to find such an S_j, up to splittings of the regular chain. This again suggests that RegularGCD could be implemented as an asynchronous generator for Intersect.

We now consider Regularize, focusing on the most common scenario as with Intersect. Algorithm 3 presents this case, stating the assumptions which follow from Formulas (G_4) and (G_5) in Section 2, together with a reasoning by induction on $\dim(T_v^-)$. Just as in the previous two algorithms, Regularize may both be implemented as an asynchronous generator and use generators as it calls Intersect.

Algorithm 3 Regularize(p, T)

Input: $p \in \mathbf{k}[\underline{X}]$, $p \notin \mathbf{k}$, $v := \text{mvar}(p)$, a regular chain $T \subseteq \mathbf{k}[\underline{X}]$ such that init(p) regular w.r.t. sat(T_v^-) and $T_v^+ = \emptyset$.
Output: regular chains $T_1, \ldots, T_e \subseteq \mathbf{k}[\underline{X}]$ such that (R_1), (R_2) hold.

```
1:  if v ∉ mvar(T) then return T
2:  for (g_i, T_i) ∈ RegularGcd(p, T_v, v, T_v^-) do
3:      if dim(T_i) < dim(T_v^-) then
4:          for T_{i,j} ∈ Regularize(p, T_i) do
5:              yield T_{i,j}
6:      else
7:          if g_i ∈ k or mvar(g_i) < v or mdeg(g_i) = mdeg(T_v) then
8:              yield T_i
9:          else
10:             yield T_i ∪ {g_i}
11:             q_i := pquo(T_v, g_i, v)
12:             yield T_i ∪ {q_i}
13:             for T_{i,j} ∈ Intersect(lc(g_i, v), T_i) do
14:                 for T_{i,j,k} ∈ Regularize(p, T_{i,j}) do
15:                     yield T_{i,j,k}
```

The above discussion of Intersect, RegularGcd and Regularize shows that each of those routines can be implemented as a generator function. Each top-level call to Intersect thus creates a tree of generator function calls, most of which being both producers and consumers of values. This hints towards using the *producer-consumer* and *pipeline* patterns, as discussed in Section 2.3.

These concurrency opportunities represent more fine-grained parallelism as the amount of work diminishes with each recursive call. Further, it is worth noting that the work between splittings is likely unbalanced. For instance, the polynomials g_i and q_i, returned with the regular chain T_i at Lines 10 and 12 of Algorithm 3, may have very different degrees. These irregular parallelism challenges are addressed through cooperation between the generators and the coarse-grained parallelism in Triangularize (see Section 4.3).

3.3 Fork-join approach for removing redundant components

To remove redundant components efficiently we must address two issues: how to efficiently test single inclusions, e.g. $W(T_i) \subseteq W(T_j)$ and how to efficiently remove redundant components from a large set. The first issue is addressed by taking advantage of the heuristic algorithm IsNotIncluded (see [24, pp. 166–169]) which is very effective in practice. Handling large sets of regular chains is possible by structuring the computation as a divide-and-conquer algorithm.

Given a set $\mathcal{T} = \{T_1, \ldots, T_e\}$ of regular chains, Algorithm 4, RemoveRedundantComponents($\mathcal{T}$), removes redundant chains by dividing $\mathcal{T}$ into two subsets, producing two irredundant sets by means of recursion. Then, the two sets are merged by checking for pair-wise inclusions between the two sets. The divide-and-conquer nature of RemoveRedundantComponents is undoubtedly admissible to ubiquitous fork-join parallelism, Particularly, one *forks* the computation to compute one of the recursive calls in parallel, and then *joins* upon return. These are indicated by the keywords **spawn** and **sync**, respectively. The merge-step is also embarrassingly parallel and can use the map pattern for each of the inner loops.

Algorithm 4 RemoveRedundantComponents($\mathcal{T}$)

Input: a finite set $\mathcal{T} = \{T_1, \ldots, T_e\}$ of regular chains
Output: a set of regular chains forming an irredudant decomposition of the same algebraic set as $\mathcal{T}$

```
if e = 1 then return {T_1}
ℓ := ⌈e/2⌉; T_≤ℓ := {T_1, ..., T_ℓ}; T_>ℓ := {T_ℓ+1, ..., T_e}
T'_1 := spawn RemoveRedundantComponents(T_≤ℓ)
T'_2 := RemoveRedundantComponents(T_>ℓ)
sync
for T_1 in T'_1 do
    if ∀T_2 in T'_2 IsNotIncluded (T_1, T_2) then
        T''_1 := T''_1 ∪ {T_1}
for T_2 in T'_2 do
    if ∀T_1 in T''_1 IsNotIncluded (T_2, T_1) then
        T''_2 := T''_2 ∪ {T_2}
return T''_1 ∪ T''_2
```

4 IMPLEMENTATION

Our implementation of regular chains and the Triangularize algorithm follows that of [10], hence, here we look only at the implementation of parallel aspects. Our implementation is written in the C and C++ languages. For the simple fork-join parallelism in the removal of redundant components, we simply use Cilk [15], built-in to the GCC compiler, mirroring Algorithm 4 and requires no additional explanation. In all other cases we implement our own parallel constructs using the Thread Support Library of C++11. Our implementation is freely available in source as part of the Basic Polynomial Algebra Subprograms (BPAS) library [1] at www.bpaslib.org. We begin by describing two reorganizations of the Triangularize algorithm and then describe the underlying parallelization.

The first reorganization of Triangularize is "by level". It is simple restructuring of Algorithm 1 to move the removal of redundant components inside the loop, thus removing redundancies after each incremental step ("level"). We apply the map pattern to the inner for loop, thus spawning $|\mathcal{T}| - 1$ additional threads to execute the $|\mathcal{T}|$ independent calls to Intersect. As previously described for the map pattern, if the intersections at a particular level are unbalanced then the program must wait for the slowest, reducing parallelism.

The second reorganization of Triangularize is "by tasks" (Algorithm 5), and makes use of the workpile pattern to combat the issues incurred by applying the map pattern to irregular parallelism. Here, we essentially invert the loops of Triangularize to first iterate over the current collection of regular chains and then iterate over polynomials in the input system. Since the former is actually of a variable and unknown size, this is achieved by creating tasks, one per regular chain, with a list of polynomials associated to each task. Splittings create new tasks to be added to the work pile. Once a task has finished intersecting its list of polynomials, it is complete and added to a list of results. In this scheme, the potential parallelism is greater than TriangularizeByLevel but the potential amount of work is also greater since redundancies are no longer removed at each step. We discuss these differences later in Section 5.

4.1 Executor Thread Pool

A fundamental structure of most parallel systems is a thread pool. Thread pools maintain a collection of long-running threads which wait to be given a task, execute that task, and then return to the pool. This avoids the overhead of repeatedly spawning threads and limits the number of threads to avoid over-subscription. When tasks outnumber threads the pool also maintains a queue of tasks.

Algorithm 5 TriangularizeByTasks(F)

Input: a finite set $F \subseteq \mathbf{k}[\underline{X}]$
Output: regular chains $T_1, \ldots, T_e \subseteq \mathbf{k}[\underline{X}]$ such that $V(F) = W(T_1) \cup \cdots \cup W(T_e)$

```
1: Tasks := { (F, ∅) }; T := {}
2: while |Tasks| > 0 do
3:     (P, T) := pop a task from Tasks
4:     Choose a polynomial p ∈ P; P' := P \ {p}
5:     for T' in Intersect(p, T) do
6:         if |P'| = 0 then T := T ∪ {T'}
7:         else Tasks := Tasks ∪ {(P', T')}
8: return RemoveRedundantComponents(T)
```

Often threads in a pool execute a predetermined task. However, the many different subroutines in the Triangularize algorithm all need attention. We thus make use of *functors* (e.g. `std::function`), objects which encapsulate a function as a first-class object, to model generic tasks. Our `ExecutorThreadPool` then maintains a queue of functor tasks and a pool of `ExecutorThreads`, threads capable of executing any functor. For genericity, our implementation requires `void` functors, hence returning values by reference. Moreover, values can be returned one at a time if the functor is a generator.

4.2 Asynchronous Generators & Object Streams

Following the object-oriented nature of C++, and much like functor objects, we look to encapsulate generators as objects, providing a generic interface for generators producing many different kinds of objects. We have created a generic `AsyncGenerator` class, where objects are created very simply by passing it a functor whose underlying function creates a collection of objects. The caller then requests data from the generator object itself instead of the functor.

Serially, a generator object could be implemented by collecting the objects returned by the functor in a queue and yielding them one at a time to the caller. To achieve parallelism the `AsyncGenerator` facilitates the producer-consumer pattern; the caller is the consumer, the functor is the producer, and the `AsyncGenerator` itself works as the intermediary and common interface between the two. The interface of `AsyncGenerator` can be seen in Listing 1.

In practice, the subroutines like Intersect, RegularGCD and Regularize, are mutually recursive and simultaneously act as both producers and consumers, using multiple `AsyncGenerator` objects.

The `AsyncGenerator` fulfills producer-consumer by first inserting itself as a parameter to the functor, so that the producer has a handle on the generator object, and then invokes the functor as follows. The generator requests a thread from the `ExecutorThreadPool` and, where one is available, asynchronously executes the functor on that thread. Otherwise, the generator acts serially, as just explained, maintaining a queue of objects returned by the functor.

The final detail to the `AsyncGenerator` is a mechanism to sleep the consumer when no object is available to consume. The solution is provided generically as the `AsyncObjectStream` class which provides a thread-safe queue interface and an internal mechanism to sleep the consumer until an object is ready to be consumed (i.e. a condition variable or semaphore, see [4, Ch. 6]). Ultimately, the

```
template <class Object>
class AsyncGenerator {

  /* CONSUMER: create generator to encapsulate a function call. */
  template<class Function, class... Args>
  AsyncGenerator(Function&& f, Args&&... args);

  /* PRODUCER: Add a new Object to be retrieved later. */
  virtual void generateObject(Object& obj) = 0;

  /* PRODUCER: Finalize the AsyncGenerator by declaring it has
  finished generating all possible objects. */
  virtual void setComplete() = 0;

  /* CONSUMER: Obtain the next generated Object by reference.
  returns false iff no more objects available and setComplete() */
  virtual bool getNextObject(Object& obj) = 0;
};
```

Listing 1: The `AsyncGenerator` interface which implements an asynchronous producer-consumer pattern.

object stream is completely encapsulated by the `AsyncGenerator` and may or may not be used depending on if the generator is truly executing asynchronously.

4.3 A "Cooperative" Task Scheduler

A task scheduler is one possible implementation of the workpile pattern. We look to facilitate the scheduling of an unknown number tasks where the tasks themselves can produce more tasks. This is exactly the case as in TriangularizeByTasks. Our `TaskScheduler`, much like the `ExecutorThreadPool`, encapsulates tasks as a queue of functors, where each functor has a reference to the scheduler in order to schedule more tasks.

The scheduler internally makes use of an `ExecutorThreadPool` to launch new tasks immediately (if the pool is not empty) and otherwise add it to the queue of tasks. To reap the most benefits of workpile, active threads should add tasks to the scheduler as early as possible to expose more parallelism. For tasks which produce exactly one task, instead of adding the new task to the scheduler, the producing task should execute the new task directly in order to avoid synchronization overhead.

Consider also that we want to simultaneously use generators and a task scheduler. However, they both use an `ExecutorThreadPool`, which may lead to too many active threads and thus resource contention. One solution would be to limit the sum of the number of threads in both pools to be the number of hardware threads. However, this static solution is not receptive to dynamic load-balancing. In particular, if the number of scheduler tasks is low then more threads should be made available to generators, or vice-versa. This leads to a "cooperative" task scheduler and generators.

The cooperation begins by sharing a single thread pool between the `TaskScheduler` and all `AsyncGenerators`. However, this alone is not enough. The tasks to be scheduled (i.e. the calls to Intersect from Triangularize) represent a larger amount of work than any subroutine generator. Hence, to support more coarse-grained parallelism and less parallel overhead, the scheduler should dynamically be given more resources, as needed. Simply stated, the thread pool creates a new "high priority" thread if the thread pool is empty when a new task becomes available. Although the number of active threads may now exceed hardware resources, this is only temporary until a generator finishes. As threads are returned to the thread pool, they may be terminated to account for new high priority threads, keeping the total number of threads within hardware limits. To avoid a runaway task scheduler, the pool limits the number of priority threads that can be created. Naturally, this pattern also works in reverse, when there are few tasks, as more threads will be available in the pool to support more asynchronous generators.

5 EXPERIMENTATION AND DISCUSSION

The preceding two sections have explored various opportunities for parallelism within triangular decomposition algorithms and their implementations. In particular, we have described coarse-grained parallelism where Triangularize calls Intersect in parallel, and the more fine-grained parallelism brought by generators. We now look to evaluate the effectiveness of the different *configurations* of Triangularize. A configuration is parameterized by (1) the type of decomposition being computed (Lazard-Wu or Kalkbrener), (2) the organization of the top-level algorithm (TriangularizeByLevel or TriangularizeByTasks), and (3) the level of parallelization employed (serial, coarse-grained, or coarse- and fine-grained together).

We test the 12 possible configurations of our implementation by considering a suite of over 3000 real-world polynomial systems coming from the scientific literature as well as user-data and bug reports provided by MapleSoft and the RegularChains [16] library. In particular, we look at non-trivial systems taking at least 100ms to solve, in order to warrant the overhead of parallelism. This yields 828 systems. Our results herein are a median of 3 trials and were collected on a node running Ubuntu 14.04.5 with two Intel Xeon X5650 processors each with 6 cores (12 physical threads with hyperthreading) at 2.67 GHz, and a 12x4GB DDR3 memory configuration at 1.33 GHz.

On our test suite, speed-ups of up to 9.5× were found. Further, 824 of the 828 systems saw at least some parallel gains from at least one of the parallel configurations, with 133 having at least 2.0× speed-up. Considering that 203 of those systems contain only a single component, and thus have no potential for parallelism, implies that our implementation limits parallel overheads well. Table 1 presents some examples from the literature with timings and speed-up factors for all 12 configurations. We also compare timings there against the RegularChains package of Maple 2019.

Figure 1 summarizes the data collected for Kalkbrener decomposition as a two-dimensional histogram (the trends are the same for Lazard-Wu decomposition). For each subplot, the x-axis is serial execution time while the y-axis is parallel speed-up factor. It may appear that the task-based method incurs some slow-downs, but, this is mainly for cases running in less than 1s. There, some parallel overhead is expected, particularly as 203 systems have no potential parallelism. Nonetheless, if we consider more substantial examples—those requiring at least 1s to solve—then only 9 examples in the *Kalkbrener-Tasks-Fine* configuration have a speed-up less than 0.9, with the minimum being 0.84. From the trends in the data, we make two observations: (i) TriangularizeByTasks has the potential for higher parallelism than TriangularizeByLevel, and (ii) TriangularizeByLevel is, in general, slowed down by the use of generators while the performance TriangularizeByTasks has improved performance.

From our discussions in the preceding sections, (i) should be very apparent. Our task scheduler, built using the workpile parallel pattern is more receptive to the irregular parallelism present in

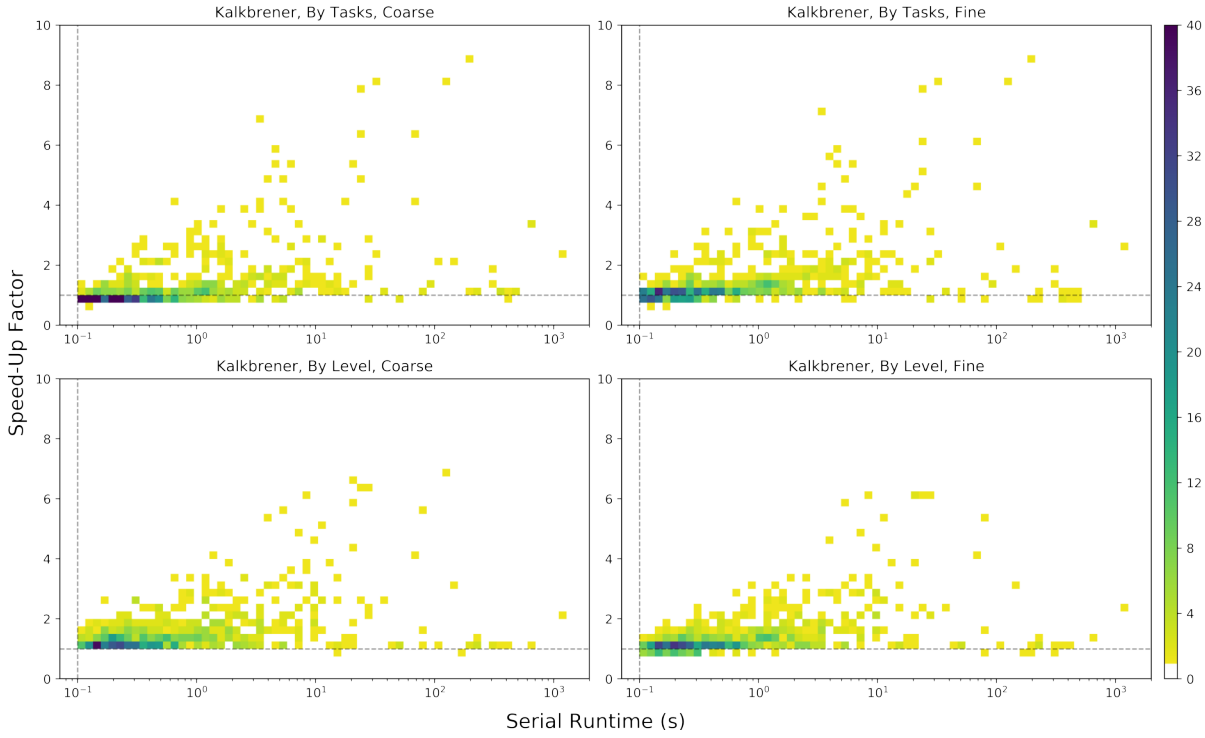


Figure 1: Histograms illustrating the distribution of serial runtime against speed-up factor for Kalkbrener decompositions.

triangular decomposition. Specifically, it allows new tasks to be taken up immediately by the scheduler's threads. This contrasts with the level-wise scheme where threads must operate in lockstep for each map step. Note (ii) also follows from our discussion of over-subscription and resource contention. The task scheduler was implemented to specifically cooperate with the generators, their interplay acting as a sort of dynamic load balancing depending on the number of components discovered during the decomposition. On the other hand, the rather naïve implementation of the map pattern used by TriangularizeByLevel created resource contention with the generator threads and worsened parallel performance compared to using map alone. While TriangularizeByLevel was the weaker performer in terms of parallel speed-up, there still exists examples where intermediate removal of redundant components is an important optimization step (e.g. W41 in Table 1, where TriangularizeByLevel is twice as fast). One should not forgo intermediately removing redundant components just for the parallel benefits of workpile. Indeed, combining redundant component removal alongside the task scheduler is an important area for future work.

Lastly, we consider two specific examples, Systems 2691 and 3295, illustrated as trees in Figure 2. These plots show the evolution of the decomposition as Intersect is called in parallel on independent components in the *Kalkbrener-Tasks-Fine* configuration. Two typical patterns are shown. For 2691, each call to Intersect creates two components. The dynamic and irregular nature of the parallelism is highlighted where each branch is discovered at different times, with each having different workloads. For 3295, the first component splits into several relatively equal branches, except for one branch

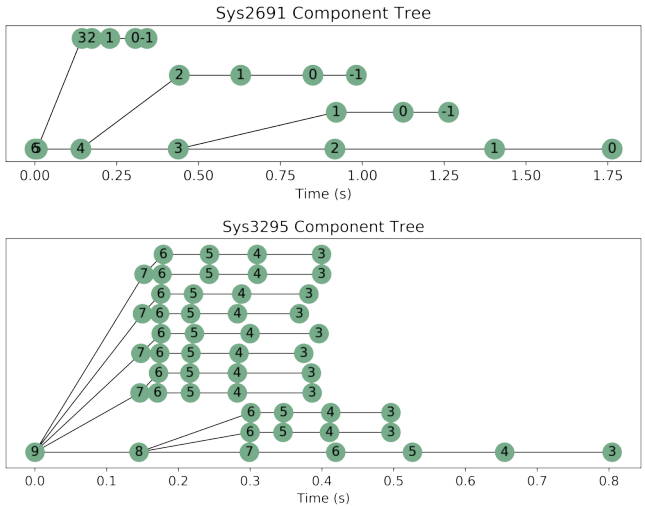


Figure 2: The *component tree* of two systems, showing that components and independent branches of computation are found dynamically during the decomposition. Each node depicts a component, and the node's label is the component's dimension (-1 being the empty set). Edges are drawn where a call to Intersect **on the parent component returned the child.**

with considerably more work. Despite the overall high number of components, computations in all branches overlap only briefly since the split in the bottom branch is not found until later, further highlighting the irregularity in parallelism.

System Name	Kalkbrener Decomposition							Lazard-Wu Decomposition						
	Level			Tasks			Maple 19	Level			Tasks			Maple 19
	S. Time	C	C+F	S. Time	C	C+F		S. Time	C	C+F	S. Time	C	C+F	
8-3-config-Li	8.188	3.81	3.14	8.275	2.80	2.84	5.836	36.299	2.93	2.89	38.825	3.07	3.06	26.660
dessin-2	46.974	1.12	1.08	37.090	1.11	1.11	126.008	50.185	1.19	1.14	36.557	1.11	1.07	125.276
dgp6	80.134	5.66	5.47	69.147	6.31	6.06	54.368	78.067	4.02	3.98	67.460	6.12	5.72	49.496
Ducos-7-5	49.001	1.04	1.02	49.542	0.99	1.00	1520.692	48.136	1.00	0.97	50.151	1.03	1.01	1537.293
Gerdt	3.939	3.01	2.88	3.259	3.91	4.57	0.932	3.755	2.65	2.62	3.172	4.03	4.28	0.952
Gonnet	1.406	2.42	2.43	1.683	2.57	2.46	1.924	1.399	2.26	2.09	1.680	2.25	2.77	1.984
Hereman-2	1.178	2.56	2.24	1.117	2.24	2.43	0.472	1.248	2.52	2.30	1.145	2.57	2.32	0.592
Issac97	3502.410	1.40	1.40	311.231	1.34	1.32	445.312	3571.450	1.40	1.42	318.640	1.37	1.34	450.880
Leykin-1	7.194	2.48	2.20	6.376	1.77	2.13	3.316	10.043	2.29	2.10	9.053	1.96	1.85	5.424
lhlp3	0.254	1.39	1.24	0.247	1.09	1.21	0.016	0.193	0.98	0.94	0.192	0.95	0.92	0.016
MacLane	1.137	2.04	1.74	1.170	1.92	1.71	1.748	3.816	1.50	1.42	4.042	1.49	1.41	4.828
MontesS16	2.233	2.33	2.11	2.650	2.30	2.32	2.400	2.177	2.11	2.09	2.592	2.40	2.30	2.488
Pappus	1.839	2.13	1.59	1.940	2.34	1.77	2.704	6.255	2.95	2.30	10.671	3.53	2.87	16.312
Pavelle	1.178	1.62	1.44	1.165	1.28	1.43	0.259	33.179	1.02	1.21	39.707	1.15	1.39	5.352
Reif	20.547	4.33	3.91	20.382	5.34	4.58	10.899	18.465	3.77	3.60	18.703	5.00	4.23	10.691
SEIT	0.593	1.89	1.49	0.635	1.69	1.54	0.448	4.275	3.14	2.48	4.606	3.07	2.64	4.368
W1	10.137	3.08	2.93	10.525	2.79	3.26	2.304	9.806	2.81	2.73	10.160	2.88	3.08	2.500
W41	12.960	3.94	3.80	25.266	4.82	5.16	18.644	12.645	3.65	3.57	24.757	4.86	4.72	15.892
W44	5.294	3.44	3.23	4.251	4.92	5.64	1.132	5.226	3.10	3.01	4.175	4.92	5.11	1.184
W5	20.247	5.98	6.20	22.435	6.38	6.15	14.260	20.248	5.90	6.16	21.951	6.19	6.07	13.180
YangBaxterRosso	0.675	2.42	2.17	0.674	4.03	4.18	0.348	0.638	2.18	1.98	0.663	4.32	4.13	0.371

Table 1: A comparison of timings for the 12 configurations of Triangularize**. Here, serial timings are given along with speed-up factors for coarse (C) and coarse and fine (C+F). Timings for solving using** RegularChains **in Maple 2019 are also included.**

Using these trees and the terminology of fork-join parallelism, we may consider a crude upper-limit on potential speed-up as the ratio of work (i.e. sum of edges) to span (i.e. the overall decomposition time). This gives 2.13 and 4.97 for Systems 2691 and 3295, respectively, and an "efficiency" (the ratio of actual to potential speed-up) as 87.8% and 74.4%, respectively. This suggests that our implementation indeed exploits the irregular parallelism available, and is able to exploit more parallelism when more is available.

Despite the inherent challenges of irregular parallelism in triangular decomposition, our implementation effectively utilizes that which is available through task parallelism and asynchronous generators. The use of generators in computer algebra is something which we hope to see applied elsewhere to improve upon irregular parallelism. For example, generators could also be applied to polynomial factorization, where factors could be produced and consumed along a pipeline consisting of square free factorization, distinct degree factorization, and equal degree factorization.

For the future of triangular decompositions, we hope to include methods which support more regular parallelism, for example, evaluation/interpolation schemes for the computation of subresultant chains [17] needed by RegularGCD. We also look to perform some of the solving over a prime field, where computations are more likely to split, and then lift the solutions [11].

Acknowledgments

The authors would like to extend thanks to IBM Canada Ltd (CAS project 880), NSERC of Canada (CRD grant CRDPJ500717-16, award PGSD3-535362-2019), and John P. May (Maplesoft).

REFERENCES

[1] M. Asadi, A. Brandt, C. Chen, S. Covanov, F. Mansouri, D. Mohajerani, R. H. C. Moir, M. Moreno Maza, Lin-Xiao Wang, Ning Xie, and Yuzhen Xie. 2018. Basic Polynomial Algebra Subprograms (BPAS). http://www.bpaslib.org.

[2] G. Attardi and C. Traverso. 1996. Strategy-Accurate Parallel Buchberger Algorithms. *J. Symbolic Computation* 22 (1996), 1–15.

[3] Philippe Aubry, Daniel Lazard, and Marc Moreno Maza. 1999. On the Theories of Triangular Sets. *J. Symb. Comput.* 28, 1-2 (1999), 105–124.

[4] Mordechai Ben-Ari. 1990. *Principles of concurrent and distributed programming.* Prentice Hall.

[5] Francesco Biscani. 2012. Parallel sparse polynomial multiplication on modern hardware architectures. In *ISSAC 2012, Grenoble, France, 2012.* 83–90.

[6] Janko Böhm, Wolfram Decker, Santiago Laplagne, Gerhard Pfister, Andreas Steenpaß, and Stefan Steidel. 2013. Parallel algorithms for normalization. *J. Symb. Comput.* 51 (2013), 99–114.

[7] F. Boulier, F. Lemaire, and M. Moreno Maza. 2006. Well known theorems on triangular systems and the D5 principle. In *Proc. of* Transgressive Computing 2006. Granada, Spain.

[8] B. Buchberger. 1987. The parallelization of critical-pair/completion procedures on the L-Machine. In *Proc. of the Jap. Symp. on functional programming.* 54–61.

[9] Reinhard Bündgen, Manfred Göbel, and Wolfgang Küchlin. 1994. A fine-grained parallel completion procedure. In *Proceedings of ISSAC.* ACM, 269–277.

[10] C. Chen and M. Moreno Maza. 2012. Algorithms for computing triangular decomposition of polynomial systems. *J. Symb. Comput.* 47, 6 (2012), 610–642.

[11] Xavier Dahan, Marc Moreno Maza, Éric Schost, Wenyuan Wu, and Yuzhen Xie. 2005. Lifting techniques for triangular decompositions. In *ISSAC 2005, Beijing, China, 2005, Proceedings.* 108–115.

[12] J. C. Faugere. 1994. Parallelization of Gröbner Basis. In *Parallel Symbolic Computation PASCO 1994 Proceedings*, Vol. 5. World Scientific, 124.

[13] M. Gastineau and J. Laskar. 2015. Parallel sparse multivariate polynomial division. In *Proceedings of PASCO 2015.* 25–33.

[14] Jiaxiong Hu and Michael B. Monagan. 2016. A Fast Parallel Sparse Polynomial GCD Algorithm. In *ISSAC 2016, Waterloo, ON, Canada, July 19-22, 2016.* 271–278.

[15] C. E. Leiserson. 2011. Cilk. In *Encyclopedia of Parallel Computing.* 273–288.

[16] F. Lemaire, M. Moreno Maza, and Y. Xie. 2005. The RegularChains library in MAPLE. *ACM SIGSAM Bulletin* 39, 3 (2005), 96–97.

[17] Xin Li, Marc Moreno Maza, and Wei Pan. 2009. Computations modulo regular chains. In *ISSAC 2009, Seoul, Republic of Korea, Proceedings.* 239–246.

[18] M. McCool, J. Reinders, and A. Robison. 2012. *Structured parallel programming: patterns for efficient computation.* Elsevier.

[19] M. Monagan and R. Pearce. 2010. Parallel sparse polynomial division using heaps. In *Proceedings of PASCO 2010.* ACM, 105–111.

[20] Michael B. Monagan and Baris Tuncer. 2018. Sparse Multivariate Hensel Lifting: A High-Performance Design and Implementation. In *ICMS 2018 - 6th International Conference, South Bend, IN, USA, July 24-27, 2018, Proceedings.* 359–368.

[21] Marc Moreno Maza and Yuzhen Xie. 2007. Component-level parallelization of triangular decompositions. In *PASCO 2007 Proceedings.* ACM, 69–77.

[22] B. D. Saunders, H. R. Lee, and S. K. Abdali. 1989. A parallel implementation of the cylindrical algebraic decomposition algorithm. In *ISSAC*, Vol. 89. 298–307.

[23] Michael L. Scott. 2009. *Programming Language Pragmatics (3. ed.).* Academic Press.

[24] Y. Xie. 2007. *Fast Algorithms, Modular Methods, Parallel Approaches, and Software Engineering for Solving Polynomial Systems Symbolically.* Ph.D. Dissertation.

The Orbiter Ecosystem for Combinatorial Data

Anton Betten
betten@math.colostate.edu
Department of Mathematics
Colorado State University
Fort Collins, CO

ABSTRACT

We describe a very versatile, fast and useful open source software package to compute combinatorial objects up to isomorphism called Orbiter. We provide an overview of some of the design decisions made during development, and we point out similar software packages. We discuss ways in which combinatorial data can be computed, analyzed and permanently stored for later use. This paper expands on earlier work published in [8].

CCS CONCEPTS

• **Mathematics of computing → Combinatorial algorithms**.

KEYWORDS

combinatorics, classification, combinatorial objects, codes, finite geometry

ACM Reference Format:
Anton Betten. 2020. The Orbiter Ecosystem for Combinatorial Data. In *International Symposium on Symbolic and Algebraic Computation (ISSAC '20), July 20–23, 2020, Kalamata, Greece.* ACM, New York, NY, USA, 8 pages. https://doi.org/10.1145/3373207.3403984

1 INTRODUCTION

For the purposes of this paper, combinatorial objects are elements of a finite set S on which a group G acts. We say that the set S is the permutation domain for the group G. We write the action on the right, so for $x \in S$ and $g \in G$ we let xg be the image of x under g. The orbit of x is the set of all images of x under G. It is denoted xG. We say that two objects $x, y \in S$ are isomorphic if they belong to the same orbit of G, and we write $x \sim y$ (or $x \sim_G y$ if need be). The set of orbits partition the set S. Examples of combinatorial objects that are of interest are combinatorial designs (which are set systems with special properties), linear codes (which are vector spaces over finite fields), graphs (also studied in Computer Science, Chemistry [45], Bioinformatics and Molecular Biology [31, 37] etc.), Hadamard matrices, Paley matrices, orthogonal arrays, and many other objects covered in basic combinatorics textbooks such as [35, 69]. A relatively recent addition to the field of combinatorics is the area of finite geometry. Here, geometric concepts are considered in geometries over finite fields. The existence theory of some of the objects is much more complicated over finite fields than for instance over the real numbers. The desire to learn more about these objects gives a strong motivation for using the computer to produce classifications for small instances of the problem.

Our interst in this paper is in the whole ecosystem of discovery based on experimental data. We describe our software Orbiter which can classify combinatorial objects of various kinds. The Orbiter ecosystem contributes towards the collection of mathematical data and the dissemination of it. For instance, tables of BLT-sets, optimal linear codes and tables of cubic surfaces with 27 lines over finite fields have been compiled with Orbiter. The data is computed using an algorithm called poset classification. Refinements of the algorithm for faster deep search techniques are possible. The lists of classified objects are written out in C++ source code, which is shared with users through GitHub. The next time Orbiter is compiled, the data becomes available for use. The cycle of discovery can be repeated. The process stops when the time or space complexity gets out of hand. As part of the process of working with mathematical data, new families of objects can be observed, and mathematical discovery can thus be promoted.

This paper is trying to pull together many ideas which serve the goal of classification and discovery. At times, these ideas may seem rather disconnected. However, given the constraints in time and space complexity, each idea contributes a little bit towards the greater goal. There is no silver bullet in the field of classification. Progress is inherently incremental. Every new classification result is a small victory. It does mean that we need to consider efficiency of algorithms carefully. For a problem with exponential complexity, constants in the complexity matter. For this reason, we will often discuss ways to improve algorithms slightly.

2 THE CLASSIFICATION PROBLEM

The classification problem for a class of combinatorial objects is the problem of producing a transversal (implicit or explicit) of the orbits of the group G on the set S [45]. As the classification depends on the group, we need to make clear which group and which action we consider. In many cases, we also care about the stabilizer subgroup of objects: For $x \in S$, we define $\mathrm{Stab}_G(x)$ the set of elements $g \in G$ for which $xg = x$. This group is also called the automorphism group of x. The recognition problem for combinatorial objects is this: Suppose that a class of objects S with action by the group G has been classified, and that $\mathcal{T}$ is a transversal of the G-orbits on S. Given an arbitrary element $x \in S$, determine the unique element $t \in \mathcal{T}$ such that $x \sim t$. The constructive version of the recognition problem asks

ISSAC '20, July 20–23, 2020, Kalamata, Greece

ACM ISBN 978-1-4503-7100-1/20/07…$15.00
https://doi.org/10.1145/3373207.3403984

for a group element $g \in G$ with $xg = t$. All these problems are part of the field of computational group theory, which is a rather young field of study that uses computational methods to explore (among others) apects of problems related to groups and group actions. For background reading, we refer the reader to [28, 36, 62]. For background on group actions, see [38, Section 1.12]. The orbit-stabilizer theorem is of great importance. We do not have space to recall it here.

Enumerative combinatorics [44, 65] is concerned with finding formulae to count classes of combinatorial objects (up to isomorphism or not). Enumerative results and constructive results suplement each other. An enumerative result may be used to confirm the validity of a computer classification. The problem of classifying combinatorial objects is difficult. The time and space complexity of an algorithm is often very hard to estimate. Problems arise from the fact that the sets may be large, or the orbits are too big to be generated fully. The question of whether two elements belong to the same orbit is known as the isomorphism problem. It is the subject of one of the Millenium Problems ("P vs NP") identified by the Clay institute. The problem is whether there is a polynomial time algorithm to solve the problem. For many interesting combinatorial objects, enumerative results are not known, so construction and classification is the only way to count these structures. For some combinatorial objects, we can count their total number, but not the number of isomorphism classes. Of course, the famous lemma of Burnside in group theory makes it clear that counting objects up to isomorphism is related to getting hold of explicit information about the stabilizer of objects. Combinatorial objects mostly come in families, parametrized by integers such as the order of a field or the size of a set.

Several algorithms have been proposed for classifying combinatorial objects. Many use the lexicographic order to produce a total order on the objects and sift through the set systematically. Faradhzev [32] and Read [58] are considered pioneers of this technique. The bottleneck is the need to test for isomorphisms which seems to have exponential complexity (but note the comment on Babai's work below). Modern algorithms often use canonical forms to decide the isomorphism problem, though canonical forms are similarly difficult to compute. The advantage of using canonical forms is that the number of canonical form computations is much less than the number of isomorphism test in a classification process. Many researchers have produced efficient algorithms for isomorphism testing of combinatorial objects. These algorithms often rely on a technique known as partition backtrack. The purpose of partition backtrack is to examine all different forms in which the object can appear, and to either decide isomorphims between two objects or compute automorphisms of an object (see [49, 50]). Nauty [52] is a software package for graph canonization developed originally by McKay, and later by McKay and Piperno (see also [53, 54]). Babai [4] claims that graph isomorphism is quasipolynomial. Though there had been some concerns about this claim, these concerns seem to have been resolved. However, so far this theoretical advance has not led to faster software.

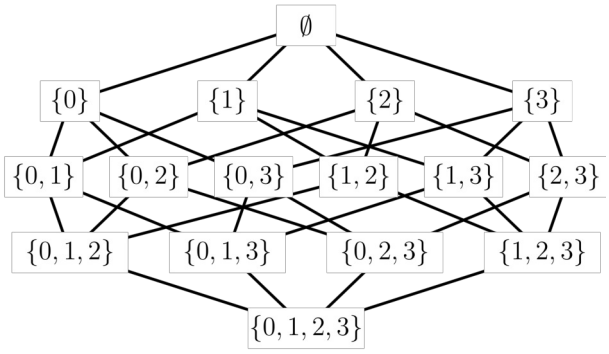


Figure 1: The poset of subsets of a 4-element set

One deficit with partition backtrack is that it needs to be tailored to the particular structure at hand. A good part of this is finding the right kind of isomorphism invariants which are able to distinguish well. An isomorph invariant is a function defined on the set of objects such that isomorphic objects have the same function value. If the invariants differ, the objects cannot be isomorphic. Computing invariants is often cheaper than doing an isomorphism test or runnning a canonical form algorithm. Often, the invariants are evaluated for smaller (related) objects. In the partition backtrack jargon, these invariants are the refiners (of the partition). Whenever two points have different invariants, they must lie in different classes of the partition. The hope is that the refiners help to discretize the partition quickly. A partition with many parts is good because then there are only few ways in which the object can be mapped, considering that classes must be preserved. There is – however – no guarantee that these invariants separate all the time. For instance, in the mentioned work of Babai, an old invariant of graphs due to Weisfeiler and Lehman [71] is utilized. It can be expressed in terms of counting colors between neighbors of vertices.

Some of the most powerful algorithms for classifying combinatorial objects make use of a poset structure. The poset of subsets of a four-element set is shown in Figure 1. Often, the combinatorial objects of interest are related to smaller objects, and the relation is invariant under the group. Smaller is not a very precise term, but practically it means that the objects are easier to classify. The smaller objects are classified first, and the larger objects are classified using the transversal of the smaller objects and the relation between small and large objects. This reduction process can be repeated, resulting in a poset structure of combinatorial objects and subobjects. For instance, if a combinatorial structure is made up of k points chosen from a set S of size n, we might consider the set of subsets of S of size at most k. It is often the case that the condition that makes the objects at level k induces a condition on the objects of size i for all $i \leq k$. For instance, if we are classifying graphs on n vertices which are regular of degree d, then we need to consider the poset of graphs on n vertices in which every vertex has degree *at most* d. In this poset, we try to find the objects at level $k = \frac{nd}{2}$ and list their orbits under the group $\mathrm{Sym}(n)$. For many types of combinatorial objects, it is possible to create posets of partial objects, so poset techniques can be applied.

McKay [53] uses the approach of posets and combines it with canonical forms. The result is an algorithm called canonical augmentation. It proceeds through the poset of combinatorial objects and subobjects in a depth first manner. Schmalz [61] has developed a technique of trading time versus memory and traverses the poset breadth first. We will refer to this technique as poset classification. The algorithm produces a data structure which describes a chosen transversal of the combinatorial object (and all subobjects), and allows for efficient constructive recognition.

Many authors have developed tools to classify combinatorial objects. Often, these tools are tailored to specific classes of objects. For instance, the book [43] is devoted exclusively to the classification problem of codes and designs. Some other software packages are GAP [33], Magma [26], and McKay's "Geng" (generate graphs) which comes with Nauty. The Orb [55] package in GAP has algorithms to compute long orbits. Theissen [68] used partition backtrack for computing normalizers of groups. In unpublished work, Steve Linton designed a least-image algorithm in GAP. The least image within an orbit is of course a canonical form. Using Leon's partition refinement technique, Jefferson et.al. [40, 41] recently improved on this. The original implementation of Leon's partition backtrack algorithm is available through Magma. Leon's program is called "PART". It was available on Leon's website at the University of Illinois Chicago. Unfortunately, Leon passed away and the website is no longer available. A reimplementation in C++ is available as GAP package ferret [39].

Orbiter [10] is a software package for the classification and recognition of general combinatorial objects, based on poset classification. It is a library of C++ classes and a command line application. Orbiter does not have a user interface. It can be controlled using Unix makefiles or shell scripts. Orbiter is open source software which is distributed on GitHub. A user's guide is available [11]. This paper expands on earlier work published in [8]. A software presentation of Orbiter will be held at ICMS2020 [22].

Orbiter was developed by the author at Colorado State University. It grew out of earlier work with DISCRETA [19], a system for computing combinatorial designs with assumed automorphism group. DISCRETA in turn was influenced by SYMMETRICA [46], a system developed by Axel Kohnert and devoted to the representation theory of symmetric groups. Another influence was a program called Double Coset Generator (DCC) by Bernd Schmalz. Most of this work was done in the 1990s at the Lehrstuhl of Professor Kerber and Professor Laue in Bayreuth, Germany. Sadly, Axel Kohnert passed away untimely in 2013.

3 WHY ORBITER?

As we have seen, there is now a great variety of different software packages for orbit computations. So, why do we need Orbiter? Perhaps it is helpful to look at some comparisons between Orbiter and GAP. Let us pick a small problem for which we know the answer. Suppose we want to compute the orbits of the collineation stabilizer $P\Gamma O^+(6,q)$ of the $Q(5,q)$ quadric on planes (three dimensional vector subspaces) in $PG(5,q)$. We know that there are 5 orbits. In Table 1, some computing times are collected. We use our own implementation of poset classification in GAP (witten with the help of Michel Lavrauw), which in turn relies on the GAP packages Fining, Forms, Cvec, Orb, Genss and Grape.

	Poset Classification		Basic Schreier
Problem	GAP	Orbiter	Orbiter
$Q(5,2)$	2.4 sec	0 sec	0 sec
$Q(5,3)$	2.76 sec	0 sec	0 sec
$Q(5,4)$	34 sec	0 sec	8 sec
$Q(5,5)$	13.5 sec	0 sec	1 min 12 sec
$Q(5,7)$	1 min 51 sec	1 sec	17 min 42 sec
$Q(5,9)$	> 3 hrs	8 sec	
$Q(5,11)$		19 sec	
$Q(5,13)$		42 sec	
$Q(5,17)$		3 min 17 sec	

Table 1: GAP vs Orbiter and Poset Classification vs Basic Schreier

For q prime, the collineation group equals the projectivity group. For instance, for $q = 7$, the specific Orbiter command that we ran for the poset classification approach was

```
orbiter.out -v 5 -linear_group -PGL 6 7 \
  -orthogonal 1 -end \
  -group_theoretic_activities \
  -orbits_on_subspaces 3
```

This command creates the orthogonal group of plus type (notice the "`-orthogonal 1`") as a subgroup of PGL(6, 7) and then computes the orbits on subspaces of vector space dimension 3. In this example, the 7 can be replaced by any other value of q. For q not prime, we used the collineation group, so the command was slightly different. Here is the example for $q = 9$:

```
orbiter.out -v 5 -linear_group -PGGL 6 9 \
  -orthogonal 1 -end \
  -group_theoretic_activities \
  -orbits_on_subspaces 3
```

Notice the double G in the group label "`-PGGL`". In Orbiter, the double letter represents the semilinear matrix group. More about Orbiter group will be said in Section 7. For the basic Schreier orbit algorithm, the Orbiter command was

```
orbiter.out -v 5 -linear_group -PGL 6 3 \
  -orthogonal 1 -on_k_subspaces 3 -end \
  -group_theoretic_activities -orbits_on_points
```

There are several points to be made. The first is efficiency. The C++ language allows more efficient data structures and algorithms than interpreted code written in a high level computer algebra system. In the literature on classification of combinatorial objects, benchmark problems are not always considered. In fact, many mathematical papers do not mention issues about implementation and computing time at all. It is therefore difficult to compare different algorithms and different systems on meaningful problems. Also, benchmarking natually measures the implementation and the speed of the machine as well. Nevertheless, Table 1 shows great speedup of poset

classification over the basic Schreier orbit algorithm. For $q = 7$, a speedup by a factor of 1000 can be measured! Despite using the best orbit packages that GAP has to offer, the timings for GAP are disappointing. The computation for $Q(5, 9)$ ran out of memory and did not finish (GAP seems to set itself a memory limit and does not seem to be able to automatically get more memory; it is not clear why a computer algebra system would not take advantage of the full system memory). The fact that $Q(5, 4)$ took longer than $Q(5, 5)$ seems to indicated an inefficiency in the collineation functions (which would point to a problem of Fining). As the collineation group is larger than the projectivity group, one would expect that $Q(5, 4)$ is faster than $Q(5, 5)$.

The second point is versatility. Orbiter is designed to solve the classification problem based on an arbitrary combination of a group, an action, and a partially ordered set. This cannot be said for many other software packages.

Other types of combinatorial objects for which Orbiter's poset classification has been used are cubic surfaces [20, 23], BLT-sets [2, 13], optimal linear codes [16], dual hyperovals [18], arcs [3, 17], unitals [1, 7], packings [14, 24], and spreads [25]. The work in [15] shows that Orbiter is fast in many but not all cases of combinatorial objects that were considered.

4 POSET CLASSIFICATION VERSUS PARTITION BACKTRACK

Computing canonical forms and automorphism groups of combinatorial objects of geometric nature is a lot slower than expected. For instance, computing the stabilizer of a (meaning *one*) $[18, 9, 8]$ code over $\mathbb{F}_4$ (a twisted tensor product code, see [12]) takes about 25 minutes using Magma (which uses Leon's partition backtrack software). However, in Orbiter, the classification of *all* $[18, 9, 6]$ codes over $\mathbb{F}_4$ is done in a matter of one or two minutes, and this includes computing the stabilizer of the code and the related smaller codes.

We also found that computing the canonical form of cubic surfaces using Nauty is slow (using an associated representation as a graph). For instance, surfaces in $\mathrm{PG}(3, 16)$ take quite some time in Nauty. On the other hand, the classification of *all* cubic surfaces in Orbiter is done in no more than 30 seconds. This includes computing the stabilizers of all surfaces (see [23]). What these examples show is that partition backtrack does not scale well. Also, the reduction of the isomorphism problem from projective space to a problem in graph theory may not be efficient. It seems that wore work is needed to say something meaningful about partition backtrack versus poset classification.

5 UNDER THE HOOD

The connection between combinatorial objects and double cosets in groups has been observed many times. For instance [59], describes an application in Chemisty. Following Kerber and Laue [45], we can say this: If the group G is transitive on the set S, and if H is a subgroup of G, then the H-orbits on S are in one-to-one correspondence to the double cosets

$$\mathrm{Stab}_G(x)\backslash G/H. \tag{1}$$

This point of view allows to reduce poset classification purely as a problem of determining double cosets in groups. This was the path taken by Schmalz [61] (see also [60]), who created an algorithm to classify double cosets in groups. Schmalz implemented his algorithm in a software package called DCC ("double coset generator"). One important aspect of the work by Schmalz was the use of subgroup ladders. Subgroup ladders are sequences of subgroups, each either contained in or containing the next group in the sequence, such that the subgroup index between consecutive terms is small. Good ladders starting with G and ending in H lead to efficient classification algorithms. The author used this algorithm as a basis for the system DISCRETA to classify designs with assumed automorphism groups (see [19]). The extension from orbits on sets to orbits on subspaces was done by Braun [27] based on earlier work by Weinrich [70].

The difference between poset classification for subsets and poset classification for subspaces lies in the ladder of subgroups that is used. For orbits on subsets of size t, with $|S| = n$, the subgroups are $H^{(2i+j)}$ where

$$H^{(2i+j)} = \begin{cases} \mathrm{Sym}_i \times \mathrm{Sym}_{n-i} & \text{if } i \le t,\ j = 0, \\ \mathrm{Sym}_i \times \mathrm{Sym}_1 \times \mathrm{Sym}_{n-i-1} & \text{if } i < t,\ j = 1, \end{cases}$$

For orbits on subspaces of dimension t, with $S = \mathbb{F}_q^n$, the subgroups are $H^{(2i+j)}$ where

$$H^{(2i+j)} = \begin{cases} (\mathrm{GL}(i, q) \times \mathrm{GL}(n - i, q))/Z & \text{if } i \le t,\ j = 0, \\ (\mathrm{GL}(i, q) \times \mathrm{GL}(1, q) \times \mathrm{GL}(n - i - 1, q))/Z & \text{if } i < t,\ j = 1, \end{cases}$$

Here, Z is the center, i.e. the group of invertible diagonal matrices. The subgroup indices are bounded from above by $n = |S|$ for subset lattices and by $\frac{q^n-1}{q-1}$ for subspace lattices. For many applications, this is sufficiently small and hence the algorithm is efficient in these cases. Orbiter implements both cases. A brief description of poset classification can be found in [15]. This paper is just a rephrasing of the original Schmalz algorithm into the language of group invariant relations and posets with group actions. Further comments will be made in Section 9 below.

The use of homomorphisms of group action is an idea that was considered by many authors, among them Kerber and Laue [45, Section 4]. Parker, in unpublished work from 1994, pioneered the use of helper groups for enumerating large orbits. This was picked up by Lübeck and Neunhöffer [51] and generalized to several helper groups in [56]. In recent applications of poset classification to finite geometry, action homomorphisms played an important role. For instance, in [23], the fact that $\mathrm{PGL}(4, q)$ is transitive on lines of $\mathrm{PG}(3, q)$ was important. In [21], the action of $\mathrm{PGL}(4, q)$ on arcs contained in hyperplanes of $\mathrm{PG}(3, q)$ is considered. In these two applications, the lowest level helper groups are the stabilizer of a line and the stabilizer of a plane, respectively.

One important contribution of Schmalz was the idea of trading time complexity versus space complexity. Instead of using a backtrack canonical form algorithm, the data structure built up during

the algorithm is used to compute canonical forms by means of constructive recognition. This is an important aspect of poset classification, and this is where Schmalz differs from McKay's canonical augmentation. We speculate that this is a major contributor to the success of Orbiter on problems in geometry. On the one hand, Orbiter makes use of the previously computed isomorphisms of subobjects. On the other hand, the known group is utilized right away. If reduction to graphs is used, the graph canonization algorithm has to rediscover the group each time, leading to inefficiencies.

6 MODELING COMBINATORIAL OBJECTS

To model combinatorial structures, we aim at representing the object as integers or sets of integers. This way, the permutation action can be set up. It also makes it easier to store objects. Lastly, it allows a uniform treatment of the combinatorial objects in the C++ style of object oriented programming.

We distinguish between basic objects ("atoms") and compound objects. Compound objects are made up of sets of basic objects. Since integers are the basic units for data structures on computers, it is natural to map basic objects to integers. These mappings are bijections from the set of combinatorial objects S to the integer interval $[0, \ldots, |S| - 1]$. In GAP, these bijections are called enumerators. In Magma, they are called indexing functions. In orbiter, the terminology rank and unrank function is used. The rank function assigns to each element $s \in S$ an integer $i \in [0, \ldots, |S| - 1]$. The unrank function computes the element s associated with an integer $i \in [0, \ldots, |S| - 1]$. The two functions are inverses of each other.

Once these basic objects can be enumerated, compound objects become subsets of the set of atoms. Vector spaces over finite fields can be represented by the set of ranks of a basis. This process of mapping combinatorial objects to sets of integers enables a uniform treatment of various different kinds of combinatorial objects. The group action can be represented as permutation groups acting on subsets. The combinatorial objects can be identified with maximal elements in certain subposets of the subset lattice on the atomic set.

7 THE GROUPS

As already noted, a classification problem consists of a triple of things: There is a set of objects, a group acting on them, and a partially ordered set. In order to get started, Orbiter needs to learn what the group is and how the group acts on the set. In the interest of versatility, these ingredients need to be pulled together with little or no extra programming. This leads us to look at a mechanism that allows to create (almost) arbitrary groups through command line arguments. Moreover, parts of the poset classification algorithm involve group theoretic tasks, for instance when computing the stabilizer of orbit representatives. This is yet another reason to look at groups and group theoretic algorithms.

There is a well-developed theory of algorithms for permutation groups. A central element of this theory is that of a stabilizer chain (Sims chain). Using these techniques, very large groups can be represented efficiently on a computer. Randomized algorithms to build up a stabilizer chain are utilized. Such algorithms have a small probability that the output is incorrect. However, if the group order is known in advance (which it often is), it can be used as a check to see whether the output is correct.

Command	Arguments	Group
-GL	$n\ q$	$\mathrm{GL}(n,q)$
-GGL	$n\ q$	$\Gamma\mathrm{L}(n,q)$
-SL	$n\ q$	$\mathrm{SL}(n,q)$
-SSL	$n\ q$	$\Sigma\mathrm{L}(n,q)$
-PGL	$n\ q$	$\mathrm{PGL}(n,q)$
-PGGL	$n\ q$	$\mathrm{P}\Gamma\mathrm{L}(n,q)$
-PSL	$n\ q$	$\mathrm{PSL}(n,q)$
-PSSL	$n\ q$	$\mathrm{P}\Sigma\mathrm{L}(n,q)$
-AGL	$n\ q$	$\mathrm{AGL}(n,q)$
-AGGL	$n\ q$	$\mathrm{A}\Gamma\mathrm{L}(n,q)$
-ASL	$n\ q$	$\mathrm{ASL}(n,q)$
-ASSL	$n\ q$	$\mathrm{A}\Sigma\mathrm{L}(n,q)$
-GL_d_q_wr_Sym_n	$d\ q\ n$	$\mathrm{GL}(d,q) \wr \mathrm{Sym}(n)$

Table 2: Basic Matrix Groups in Orbiter

A group in Orbiter can be created using command line arguments. Orbiter groups fall into two main categories, depending on what data structure is employed to store group elements. Elements of permutation groups are represented as functions using a table of values. Semilinear matrix group elements are stored using matrices, possibly extended by a field automorphism, and possibly extended by a translation vector. Every group comes with a default permutation domain. For some matrix groups, permutation domains made up of known short orbits are used (singular vectors for orthogonal group, the Hermitian variety for unitary groups). Table 2 shows the basic matrix groups available in Orbiter. Many other groups can be created as subgroups of these groups. For instance, the classical groups [66] arise as the stabilizer of a sesquilinear or quadratic form. For the orthogonal groups, the permutation representation on the points and/or lines of the underlying quadric can be considered using suitable enumerators. The symplectic group acts on the projective space. The unitary group comes with a permutation representation on the hermitian surface, using yet another enumerator. Many subgroups are available (for instance parabolics, Borel-subgroup, Singer cycle, Frobenius automorphism, etc.). Groups can also be created directly from a set of generators, for instance through command line arguments.

Orbiter distinguishes between the group and the action. The same group can appear in many different actions. One way to create new actions is by inducing the action. A base is an ordered sequence of points in the permutation domain which defines a chain of point stabilizer subgroups. The ith basic orbit is the orbit of the ith base point under the pointwise stabilizer of the first $i-1$ base points. This way, the group is decomposed into cosets of subgroups which are manageable. Many of these algorithms become inefficient for large basic orbits. For this reason, it is important to look for short orbits.

Not every action has short orbits. Regular actions (where the stabilizer of a point is trivial) are particularly difficult to handle. Some actions are imprimitive. These actions preserve a non-trivial partition of the permutation domain (cf. [30]). In this case, the induced action on the blocks of the partition gives rise to shorter orbits. Orbiter uses this technique for groups of wreath product or product action type. For those kinds of actions, the permutation domain is extended. The additional set is a permutation domain for the action on the blocks of the partition (in the case of product actions) or for the action of the factors as well as the action on the set of factors in the case of the wreath product action. These extended permutation domains make sure that short basic orbits can be found easily. On the other hand, the extended permutation action can be restricted to the natural action so that the process is transparent for the user of these groups.

8 BASIC ORBIT ALGORITHMS

One fundamental problem in computational group theory is that of computing orbits of finite groups acting on finite sets. The most basic orbit algorithm utilizes the technique of Schreier trees and Schreier vectors (see [28, 36, 62]) The idea is this: Fix a generating set $s_1, \ldots, s_k$ for the group G. Fix an element $a \in X$, whose orbit under G we wish to compute. The Schreier graph has as vertices the elements of the orbit. A directed edge labeled s_i goes from x to y whenever $xs_i = y$. The Schreier tree for the orbit of $a \in X$ is a spanning tree of the connected component containing a in the directed Schreier graph. The time and space complexity of a Schreier tree is linear in the size of the tree. This makes Schreier trees prohibitive for very large orbit problems. However, we need to know about basic orbit algorithms because poset classification reduces to them.

The Schreier vector allows constructive recognition: Given any element in the orbit, the Schreier vector allows us to find a group element which maps the orbit representative to the given element of the orbit. The group element is found by tracing the unique path in the tree that starts at the root node and ends at the given node. Notice that the set of elements created by tracing each vertex individually forms a system of coset representatives for the stabilizer. This follows from the well-known orbit-stabilizer theorem. Schreier vectors were introduced by Schreier in connection with the Nielsen-Schreier problem of finding generators of subgroups of free groups (cf. [42]). For a description, see [28]. The Nielsen-Schreier result is often used to find generators for the stabilizer of a point in a finite group action.

One of the main applications of Schreier trees and Schreier vectors is that they allow a technique to represent a permutation group (in a fixed action) on the computer, using a data structure called a Sims chain. Such a chain defined by a sequence of subgroups, each of which is the stabilizer of a point in the previous. The conditions are that the chain of subgroups starts with the group G and ends in the trivial group. The ith group in the chain $G^{(i)}$ has the form $\mathrm{Stab}_{G^{(i-1)}}(a_i)$ for some point a_i in the permutation domain (with the exception that $G^{(0)} = G$). We assume that $G^{(k)}$ is trivial for some k. The points $b_1, \ldots, b_k$ that are stabilized in the chain are called the base points. A generating set X for G is called a strong generating set if

$$G^{(i)} = \langle X \cap G^{(i)} \rangle$$

for $i = 0, \ldots, k$. The construction of a base and a strong generating set is an important algorithm in the field of computational group theory. Many implementations use a randomized version of the Schreier-Sims algorithm (see [48, 63, 64]).

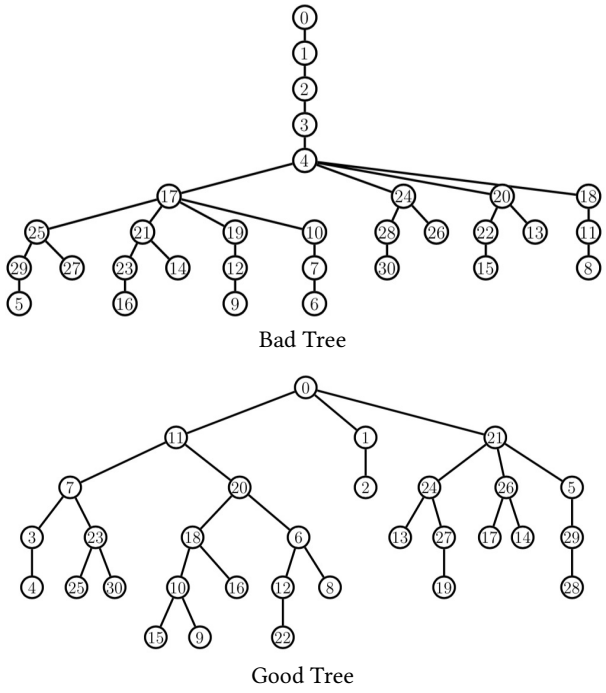


Figure 2: Two Schreier trees

Good Schreier trees are shallow, as this reduces the average time complexity to compute the group element needed for the constructive recognition problem of orbit elements. The shape of the tree depends on the choice of the generating set for the group. Techniques to choose generators so that the Schreier trees are shallow are described by Seress [62], based on earlier work of Babai [5]. In Figure 2, two Schreier trees for the action of $\mathrm{PGL}(5, 2)$ on $\mathrm{PG}(4, 2)$ are shown. The trees are computed using different generating sets. The expected word length is a term from information theory. It measures the average depth of the vertices in a tree (often using a probabililty distribution on the labels). A low expected word length means that the coset representatives are easier to compute, thus making the data structure more efficient. In this regard, the second tree is better because branching happens early which reduces the expected word length. In the example, the second tree has been created using the randomized version of the algorithm of Babai and Seress. GAP and Magma use schallow Schreier trees to make Sims chains for permutation groups more efficent.

9 POSET CLASSIFICATION

The use of poset orbits for studying combinatorial objects has a long history. In most applications, knowing the orbits helps to learn about relations between them, such as intersection numbers, covering numbers, decomposition matrices etc.

In design theory, structure constants from the subset lattice under the action of a group are used to construct designs with assumed symmetries [47]. Conway uses the orbits of M_{24} (the Mathieu group) in the action on subsets to study the Witt design in [29]. A theoretical framework for groups acting on posets was given by Plesken [57]. Intersection numbers of designs have been computed in [9]. Applications of structure constants play a role in representation theory as well. In his MathSciNet report of [51], J. Dixon writes:

> In the present paper the authors consider the computational problem of dealing with orbits of a group G (a finite permutation group or linear group) where the orbits are very large. They develop a technique where, by partitioning a G-orbit W into K-orbits $W_1, \ldots, W_m$ for a suitable subgroup K of G, only a small part of each G-orbit needs to be stored. At the same time they can compute the entries $a_{ij}(g) := |W_i g \cap W_j| (g \in G)$ of the "intersection matrix" efficiently. The latter turns out to be useful in applications of the condensation method.

In his last sentence, Dixon refers to the thesis of Thackray [67]. Roughly speaking, the structure constants tell us how different orbits relate to each other. From an algorithmic point of view, poset classification is often faster than the naive orbit algorithms. This is because unless the group is trivial, poset classification does not have to touch every element in the orbit. This makes it possible to construct orbits sub-linear in the size of the orbit, which is impossible with Schreier trees. In order to achieve this, the poset structure of the underlying set is exploited.

The orbits of a group G acting on a poset $\mathcal{P}$ allow us to define a new poset, called the poset of orbits. The goal of poset classification is to compute this poset. If $\mathfrak{L}$ is the lattice, and G is the group acting on $\mathfrak{L}$, we can define a poset as

$$\mathcal{P} = \{x \in \mathfrak{L} \mid P(x) = \text{true}\},$$

for some test-function $P : \mathfrak{L} \to \{\text{true}, \text{false}\}$. We require the following two conditions on P:

(1) P is invariant under the action of the group. That is, $P(xg) = P(x)$ for all $x \in \mathfrak{L}$ and all $g \in G$.
(2) If $x, y \in \mathfrak{L}$ with $x \le y$ then $P(y) = \text{true} \implies P(x) = \text{true}$.

The first property ensures that G acts on $\mathcal{P}$. The second property implies that the property is hereditary. The two lattices currently implemented in Orbiter are the subset lattice of a finite set and the subspace lattice of a finite dimensional vector space over a finite field. In both cases, Orbiter will compute the poset of orbits with respect to the given group.

Following [15], poset classification computes the orbits of the group G on the poset $\mathcal{P}$ one layer at a time, in a breadth first manner. The next level is classified using the information already computed at all lower levels. In order to ascend from one level to another, certain orbits called flag orbits are considered. A flag is a geometric term for an incident pair in a relation. A flag orbit is an orbit of the group on flags.

Type	Range
Primitive polynomials over finite fields	varies
BLT-sets	$q \le 73$, see [2, 13]
Cubic surfaces	$q \le 101$, $q = 128$, see [23]
Spreads	varies
Dual hyperovals	as in [18]

Table 3: Combinatorial Data in Orbiter

10 TABLES OF COMBINATORIAL DATA

Catalogues of combinatorial data are maintained by many authors, and for many different structures. Some tables record the optimal structures of a given kind, other tables are devoted to complete classifcations. There is a table for curves over finite fields with many points (cf. [34]). Ball maintains a table of optimal arcs (cf. [6]). The combinatorial data that is currently available in Orbiter is listed in Table 3. New results can be incorporated in the open source system much faster. In some cases, the data in Orbiter is more up-to-date than what is published in the literature.

ACKNOWLEDGMENTS

The author thanks the referees of this paper for valuable comments and many new leads, which found their way into the final version of the paper. He also thanks the following people who contributed directly or indirectly to the development of Orbiter: Abdullah AlAzemi for help with interfacing Nauty from Orbiter; Stefaan De Winter for discussions on the orthogonal geometries; Robert Lazar for very helpful comments on an earlier version of the manuscript; Sajeeb Roy Chowdhury for help with Schreier trees and clique finding algorithms; Michel Lavrauw for help with writing the GAP code for the poset classification algorithm.

REFERENCES

[1] Abdullah Al-Azemi, Anton Betten, and Dieter Betten. Unital designs with blocking sets. *Discrete Appl. Math.*, 163(part 2):102–112, 2014.
[2] Abdullah Al-Azemi, Anton Betten, and Sajeeb Roy Chowdhury. A rainbow-clique search algorithm for BLT-sets. In *ICMS 2018—Proceedings of the International Congress on Mathematical Software; James H. Davenport, Manuel Kauers, George Labahn, Josef Urban (ed.)*, pages 71–79. Springer, 2018.
[3] Awss Al-ogaidi and Anton Betten. Large Arcs in Small Planes, to appear in Congressus Numerantium.
[4] László Babai. Group, graphs, algorithms: the graph isomorphism problem. Proceedings of the International Congress of Mathematicians-Rio de Janeiro 2018. Vol. IV. Invited lectures, 3319-3336, World Sci. Publ., Hackensack, NJ, 2018. 68R05.
[5] László Babai, Gene Cooperman, Larry Finkelstein, and Ákos Seress. Nearly linear time algorithms for permutation groups with small base. In *Proc. of International Symposium on Symbolic and Algebraic Computation ISSAC '91*, pages 200–2009. ACM Press, New York, 1991.
[6] Simeon Ball. *Table of bounds on three dimensional linear codes or (n,r) arcs in PG(2,q)*, Available at https://mat-web.upc.edu/people/simeon.michael.ball/codebounds.html, accessed 11/16/2019.
[7] John Bamberg, Anton Betten, Cheryl E. Praeger, and Alfred Wassermann. Unitals in the Desarguesian projective plane of order 16. *J. Statist. Plann. Inference*, 144:110–122, 2014.
[8] A. Betten. Classifying discrete objects with Orbiter. ACM Communications in Computer Algebra 01/2014; 47(3/4):183-186.

[9] A. Betten. Intersection Numbers of Designs, Ph.D. thesis, Bayreuth University, 2001.

[10] Anton Betten. Orbiter – A program to classify discrete objects, 2019, https://github.com/abetten/orbiter.

[11] Anton Betten. Orbiter User's Guide. 2020. Distributed with Orbiter.

[12] Anton Betten. Twisted tensor product codes. *Des. Codes Cryptogr.*, 47(1-3):191–219, 2008.

[13] Anton Betten. Rainbow cliques and the classification of small BLT-sets. In *ISSAC 2013—Proceedings of the 38th International Symposium on Symbolic and Algebraic Computation*, pages 53–60. ACM, New York, 2013.

[14] Anton Betten. The packings of PG(3, 3). *Des. Codes Cryptogr.*, 79(3):583–595, 2016.

[15] Anton Betten. How fast can we compute orbits of groups? In *ICMS 2018—Proceedings of the International Congress on Mathematical Software; James H. Davenport, Manuel Kauers, George Labahn, Josef Urban (ed.)*, pages 62–70. Springer, 2018.

[16] Anton Betten, Michael Braun, Harald Fripertinger, Adalbert Kerber, Axel Kohnert, and Alfred Wassermann. *Error-correcting linear codes*, volume 18 of *Algorithms and Computation in Mathematics*. Springer-Verlag, Berlin, 2006. Classification by isometry and applications, With 1 CD-ROM (Windows and Linux).

[17] Anton Betten, Eun Ju Cheon, Seon Jeong Kim, and Tatsuya Maruta. The classification of $(42, 6)_8$ arcs. *Adv. Math. Commun.*, 5(2):209–223, 2011.

[18] Anton Betten, Ulrich Dempwolff, and Alfred Wassermann. On dual hyperovals of rank 4 over $\mathbb{F}_2$. *J. Geom.*, 108(1):75–98, 2017.

[19] Anton Betten, Evi Haberberger, Reinhard Laue, and Alfred Wassermann. *DISCRETA – A program system for the construction t-designs*. Lehrstuhl II für Mathematik, Universität Bayreuth, 1999. http://www.mathe2.uni-bayreuth.de/~discreta.

[20] Anton Betten, James W. P. Hirschfeld, and Fatma Karaoğlu. Classification of cubic surfaces with twenty-seven lines over the finite field of order thirteen. *Eur. J. Math.*, 4(1):37–50, 2018.

[21] Anton Betten and Fatma Karaoglu. The Number of Cubic Surfaces with 27 Lines Over a Finite Field. Accepted for publication in Journal of Algebraic Combinatorics.

[22] Anton Betten and Fatma Karaoğlu. Classifying Cubic Surfaces over Finite Fields with Orbiter. Software Demonstration at the International Congress on Mathematical Software – ICMS2020.

[23] Anton Betten and Fatma Karaoğlu. Cubic surfaces over small finite fields. *Des. Codes Cryptogr.*, 87(4):931–953, 2019.

[24] Anton Betten, Svetlana Topalova, and Stela Zhelezova. Parallelisms of PG(3, 4) invariant under a Baer involution. Sixteenth International Workshop on Algebraic and Combinatorial Coding Theory ACCT XVI. September 2-8, 2018. Svertlogorsk near Kaliningrad. acct2018.skoltech.ru.

[25] Anton Betten and Alfred Wassermann. Spreads of PG(3, 8) and PG(3, 9) containing a regulus. *Congr. Numer.*, 226:289–299, 2016.

[26] Wieb Bosma, John Cannon, and Catherine Playoust. The Magma algebra system. I. The user language. *J. Symbolic Comput.*, 24(3-4):235–265, 1997. Computational algebra and number theory (London, 1993).

[27] M. Braun. Konstruktion diskreter Strukturen unter Verwendung von Operationen linearer Gruppen auf dem linearen Verband. *Bayreuth. Math. Schr.*, (69):viii+153, 2004. Dissertation, Universität Bayreuth, Bayreuth, 2003.

[28] G. Butler. *Fundamental algorithms for permutation groups*, volume 559 of *Lecture Notes in Computer Science*. Springer-Verlag, Berlin, 1991.

[29] J. H. Conway and N. J. A. Sloane. *Sphere packings, lattices and groups*, volume 290 of *Grundlehren der Mathematischen Wissenschaften [Fundamental Principles of Mathematical Sciences]*. Springer-Verlag, New York, third edition, 1999. With additional contributions by E. Bannai, R. E. Borcherds, J. Leech, S. P. Norton, A. M. Odlyzko, R. A. Parker, L. Queen and B. B. Venkov.

[30] John D. Dixon and Brian Mortimer. *Permutation groups*, volume 163 of *Graduate Texts in Mathematics*. Springer-Verlag, New York, 1996.

[31] Mourad Elloumi and Albert Y. Zomaya. Algorithms in Computational Molecular Biology, Techniques, Approaches and Applications. Wiley, 2011.

[32] I. A. Faradžev. Constructive enumeration of combinatorial objects. In *Problèmes combinatoires et théorie des graphes (Colloq. Internat. CNRS, Univ. Orsay, Orsay, 1976)*, volume 260 of *Colloq. Internat. CNRS*, pages 131–135. CNRS, Paris, 1978.

[33] The GAP Group. *GAP – Groups, Algorithms, and Programming, Version 4.8.7*, 2017.

[34] Gerard van der Geer, Everett W. Howe, Kristin E. Lauter, and Christophe Ritzenthaler. Tables of curves with many points, 2009. Retrieved 11/14/2019.

[35] Marshall Hall, Jr. *Combinatorial theory*. Wiley Classics Library. John Wiley & Sons, Inc., New York, second edition, 1998. A Wiley-Interscience Publication.

[36] Derek F. Holt, Bettina Eick, and Eamonn A. O'Brien. *Handbook of computational group theory*. Discrete Mathematics and its Applications (Boca Raton). Chapman & Hall/CRC, Boca Raton, FL, 2005.

[37] Wolfgang Huber, Vincent J. Carey, Li Long, Seth Falcon, and Robert Gentleman. Graphs in molecular biology. BMC Bioinformatics 8, S8 (2007). https://doi.org/10.1186/1471-2105-8-S6-S8.

[38] Nathan Jacobson. *Basic algebra. I*. W. H. Freeman and Company, New York, second edition, 1985.

[39] Christopher Jefferson. *ferret – A gap package*, 2019.

[40] Christopher Jefferson, Eliza Jonauskyte, Markus Pfeiffer, and Rebecca Waldecker. Minimal and canonical images. *J. Algebra*, 521:481–506, 2019.

[41] Christopher Jefferson, Markus Pfeiffer, and Rebecca Waldecker. New refiners for permutation group search. *J. Symbolic Comput.*, 92:70–92, 2019.

[42] D. L. Johnson. *Topics in the theory of group presentations*, volume 42 of *London Mathematical Society Lecture Note Series*. Cambridge University Press, Cambridge-New York, 1980.

[43] P. Kaski and P. Östergård. *Classification algorithms for codes and designs*, volume 15 of *Algorithms and Computation in Mathematics*. Springer-Verlag, Berlin, 2006.

[44] Adalbert Kerber. *Applied finite group actions*, volume 19 of *Algorithms and Combinatorics*. Springer-Verlag, Berlin, second edition, 1999.

[45] Adalbert Kerber and Reinhard Laue. Group actions, double cosets, and homomorphisms: unifying concepts for the constructive theory of discrete structures. volume 52, pages 63–90. 1998. Algebra and combinatorics: interactions and applications (Königstein, 1994).

[46] Axel Kohnert. Symmetrica – representations of the symmetric group. http://www.algorithm.uni-bayreuth.de/en/research/SYMMETRICA/, circa 1990-2013.

[47] Earl S. Kramer and Dale M. Mesner. t-designs on hypergraphs. *Discrete Math.*, 15(3):263–296, 1976.

[48] Jeffrey S. Leon. On an algorithm for finding a base and a strong generating set for a group given by generating permutations. *Math. Comp.*, 35(151):941–974, 1980.

[49] Jeffrey S. Leon. Permutation group algorithms based on partitions. I. Theory and algorithms. *J. Symbolic Comput.*, 12(4-5):533–583, 1991. Computational group theory, Part 2.

[50] Jeffrey S. Leon. Partitions, refinements, and permutation group computation. In *Groups and computation, II (New Brunswick, NJ, 1995)*, volume 28 of *DIMACS Ser. Discrete Math. Theoret. Comput. Sci.*, pages 123–158. Amer. Math. Soc., Providence, RI, 1997.

[51] Frank Lübeck and Max Neunhöffer. Enumerating large orbits and direct condensation. *Experiment. Math.*, 10(2):197–205, 2001.

[52] Brendan McKay. Nauty and Traces (Version 2.7r1), Australian National University, 2020.

[53] Brendan D. McKay. Isomorph-free exhaustive generation. *J. Algorithms*, 26(2):306–324, 1998.

[54] Brendan D. McKay and Adolfo Piperno. Practical graph isomorphism, II. *J. Symbolic Comput.*, 60:94–112, 2014.

[55] J. Müller, M. Neunhöffer, and F. Noeske. *Orb – A gap package*, 2006.

[56] Jürgen Müller, Max Neunhöffer, and Robert A. Wilson. Enumerating big orbits and an application: B acting on the cosets of Fi_{23}. *J. Algebra*, 314(1):75–96, 2007.

[57] Wilhelm Plesken. Counting with groups and rings. *J. Reine Angew. Math.*, 334:40–68, 1982.

[58] Ronald C. Read. Every one a winner or how to avoid isomorphism search when cataloguing combinatorial configurations. *Ann. Discrete Math.*, 2:107–120, 1978. Algorithmic aspects of combinatorics (Conf., Vancouver Island, B.C., 1976).

[59] E. Ruch, W. Hässelbarth, and B. Richter. Doppelnebenklassen als Klassenbegriff und Nomenklaturprinzip für Isomere und ihre Abzählung, Theor. Chim. Acta(Berlin) 19(1970), 288-300.

[60] B. Schmalz. t-Designs zu vorgegebener Automorphismengruppe. *Bayreuth. Math. Schr.*, 41:1–164, 1992. Dissertation, Universität Bayreuth, Bayreuth, 1992.

[61] Bernd Schmalz. Verwendung von Untergruppenleitern zur Bestimmung von Doppelnebenklassen. *Bayreuth. Math. Schr.*, (31):109–143, 1990.

[62] Ákos Seress. *Permutation group algorithms*, volume 152 of *Cambridge Tracts in Mathematics*. Cambridge University Press, Cambridge, 2003.

[63] C. Sims. *Computation with permutation groups*, in Proc. Second Sympos. Symbolic and Algebraic Manipulation, Assoc. Comput. Mach., New York, 1971.

[64] Charles C. Sims. Group-theoretic algorithms, a survey. In *Proceedings of the International Congress of Mathematicians (Helsinki, 1978)*, pages 979–985. Acad. Sci. Fennica, Helsinki, 1980.

[65] Richard P. Stanley. *Enumerative combinatorics. Volume 1*, volume 49 of *Cambridge Studies in Advanced Mathematics*. Cambridge University Press, Cambridge, second edition, 2012.

[66] D.E. Taylor. *The geometry of the classical groups*, volume 9 of *Sigma Series in Pure Mathematics*. Heldermann Verlag, Berlin, 1992.

[67] J. G. Thackray. Modular representations of finite groups, Ph.D. thesis, Cambridge University, 1981.

[68] H. Theißen. Eine Methode zur Normalisatorberechnung in Permutationsgruppen mit Anwendungen in der Konstruktion primitiver Gruppen. Ph.D. thesis. 1995. Lehrstuhl D für Mathematik, RWTH Aachen.

[69] J. H. van Lint and R. M. Wilson. *A course in combinatorics*. Cambridge University Press, Cambridge, second edition, 2001.

[70] S. Weinrich. Konstruktionsalgorithmen für diskrete Strukturen und ihre Implementierung, Diploma thesis, University of Bayreuth, 1993.

[71] B. Weisfeiler and A. A. Lehman. A reduction of a graph to a canonical form and an algebra arising during this reduction. Nauchno-Technicheskaya Informatsia, Ser. 2, 9, 1968.

A Las Vegas Algorithm for Computing the Smith Form of a Nonsingular Integer Matrix

Stavros Birmpilis
Cheriton School of Computer Science
University of Waterloo
sbirmpil@uwaterloo.ca

George Labahn
Cheriton School of Computer Science
University of Waterloo
glabahn@uwaterloo.ca

Arne Storjohann
Cheriton School of Computer Science
University of Waterloo
astorjoh@uwaterloo.ca

ABSTRACT

We present a Las Vegas randomized algorithm to compute the Smith normal form of a nonsingular integer matrix. For an $A \in \mathbb{Z}^{n\times n}$, the algorithm requires $O(n^3(\log n + \log ||A||)^2(\log n)^2)$ bit operations using standard integer and matrix arithmetic, where $||A|| = \max_{ij} |A_{ij}|$ denotes the largest entry in absolute value. Fast integer and matrix multiplication can also be used, establishing that the Smith form can be computed in about the same number of bit operations as required to multiply two matrices of the same dimension and size of entries as the input matrix.

CCS CONCEPTS

• **Computing methodologies** → **Linear algebra algorithms**; *Exact arithmetic algorithms*; • **Mathematics of computing** → Probabilistic algorithms.

ACM Reference Format:
Stavros Birmpilis, George Labahn, and Arne Storjohann. 2020. A Las Vegas Algorithm for Computing the Smith Form of a Nonsingular Integer Matrix. In *International Symposium on Symbolic and Algebraic Computation (ISSAC '20), July 20–23, 2020, Kalamata, Greece.* ACM, New York, NY, USA, 8 pages. https://doi.org/10.1145/3373207.3404022

1 INTRODUCTION

Any nonsingular matrix $A \in \mathbb{Z}^{n\times n}$ is unimodularly equivalent to a unique diagonal matrix $S = \mathrm{diag}(s_1, s_2, \ldots, s_n)$ in Smith form. Here the diagonal entries, the invariant factors of A, are positive with $s_1 \mid s_2 \mid \cdots \mid s_n$, and unimodularly equivalent means there exist unimodular (with determinant ± 1) matrices $U, V \in \mathbb{Z}^{n\times n}$ such that $UAV = S$.

A natural goal for many computations on integer matrices is to design algorithms that have about the same cost as multiplying together two matrices of the same dimension and size of entries as the input matrix. If ω is a valid exponent for matrix multiplication — two $n \times n$ matrices can be multiplied in $O(n^\omega)$ operations from the domain of entries — then the target complexity is $(n^\omega \log ||A||)^{1+o(1)}$ bit operations, where $||A|| = \max_{ij} |A_{ij}|$ denotes the largest entry in absolute value, and the exponent $1 + o(1)$ indicates some missing $\log n$ and $\mathrm{loglog}\, ||A||$ factors. For randomized algorithms, in addition to stating the running time, we will indicate the type. A Monte Carlo type algorithm is allowed to return an incorrect result with probability at most $1/2$. A Las Vegas type algorithm is allowed to report failure with probability at most $1/2$, and if failure is not reported the output is certified to be correct.

The previously fastest algorithm for Smith form is due to Kaltofen and Villard [10]. They give a Las Vegas algorithm for computing the characteristic polynomial in time $(n^{3.2} \log ||A||)^{1+o(1)}$ assuming $\omega = 3$, and in time $(n^{2.695594} \log ||A||)^{1+o(1)}$ assuming the currently best known upper bound $\omega \le 2.3728639$ for ω [5] and the best known bound for rectangular matrix multiplication [6]. Using their characteristic polynomial algorithm together with ideas of Giesbrecht [8], they obtain a Monte Carlo algorithm for Smith form with the same running time. In this paper we give a Las Vegas algorithm for Smith form in time $(n^\omega \log ||A||)^{1+o(1)}$.

A Las Vegas algorithm with the target complexity was previously known for the determinant [13]. Like that algorithm, we utilize a "dimension × precision ≤ invariant" compromise. By Hadamard's bound, $|\det A| = s_1 s_2 \cdots s_n \le \Delta^n$ where $\Delta = n^{1/2}||A||$. Thus, the number of invariant factors with bitlength between $(1/2^i) \times n \log \Delta$ and $(1/2^{i-1}) \times n \log \Delta$ is bounded by 2^i. The determinant algorithm [13] embeds A into a matrix

$$C := \left[\begin{array}{c|ccc} A & & & \\ \hline B_t & I & & \\ \vdots & & \ddots & \\ B_0 & & & I \end{array}\right] \in \mathbb{Z}^{O(n)\times O(n)},$$

where $t = O(\log n)$ and the blocks $B_i \in \mathbb{Z}^{O(2^i)\times n}$ are chosen randomly, and then returns $|\det A| = (\det H_0)(\det H_1)\cdots(\det H_{t+1})$, where

$$H = \left[\begin{array}{c|ccc} H_{t+1} & * & \cdots & * \\ \hline & H_t & \cdots & * \\ & & \ddots & \vdots \\ & & & H_0 \end{array}\right].$$

is the (row) Hermite form of A. With high probability, the part of the determinant captured by H_{i+1}, which has about twice the dimension as H_i, can be computed at a precision about half that required to compute $\det H_i$. We remark that the diagonal blocks H_* are not computed explicitly, and the offdiagonal blocks $*$ of H are avoided entirely.

To obtain the Smith form and not just the determinant, and to facilitate certification of the output, we take here a more structured approach and compute a *Smith massager* for A. This is a tuple of $n \times n$ integer matrices (U, M, T, S) such that

$$B := \left[\begin{array}{c|c} A & \\ \hline & I_n \end{array}\right] \left[\begin{array}{c|c} I_n & \\ \hline U & I_n \end{array}\right] \left[\begin{array}{c|c} I_n & M \\ \hline & T \end{array}\right] \left[\begin{array}{c|c} I_n & \\ \hline & S^{-1} \end{array}\right] \in \mathbb{Z}^{2n\times 2n} \quad (1)$$

ISSAC '20, July 20–23, 2020, Kalamata, Greece

ACM ISBN 978-1-4503-7100-1/20/07…$15.00
https://doi.org/10.1145/3373207.3404022

is integral, with T unit upper triangular and S nonsingular and in Smith form. The algorithm succeeds if we compute a *maximal* Smith massager, meaning that S is the Smith form of A. Since (1) implies $(\det B)(\det S) = \det A$, we can conclude from the uniqueness of the Smith form that the massager is maximal if and only if B is unimodular.

Our algorithm for computing the Smith form has three phases. Phase 1 uses a Monte Carlo approach [4, Theorem 2.1] to compute the largest invariant factor s_n of A. Phase 2 iteratively computes a Smith massager of A, together with the massaged matrix B in (1), which will be maximal with probability at least $1/2$. Phase 3 uses a known algorithm [11] to assay if B is unimodular.

Phase 2 is the main part of our algorithm. It uses $O(\log n)$ iterations to build a Smith massager that extracts more and more invariant factors from A. The algorithm begins by initializing (U, M, T, S) to be the trivial Smith massager, with $U, M \in 0^{n\times n}$ and $T = S = I_n$. At the start of iteration $i = 0, 1, 2, \ldots$ we assume that the current Smith massager is such that B in (1) has the same Smith form as A but with the largest $2^i - 1 = 2^0 + 2^1 + \cdots + 2^{i-1}$ invariant factors replaced by 1. The goal at iteration i is then to compute and extract the next largest 2^i invariant factors. For example, at iterations $i = 0, 1$ and 2, the largest $1, 2$ and 4 invariant factors of the current B are equal to (s_n), (s_{n-1}, s_{n-2}) and $(s_{n-3}, s_{n-4}, s_{n-5}, s_{n-6})$, respectively. Section 3 shows how to recover the largest 2^i invariant factors of B with high probability by computing a projection $B^{-1}J$ for a randomly chosen J with column dimension $O(2^i)$. We exploit the fact that if s is a multiple of the largest invariant factor of B, then the smallest 2^i invariant factors of sB^{-1} correspond to the largest 2^i invariant factors of B. In Sections 4 and 5 we show how to compute a Smith massager that will extract the largest 2^i invariant factors from B, while in Section 6 we show how to combine the partial Smith massagers obtained at each iteration.

Cost model

Following [7, Section 8.3], cost estimates are given using a function $\mathsf{M}(d)$ that bounds the number of bit operations required to multiply two integers bounded in magnitude by 2^d. We use $\mathsf{B}(d)$ to bound the cost of integer gcd-related computations such as the extended euclidean algorithm. We can always take $\mathsf{B}(d) = O(\mathsf{M}(d)\log d)$. If $\mathsf{M}(d) \in \Omega(d^{1+\epsilon})$ for some $\epsilon > 0$ then $\mathsf{B}(d) \in O(\mathsf{M}(d))$.

As usual, we assume that M is superlinear and subquadratic. We also assume that $\mathsf{M}(ab) \in O(\mathsf{M}(a)\,\mathsf{M}(b))$ for $a, b \geq 1$. We assume that $\omega > 2$, and to simplify cost estimates we make the assumption that $\mathsf{M}(d) \in O(d^{\omega-1})$. This assumption simply stipulates that if fast matrix multiplication techniques are used, then fast integer multiplication techniques should also be used. The assumptions stated in this paragraph apply also to B.

2 COMPUTATIONAL TOOLS

A key step during the iterations of phase 2 of our Smith form algorithm is to compute a projection $B^{-1}J$ for a partially massaged matrix B and a randomly chosen J. More specifically, we will have an $s \in \mathbb{Z}_{>0}$ such that sB^{-1} is integral, and what we need is $\mathrm{Rem}(sB^{-1}J, s)$, that is, the matrix $sB^{-1}J$ with entries reduced modulo s. In this section we show how to compute $\mathrm{Rem}(sB^{-1}J, s)$ within the target complexity by reducing to a deterministic variant of high-order lifting [11, Section 3] for linear system solving.

There are two issues that arise that prevent a direct application of fast linear system solving. First, the massaged matrix B may have some entries with large bitlength, adversely affecting the cost. In Section 2.1 we recall a partial linearization technique that can be used to obtain a matrix with smoothed entries that can be used in lieu of B. Second, entries in $sB^{-1}J$ may have bitlength much larger than the bitlength of s. In Section 2.2 we develop a deterministic variant of integrality certification that allows $\mathrm{Rem}(sB^{-1}J, s)$ to be computed more directly, without computing $sB^{-1}J$ first.

2.1 Partial linearization

The number of bits in the binary representation of a positive integer a is $\lfloor \log_2 a \rfloor + 1$. Any integer a thus satisfies $|a| \leq 2^{\mathrm{length}(a)} - 1$ where

$$\mathrm{length}(a) := \begin{cases} 1 & \text{if } a = 0 \\ \lfloor \log_2 |a| \rfloor + 1 & \text{otherwise} \end{cases}.$$

For v an integer vector or matrix, define $\mathrm{length}(v) := \mathrm{length}(||v||)$.

The cost of high-order lifting [11, Section 3] is sensitive to $\mathrm{length}(A)$. This is an issue because some of the intermediate matrices that we will need to give as input to the high-order lifting algorithm will likely have some rows of large length, even though the average row length is well bounded. In some cases, for an $n \times n$ input matrix A, $\mathrm{length}(A)$ could be about n times as large as the average row length. For the purposes of giving a concrete example, and not considering such an extreme case, consider the input matrix

$$A = \begin{bmatrix} 7 & 4 & 9 & 10 \\ 1 & 1 & 3 & 7 \\ 58538 & 43609 & 77404 & 7995 \\ 72526300 & 20544909 & 66620465 & 80378234 \end{bmatrix}.$$

The lengths of the rows of A are $[4, 3, 17, 27]$. Thus $\mathrm{length}(A) = 27$. But the average length is bounded by $d := \lceil (4 + 3 + 17 + 27)/4 \rceil = 13$. With some adjustment, the partial linearization technique [9, Section 6] developed for polynomial matrices can be applied in the integer setting. Assuming integers are represented in binary, the technique allows to produce from A without computation a new matrix

$$\bar{A} = \begin{bmatrix} 7 & 4 & 9 & 10 & 0 & 0 & 0 \\ 1 & 1 & 3 & 7 & 0 & 0 & 0 \\ 1194 & 2649 & 3676 & 7995 & -8192 & 0 & 0 \\ 2524 & 7565 & 3121 & 6522 & 0 & -8192 & 0 \\ 7 & 5 & 9 & 0 & 1 & 0 & 0 \\ 661 & 2507 & 8132 & 1619 & 0 & 1 & -8192 \\ 1 & 0 & 0 & 1 & 0 & 0 & 1 \end{bmatrix}$$

that satisfies $||\bar{A}|| \leq 2^d$ and can be used in lieu of A. The next theorem summarizes the special case of partial linearization that we require.

Theorem 1. *Let $A \in \mathbb{Z}^{n\times n}$ have average row length bounded by $d \in \mathbb{Z}_{\geq 0}$. If the (2^d)-adic expansions of entries of A are available, we can construct without computation from A a new matrix $\bar{A} \in \mathbb{Z}^{\bar{n}\times\bar{n}}$ such that $\bar{n} < 2n$, $||\bar{A}|| \leq 2^d$, $\det \bar{A} = \det A$, and, if A is nonsingular, with the principal $n \times n$ submatrix of $\bar{A}^{-1}$ equal to A^{-1}.*

2.2 Integrality certification

Any rational number can be written as an integer and a proper fraction. For example,

$$\frac{9622976468279041913}{21341} = 450914974381661 + \frac{14512}{21341},$$

where 450914974381661 is the quotient and 14512 is the remainder of the numerator with respect to the denominator. A similar construction replaces the quotient with a truncated p-adic expansion of the fraction, where p should be relatively prime to the denominator. For example,

$$\frac{9622976468279041913}{21341} = 9035820194880943821 - \frac{10453}{21341} \times 2^{64}. \quad (2)$$

In our case, we only require the remainder and not the quotient. Multiplying (2) by 21341 shows that

$$14512 = -10453 \times 2^{64} \bmod 21341.$$

The same idea works for integer matrices. Suppose we are given a nonsingular integer matrix $A \in \mathbb{Z}^{n\times n}$, a $B \in \mathbb{Z}^{n\times m}$ and an $s \in \mathbb{Z}_{>0}$. Then integrality certification [13, Section 11] can test if $sA^{-1}B$ is integral, and, if so, return the matrix $\mathrm{Rem}(sA^{-1}B, s)$. High-order lifting is used to achieve a cost that is sensitive to length(s)+length(B), rather than length$(sA^{-1}B)$. The algorithm in [13] is randomized. Here we show how to solve the integrality certification problem deterministically using some recently developed techniques, provided that $\det A \perp 2$.

Our approach is to first use double-plus-one lifting [11, Section 3] to compute a high-order residue $R \in \mathbb{Z}^{n\times n}$ such

$$A^{-1} = D + A^{-1}R \times 2^h \quad (3)$$

for some h such that

$$2^h > 2sn^{n/2}||A||^{n-1}||B||. \quad (4)$$

The matrix D, which will satisfy $||D|| \le (0.6)2^h$ [11, Theorem 5], is not needed and not computed explicitly. If the dimension $\times$ precision compromise

$$m \times (\log s + \log ||B||) \in O(n(\log n + \log ||A||)) \quad (5)$$

holds, then by [11, Theorem 8] such an R can be computed in time

$$O(n^\omega \, \mathsf{M}(\log n + \log ||A||) \log n). \quad (6)$$

Now multiply equation (3) on the right by sB to see that

$$sA^{-1}B = sDB + A^{-1}(sRB) \times 2^h. \quad (7)$$

The next step is to use deterministic linear solving [1, Section 3] to compute $\mathrm{Rem}(A^{-1}(sRB), 2^\ell)$ for some ℓ such that

$$2^\ell > 2n||A||(0.6sn||B||). \quad (8)$$

Assuming (5), this can also be done in time (6) [1, Corollary 7].

Adjusting slightly the argument of the proof of [13, Theorem 46] to account for the fact that D in (3) satisfies $||D|| \le (0.6)2^h$ instead of $||D|| \le 2^h$, it can be shown, for the choices of h and ℓ in (4) and (8), respectively, that if C is set to be the matrix equal to $\mathrm{Rem}(A^{-1}(sRB), 2^\ell)$ but with entries reduced in the symmetric range modulo 2^ℓ, then $C = sA^{-1}RB$ (and hence $sA^{-1}RB$ is integral) if and only if $||C|| < 0.6sn||B||$. Considering (7), it then follows that $\mathrm{Rem}(C \times 2^h, s)$ is equal to $\mathrm{Rem}(sA^{-1}B, s)$.

We will need to apply integrality certification with an input matrix A that has skewed row lengths. To maintain a good complexity, we can work with the partial linearization $\bar{A} \in \mathbb{Z}^{\bar{n}\times\bar{n}}$ of Theorem 1. Compute a high-order residue $\bar{R}$ for $\bar{A}$. Let $\bar{B} \in \mathbb{Z}^{\bar{n}\times m}$ be equal to B but augmented with $\bar{n} - n$ zero rows. Then the first n rows of $\mathrm{Rem}(\bar{A}^{-1}(s\bar{R}\bar{B}), 2^\ell)$ comprise an integrality certificate for $sA^{-1}B$. The running time is as in (6) but with $\log ||A||$ replaced by the average of the lengths of the rows. This gives the following.

Theorem 2. *Let $A \in \mathbb{Z}^{n\times n}$ satisfying $\det A \perp 2$, $s \in \mathbb{Z}_{>0}$, and $B \in \mathbb{Z}/(s)^{n\times m}$ be given. There exists an algorithm that will test if $sA^{-1}B$ is integral, and, if so, return the matrix $\mathrm{Rem}(sA^{-1}B, s)$. If $m \times \log s \in O(n(d + \log n))$, where d is the average of the lengths of the rows of A, then the cost is $O(n^\omega \, \mathsf{M}(d + \log n) \log n)$.*

3 LARGEST INVARIANT FACTORS

In this section we show how the largest r invariant factors of a nonsingular matrix $A \in \mathbb{Z}^{n\times n}$ can be recovered with high probability by randomly sampling $r + O(\log r)$ vectors from the columns space of A^{-1}. The method assumes we know an $s \in \mathbb{Z}_{>0}$ that is a multiple of the largest invariant factor s_n of A.

Let $U, V \in \mathbb{Z}^{n\times n}$ be unimodular with $S = UAV = \mathrm{diag}(s_1, \ldots, s_n)$ the Smith form of A. Then the *reverse Smith form* of $sA^{-1} \in \mathbb{Z}^{n\times n}$ is equal to $sS^{-1} = \mathrm{diag}(s/s_1, \ldots, s/s_n)$. By reverse Smith form we simply mean that the order of both the rows and the columns is reversed. The smallest invariant factor is thus located in the last row and column. Since the largest invariant factor s/s_1 of the Smith form of sA^{-1} is a divisor of s, the Smith form of sA^{-1} can be computed modulo s over $\mathbb{Z}/(s)$. For convenience, should a diagonal entry in the Smith form over $\mathbb{Z}/(s)$ vanish modulo s, we replace it with s. For example, the reverse Smith form of $\mathrm{diag}(1, 2, 8, 16, 16)$ over $\mathbb{Z}/(16)$ is equal to $\mathrm{diag}(16, 16, 8, 2, 1)$.

To recover only the largest r invariant factors of A, the idea is to choose $J \in \mathbb{Z}/(s)^{n\times r}$ uniformly at random and hope that the submatrix comprised of the last r rows of the reverse Smith form of $sA^{-1}J \in \mathbb{Z}/(s)^{n\times r}$ is equal to $S_1 = \mathrm{diag}(s/s_{n-r+1}, \ldots, s/s_n)$. To ensure a high probability of success, we adjust the recipe slightly by augmenting J with a small number of additional columns k. The main result of this section is:

Theorem 3. *Let $A \in \mathbb{Z}^{n\times n}$ be nonsingular with Smith form $S = \mathrm{diag}(s_1, \ldots, s_n)$. Let $s \in \mathbb{Z}_{>0}$ be a multiple of s_n. If $J \in \mathbb{Z}/(s)^{n\times(r+k)}$ is chosen uniformly at random for $r \ge 1$ and $k \ge 2$, then, with probability at least $1 - \frac{1}{2^{k-1}}$, the trailing $r \times r$ submatrix of the reverse Smith form of $sA^{-1}J$ over $\mathbb{Z}/(s)$ is equal to $S_1 = \mathrm{diag}(s/s_{n-r+1}, \ldots, s/s_n)$.*

Before we prove Theorem 3, we establish a property of $J \in \mathbb{Z}/(s)^{n\times(r+k)}$ that is sufficient to ensure success. In the following lemma, recall that $U, V \in \mathbb{Z}^{n\times n}$ are unimodular matrices such that $S = UAV$, and thus $sA^{-1} = VsS^{-1}U$.

Lemma 4. *If the $r \times (r + k)$ submatrix comprised of the last r rows of $UJ \in \mathbb{Z}^{n\times(r+k)}$ is right equivalent to $\left[\begin{array}{c|c} 0_{r\times k} & I_r \end{array}\right]$ over $\mathbb{Z}/(s)$, then the trailing $r \times r$ submatrix of the reverse Smith form of $sA^{-1}J$ over $\mathbb{Z}/(s)$ is equal to $S_1 = \mathrm{diag}(s/s_{n-r+1}, \ldots, s/s_n)$.*

Proof. Decompose $sS^{-1} = \mathrm{diag}(S_2, S_1)$ where S_1 is as in the statement of the theorem, and $S_2 = \mathrm{diag}(s/s_1, \ldots, s/s_{n-r})$.

We work entirely over $\mathbb{Z}/(s)$. By assumption, we have that

$$UJ \equiv_R \left[\begin{array}{c|c} U_1 & U_2 \\ \hline & I_r \end{array}\right] \tag{9}$$

for $U_1 \in \mathbb{Z}/(s)^{(n-r)\times k}$ and $U_2 \in \mathbb{Z}/(s)^{(n-r)\times r}$. Since all entries in S_2 are divisible by the largest invariant factor s/s_{n-r+1} of S_1, it will be sufficient to show that

$$sA^{-1}J \equiv \left[\begin{array}{c|c} S_2U_1 & \\ \hline & S_1 \end{array}\right].$$

We have

$$\begin{aligned} sA^{-1}J &= V(sS^{-1})UJ && (10)\\ &\equiv_L (sS^{-1})UJ && (11)\\ &\equiv_R (sS^{-1})\left[\begin{array}{c|c} U_1 & U_2 \\ \hline & I_r \end{array}\right] && (12)\\ &= \left[\begin{array}{c|c} S_2U_1 & S_2U_2 \\ \hline & S_1 \end{array}\right] \equiv_L \left[\begin{array}{c|c} S_2U_1 & \\ \hline & S_1 \end{array}\right] && (13) \end{aligned}$$

Here, (10) follows from $S = UAV$, (11) because V is unimodular, and (12) from (9). To obtain (13), we can use a unimodular left transformation to zero out the block S_2U_2 since its entries are all multiples of the diagonal entries in S_1. □

We need two additional technical lemmas before proving the main theorem.

Lemma 5. *If $k \geq 1$, $t \geq 0$ and $0 < x \leq 1/2$, then $\prod_{i=k}^{k+t}(1-x^i) \geq 1 - 2x^k + x^{k+t}$.*

Proof. We will use induction on t. For $t = 0$ the inequality is trivially true. We assume that $\prod_{i=k}^{k+t}(1-x^i) \geq 1 - 2x^k + x^{k+t}$ for fixed t, and we need to show the same for $t \leftarrow t+1$.

$$\begin{aligned} \prod_{i=k}^{k+t+1}\left(1-x^i\right) &= (1-x^{k+t+1})\prod_{i=k}^{k+t}\left(1-x^i\right)\\ &\geq (1-x^{k+t+1})(1-2x^k+x^{k+t})\\ &= 1-2x^k+x^{k+t}-x^{k+t+1}+2x^{2k+t+1}-x^{2(k+t)+1}\\ &= 1-2x^k+x^{k+t+1}\left(1+\frac{1}{x}-2+2x^k-x^{k+t}\right)\\ &\geq 1-2x^k+x^{k+t+1} \end{aligned}$$

In the last step we used that $x \leq 1/2$. □

Lemma 6. *If $k \geq 2$, then $\zeta(k+1) - 1 < 2^{-k}$, where ζ denotes the Riemann zeta function.*

Proof. The lemma inequality is equivalent to:

$$\zeta(k+1) - 1 < 2^{-k} \Leftrightarrow \sum_{n=2}^{\infty}\frac{1}{n^{k+1}} < 2^{-k} \Leftrightarrow \sum_{n=2}^{\infty}\left(\frac{2}{n}\right)^{k+1} < 2.$$

Since the left-hand side of the last inequality is a decreasing function on k, it suffices to show the claim for $k = 2$, i.e., $\zeta(3) - 1 < \frac{1}{4}$. □

Proof (of Theorem 3). We start by defining the following event.

FR_p: For a prime p that divides s, the last r rows of the random matrix $J \in (\mathbb{Z}/(s))^{n\times(r+k)}$ have full row rank over $\mathbb{Z}/(p)$.

If the last i rows of J over $\mathbb{Z}/(p)$ are linearly independent, then they span a vector space containing p^i rows. The probability that an additional row avoids that space is $(1 - p^i/p^{r+k})$, and thus

$$\Pr[FR_p] = \prod_{j=k+1}^{r+k}\left(1-\frac{1}{p^j}\right).$$

The above result has already been shown and extensively used in the literature [2, 3]. Furthermore, by applying Lemma 5, we obtain

$$\Pr[\neg FR_p] \leq 2\frac{1}{p^{k+1}}. \tag{14}$$

Next, we define the event described by Lemma 4.

FR_U: For a matrix $U \in \mathbb{Z}^{n\times n}$, the last r rows of the random matrix $UJ \in (\mathbb{Z}/(s))^{n\times(r+k)}$ are right equivalent to $\left[\begin{array}{c|c} 0_{r\times k} & I_r \end{array}\right]$ over $\mathbb{Z}/(s)$.

A matrix J is right equivalent to $\left[\begin{array}{c|c} 0_{r\times k} & I_r \end{array}\right]$ over $\mathbb{Z}/(s)$ if and only if it has full row rank over $\mathbb{Z}/(p)$ for all primes p that divide s. Therefore,

$$\begin{aligned} \Pr[\neg FR_{I_n}] &\leq \sum_{\substack{p\mid s\\ p \text{ prime}}}\Pr[\neg FR_p] && (15)\\ &\leq 2\sum_{p=2}^{\infty}\frac{1}{p^{k+1}} && (16)\\ &= 2(\zeta(k+1)-1) < 2^{1-k}. && (17) \end{aligned}$$

We applied the union bound in (15), equation (14) in (16), and Lemma 6 in (17).

Finally, multiplying matrices from $\mathbb{Z}/(s)^{n\times(r+k)}$ with a unimodular matrix $U \in \mathbb{Z}^{n\times n}$ is an isomorphism back to $(\mathbb{Z}/(s))^{n\times(r+k)}$, which implies that $\Pr[FR_{I_n}] = \Pr[FR_U]$. So, according to Lemma 4, the probability described in Theorem 3 must be at least $\Pr[FR_U] = \Pr[FR_{I_n}] > 1 - \frac{1}{2^{k-1}}$. □

4 PROJECTION BASIS

Throughout this section let $A \in \mathbb{Z}^{n\times n}$ be nonsingular. In Section 3 we showed that the projection $A^{-1}J$, for a well chosen integer matrix J, can reveal the r largest invariant factors of A. In this section we show how these invariant factors can be extracted from A to produce a matrix B that has the same Smith form as A but with the r largest invariant factors replaced by trivial ones.

For any $J \in \mathbb{Z}^{n\times *}$, the set

$$\text{Proj}(A, J) := \{v \in \mathbb{Z}^{1\times n} \mid vA^{-1}J \in \mathbb{Z}^{1\times r}\}$$

forms an integer lattice. A basis of $\text{Proj}(A, J)$ is a matrix $H \in \mathbb{Z}^{n\times n}$ such that the set of all integer linear combinations of rows of H is equal to $\text{Proj}(A, J)$. Bases of $\text{Proj}(A, J)$ are always nonsingular and are unique up to left equivalence. For example, a basis of $\text{Proj}(A, 0_{n\times *})$ is I_n, while a basis of $\text{Proj}(A, I_n)$ is given by A itself.

Lemma 7. *If H is a basis of $\text{Proj}(A, J)$, then AH^{-1} is integral.*

Proof. Since the rows of A belong to $\text{Proj}(A, J)$, there exists a $B \in \mathbb{Z}^{n\times n}$ such that $A = BH$, hence $AH^{-1} = B$ is integral. □

The next two lemmas follow directly from the definition of $\text{Proj}(A, J)$.

Lemma 8. *If $s \in \mathbb{Z}_{>0}$ is such that $sA^{-1}J$ is integral, and $P = \mathrm{Rem}(sA^{-1}J, s)$, then*

$$\mathrm{Proj}(A, J) = \mathrm{Proj}(sI, P) = \{v \in \mathbb{Z}^{1\times n} \mid \mathrm{Rem}(vP, s) = 0\}.$$

Lemma 9. *Let $U \in \mathbb{Z}^{n\times n}$ be unimodular. Then H is a basis of $\mathrm{Proj}(AU^{-1}, J)$ if and only if HU is a basis of $\mathrm{Proj}(A, J)$.*

Our final lemma will be used in the following sections to design an algorithm for computing a Smith massager.

Lemma 10. *Suppose the Smith form of A is $S = \mathrm{diag}(S_2, S_1)$, where $S_1 \in \mathbb{Z}^{r\times r}$ and $S_2 \in \mathbb{Z}^{(n-r)\times(n-r)}$. If $U \in \mathbb{Z}^{n\times n}$ is unimodular such that $H = \mathrm{diag}(I_{n-r}, S_1)$ is a basis of $\mathrm{Proj}(AU^{-1}, J)$, then the Smith form of $AU^{-1}H^{-1}$ is $\mathrm{diag}(I_r, S_2)$.*

5 MAXIMAL INDEX SMITH MASSAGER

In this section we combine all the results from the previous sections to present a randomized algorithm for the Problem `IndexMassager` shown in Figure 1. We begin with the following definition.

Definition 11 (Index-(m, r) Smith massager). *Let $B \in \mathbb{Z}^{2n\times 2n}$ be nonsingular with the shape*

$$B = \begin{bmatrix} A & & * \\ & I_{n-m} & \\ * & & * \end{bmatrix}.$$

For $m, r \in \mathbb{Z}_{\geq 0}$ such that $m + r \leq n$, an index-(m, r) Smith massager *for B is a tuple $(U, M, T, S) \in (\mathbb{Z}^{r\times n}, \mathbb{Z}^{n\times r}, \mathbb{Z}^{r\times r}, \mathbb{Z}^{r\times r})$ such that the matrix*

$$C := B \begin{bmatrix} I_n & & & \\ U & I & & \\ & & I_r & \\ & & & I_m \end{bmatrix} \begin{bmatrix} I_n & & M & \\ & I & & \\ & & T & \\ & & & I_m \end{bmatrix} \begin{bmatrix} I_n & & & \\ & I & & \\ & & S^{-1} & \\ & & & I_m \end{bmatrix} \quad (18)$$

is integral, with S nonsingular and in Smith form, and T unit upper triangular. We say that (U, M, T, S) is maximal *for B if S is comprised of the r largest invariant factors of the Smith form of B.*

Notice that when $m = 0$ the matrix B is equal to $\mathrm{diag}(A, I_n)$. When, in addition, $r = n$, an index-(m, r) Smith massager for $\mathrm{diag}(A, I_n)$ corresponds to a Smith massager for A as defined in the introduction.

`IndexMassager`$(B, n, m, r, s, \epsilon)$

Input: B, n, m and r are as in Definition 11. In addition, $s \in \mathbb{Z}_{>0}$ and ϵ is such that $0 < \epsilon < 1$.

Output: An index-(m, r) Smith massager (U, M, T, S) for B with $T = I_r$, entries in U and M reduced modulo s, and S_{rr} a divisor of s.

Note: If s is a positive integer multiple of the largest invariant factor of B, and the last n rows and columns of B^{-1} are integral, then a maximal index-(m, r) Smith massager for B is returned with probability at least $1 - \epsilon$.

Figure 1: Problem `IndexMassager`

In the design of the algorithm we are assuming that s is a multiple of the largest invariant factor of B and that the last n rows and columns of B^{-1} are integral. If, during the course of the algorithm, we detect that either of these conditions is not satisfied then we simply return the trivial index-(m, r) Smith massager $(0_{r\times n}, 0_{n\times r}, I_r, I_r)$ in order to satisfy the output requirements of the problem.

As shown in Section 4, we can "massage" away a block of the largest invariant factors of B by computing a basis of $\mathrm{Proj}(B, J)$ for a well chosen

$$J := \begin{bmatrix} J_1 \\ \hline J_2 \end{bmatrix} \in \mathbb{Z}^{(n+n)\times r}.$$

Note that under the assumption that the last n columns of B^{-1} are integral, the basis $\mathrm{Proj}(B, J)$ will remain invariant of the choice of entries in the block $J_2 \in \mathbb{Z}^{n\times r}$. For this reason, we set J_2 to be the zero matrix. Entries in J_1 are chosen independently and uniformly at random from $\mathbb{Z}/(s)$.

Next, we use the algorithm supporting Theorem 2 to check if $sB^{-1}J$ is integral, and, if so, compute the projection

$$P := \mathrm{Rem}(sB^{-1}J, s) = \begin{bmatrix} P_1 \\ \hline P_2 \end{bmatrix} \in \mathbb{Z}/(s)^{(n+n)\times r}.$$

Under the assumption that the last n rows of B^{-1} are integral, we expect P_2 to be the $n \times r$ zero matrix. If $sB^{-1}J$ is determined not to be integral, or P_2 is not the zero matrix, then we abort and return the trivial index-(m, r) massager for B.

At this point, by Lemma 8, we have reduced the problem of computing a basis of $\mathrm{Proj}(B, J)$ to that of computing a basis of $\mathrm{Proj}(sI, P)$. A basis of $\mathrm{Proj}(sI, P)$ can be computed as follows. First, using the Smith form algorithm from [12, Section 7], compute matrices $U \in \mathbb{Z}^{r\times n}$ and $V \in \mathbb{Z}^{r\times r}$, such that $\det V \perp s$ and $D := \mathrm{Rem}(-UP_1V, s)$ is congruent to the reverse Smith form of $P_1 \in \mathbb{Z}^{n\times r}$ over $\mathbb{Z}/(s)$. Then, we have the relations

$$-UP_1V = D \bmod s \quad \text{and} \quad P_1V = MD \bmod s,$$

for some integer matrix $M \in \mathbb{Z}^{n\times r}$. We can put those two together and obtain

$$\begin{bmatrix} I_n & & & \\ -U & I & & \\ & & I_r & \\ & & & I_m \end{bmatrix} \begin{bmatrix} P_1 \\ \\ \\ \end{bmatrix} V = \begin{bmatrix} MD \\ \\ D \\ \\ \end{bmatrix} \bmod s.$$

Next, we apply a unimodular left transformation to zero out the block MD, and we take right equivalence to omit V.

$$\begin{bmatrix} I_n & & -M & \\ & I & & \\ & & I_r & \\ & & & I_m \end{bmatrix} \begin{bmatrix} I_n & & & \\ -U & I & & \\ & & I_r & \\ & & & I_m \end{bmatrix} \begin{bmatrix} P_1 \\ \\ \\ \end{bmatrix} \equiv_R \begin{bmatrix} \\ D \\ \\ \end{bmatrix} \bmod s \quad (19)$$

Finally, define $S := sD^{-1}$, which will be in regular Smith form, and notice that S is a basis of $\mathrm{Proj}(S, I_r) = \mathrm{Proj}(sI, D)$, which corresponds to the non-zero part of the matrix in the right-hand side of (19). Therefore,

$$\begin{bmatrix} I_n & & & \\ & I & & \\ & & S & \\ & & & I_m \end{bmatrix} \begin{bmatrix} I_n & & -M & \\ & I & & \\ & & I_r & \\ & & & I_m \end{bmatrix} \begin{bmatrix} I_n & & & \\ -U & I & & \\ & & I_r & \\ & & & I_m \end{bmatrix} \quad (20)$$

must be a basis of $\mathrm{Proj}(sI, P)$ according to Lemma 9, since the two matrices containing M and U are unimodular. Postmultiplying B by the inverse of this basis results in an integer matrix according

to Lemma 7. That is,

$$C := B \begin{bmatrix} I_n & & & \\ & I & & \\ U & & I_r & \\ & & & I_m \end{bmatrix} \begin{bmatrix} I_n & & M & \\ & I & & \\ & & I_r & \\ & & & I_m \end{bmatrix} \begin{bmatrix} I_n & & & \\ & I & & \\ & & S^{-1} & \\ & & & I_m \end{bmatrix}.$$

Therefore, matrices (U, M, I_r, S) form an index-(m, r) Smith massager in accordance with Definition 11.

Theorem 12. *If $r \times \log s \in O(n(d + \log n))$, where d is the average of the lengths of the rows of B, and $\epsilon = \frac{1}{8r}$, then Problem* IndexMassager *can be solved in time $O(n^\omega \, \mathsf{B}(d + \log n) \log n)$.*

Proof. The correctness of the algorithm follows directly from the preceding discussion. The proposed massager fits the description of Definition 11. We achieve the probabilistic result, for $\epsilon = \frac{1}{8r}$, by exploiting Theorem 3. Instead of working with the projection $sB^{-1}J \in \mathbb{Z}^{2n\times r}$, we augment J with $k := \log_2 r + 4$ columns. After the Smith form computation, we keep only the last r rows of U, the last r columns of M, and the r largest invariant factors of S. This massager will be maximal with probability at least $1 - \frac{1}{2^{k-1}} = 1 - \frac{1}{8r}$.

Finally, regarding the running time, the algorithm consists of only two computational parts. The first is to test if $sB^{-1}J$ is integral, and, if so, compute P. By Theorem 2 this can be done in time $O(n^\omega \, \mathsf{M}(d + \log n) \log n)$. The second is the reverse Smith form computation: by [12, Corollary 7.17] which can be done in time $O(n^\omega \, \mathsf{B}(d + \log n) \log n)$, after simplifying the cost estimate using our assumptions on B. □

5.1 Reduced index Smith massager

In this subsection, we introduce the notion of the reduced Smith massager, which keeps the overall size of the matrices well bounded.

Denote by $U \bmod S$ the matrix obtained from U by reducing entries in row i modulo S_{ii}, $1 \le i \le r$. Similarly, we denote by $M \bmod S$ the matrix obtained from M by reducing entries in column j modulo S_{jj}, $1 \le j \le r$.

Definition 13 (Reduced index-(m, r) Smith massager). *Let (U, M, T, S) be an index-(m, r) Smith massager for $B \in \mathbb{Z}^{2n\times 2n}$ as in Definition 11. We say that (U, M, T, S) is* reduced *if $U = U \bmod S$, $M = M \bmod S$ and $T = ((T - I_r) \bmod S) + I_r$*

Lemma 14. *Suppose (U, M, T, S) is an index-(m, r) Smith massager for $B \in \mathbb{Z}^{2n\times 2n}$ as in Definition 11. Let $U' = U \bmod S$ and $M' = M \bmod S$. Let T' be the matrix obtained from $-U'M' \bmod S$ except with diagonal entry T'_{ii} reset to 1 when $S_{ii} = 1$, $1 \le i \le r$. Then (U, M', T, S) and (U, M, T', S) are index-(m, r) massagers for B, and (U', M', T', S) is a reduced index-(m, r) Smith massager for B.*

Proof. Without loss of generality, in order to simplify the presentation, we consider the case of an index-$(0, m)$ Smith massager. By multiplying together the first three matrices in (21) we obtain

$$B = \begin{bmatrix} A & & & AM \\ & I & & \\ & & I_r & \\ U & & & (T + UM) \end{bmatrix} \begin{bmatrix} I_n & & & \\ & I & & \\ & & I_r & \\ & & & S^{-1} \end{bmatrix}.$$

Note that the property that AMS^{-1} is integral is equivalent to $AM \bmod S$ being the zero matrix. But then $A(M \bmod S) \bmod S$ is also the zero matrix. This shows that (U, M', T, S) is an index massager. A similar argument shows that (U, M, T', S) is an index massager. By the definition of T',

$$(T' + (U \bmod S)(M \bmod S)) \bmod S$$

is also the zero matrix. Since T is unit upper triangular and also $(T + UM) \bmod S$ is the zero matrix, we have that $-UM \bmod S$ is unit upper triangular, except that the i'th diagonal entry will be zero for $S_{ii} = 1$. Using the property that $S_{11} \mid S_{22} \mid \cdots \mid S_{mm}$ it follows that T' is also unit upper triangular. □

6 MAXIMAL SMITH MASSAGER

In this section we develop a randomized algorithm for computing a Smith massager for a nonsingular $A \in \mathbb{Z}^{n\times n}$. Section 6.1 gives a subroutine for combining an index-$(0, m)$ and index-(m, r) Smith massager to obtain an index-$(0, m + r)$ Smith massager. The algorithm is given in Section 6.2 with a proof of correctness and running time given in Sections 6.3 and 6.4, respectively.

6.1 Combining index massagers

We show how an index-$(0, n)$ Smith massager for $\mathrm{diag}(A, I_n)$ can be computed in a block iterative fashion. Suppose we have an index-$(0, m)$ Smith massager (U, M, T, S) for $\mathrm{diag}(A, I_n)$. Then

$$B := \begin{bmatrix} A & & & \\ & I & & \\ & & I_r & \\ & & & I_m \end{bmatrix} \begin{bmatrix} I_n & & & \\ & I & & \\ & & I_r & \\ U & & & I_m \end{bmatrix} \begin{bmatrix} I_n & & & M \\ & I & & \\ & & I_r & \\ & & & T \end{bmatrix} \begin{bmatrix} I_n & & & \\ & I & & \\ & & I_r & \\ & & & S^{-1} \end{bmatrix} \quad (21)$$

is integral. Let (U', M', T', S') be an index-(m, r) Smith massager for B. Then

$$C := B \begin{bmatrix} I_n & & & \\ & I & & \\ U' & & I_r & \\ & & & I_m \end{bmatrix} \begin{bmatrix} I_n & & M' & \\ & I & & \\ & & T' & \\ & & & I_m \end{bmatrix} \begin{bmatrix} I_n & & & \\ & I & & \\ & & S'^{-1} & \\ & & & I_m \end{bmatrix} \quad (22)$$

is integral. A direct computation shows that the product of the first trio of matrices post-multiplying $\mathrm{diag}(A, I_n)$ in (21) with the second trio of matrices post-multiplying B in (22) is equal to

$$\begin{bmatrix} I_n & & & \\ & I & & \\ U' & & I_r & \\ U & & & I_m \end{bmatrix} \begin{bmatrix} I_n & & M' & M \\ & I & & \\ & & T' & -U'M \\ & & & T \end{bmatrix} \begin{bmatrix} I_n & & & \\ & I & & \\ & & S'^{-1} & \\ & & & S^{-1} \end{bmatrix}. \quad (23)$$

Thus, the result of post-multiplying $\mathrm{diag}(A, I_n)$ by the combined trio in (23) is integral. The next result follows as a result of the above discussion and as a corollary of Lemma 14.

Theorem 15. *Let (U, M, T, S) be a reduced index-$(0, m)$ Smith massager for $\mathrm{diag}(A, I_n)$, and let (U', M', T', S') be a reduced index-(m, r) Smith massager for the matrix B in (21). If S'_{rr} is a divisor of S_{11}, then a reduced index-$(0, m + r)$ Smith massager for $\mathrm{diag}(A, I_n)$ is given by (U'', M'', T'', S'') where*

$$U'' = \left[\begin{array}{c} U' \\ \hline U \end{array}\right], \; M'' = \left[\begin{array}{c|c} M' & M \end{array}\right], \; S'' = \left[\begin{array}{c|c} S' & \\ \hline & S \end{array}\right],$$

and

$$T'' = \left[\begin{array}{c|c} T' & -U'M \bmod S \\ \hline & T \end{array}\right].$$

6.2 Maximal Smith massager algorithm

Algorithm SmithMassager(A) is shown in Figure 2. For convenience, assume for the moment that n is equal to one less than a power of two. Phase 2 of the algorithm initializes $B := \mathrm{diag}(A, I_n)$ and $(U, M, T, S) \in (\mathbb{Z}^{0\times n}, \mathbb{Z}^{n\times 0}, \mathbb{Z}^{0\times 0}, \mathbb{Z}^{0\times 0})$ to be the trivial index-$(0,0)$ Smith massager, and then uses $\log_2(n+1)$ applications of Theorem 15 to update (U, M, T, S) to be an index-$(0, n)$ Smith massager for $\mathrm{diag}(A, I_n)$. The technique of Lemma 14 is used to keep the intermediate index massagers reduced. At the beginning of iteration i of the for-loop, (U, M, T, S) is a reduced index-$(0, m)$ Smith massager where $m = 2^{i-1} - 1$. Iteration i then updates (U, M, T, S) to be a reduced index-$(0, m + r)$ Smith massager where $r = 2^{i-1}$.

At the end of phase 2, the algorithm has computed a Smith massager (U, M, T, S) for A. It remains only to assay if (U, M, T, S) is maximal. This is done by checking that the massaged matrix B is unimodular.

SmithMassager(A)
Input: Nonsingular $A \in \mathbb{Z}^{n\times n}$ with $\det A \perp 2$.
Output: A reduced maximal Smith massager for A or FAIL.
Note: FAIL will be returned with probability less than $1/2$.

(1) [Compute the largest invariant factor of A]
$s :=$ the largest invariant factor s_n of A
s may be a proper divisor of s_n with probability $\le 1/4$.

(2) [Compute an index-$(0, n)$ Smith massager for $\mathrm{diag}(A, I_n)$]
$(U, M, T, S) \in (\mathbb{Z}^{0\times n}, \mathbb{Z}^{n\times 0}, \mathbb{Z}^{0\times 0}, \mathbb{Z}^{0\times 0})$
$B := \mathrm{diag}(A, I_n)$
for $i = 1$ **to** $\lceil \log_2(n+1) \rceil$ **do**
$m := 2^{i-1} - 1$
$r := \min(2^{i-1}, n - m)$
if $i > 1$ **then** $s := S_{11}$

(a) [Compute an index-(m, r) massager of B and reduce]
$(U', M', I, S') :=$ IndexMassager$(B, m, r, s, 2^{-(i+2)})$
$U', M' := U' \operatorname{rmod} S', M' \operatorname{cmod} S'$
$T' := -U'M' \operatorname{cmod} S'$, 0 diagonal entries replaced by 1

(b) [Augment massager and reduce]
$$U, M, S := \left[\begin{array}{c} U' \\ \hline U \end{array}\right], \left[\begin{array}{c|c} M' & M \end{array}\right], \left[\begin{array}{c|c} S' & \\ \hline & S \end{array}\right]$$
$$T := \left[\begin{array}{c|c} T' & -U'M \operatorname{cmod} S \\ \hline & T \end{array}\right]$$

(c) [Apply massager]
$$B := \begin{bmatrix} A & & AMS^{-1} \\ & I & \\ U & & (T + UM)S^{-1} \end{bmatrix}$$

(3) [Certify that (U, M, T, S) is maximal]
if $|\det B| = 1$ **then return** (U, M, T, S)
else return FAIL

Figure 2: Algorithm SmithMassager

6.3 Correctness

We begin with two lemmas regarding properties of the massaged matrix B in phase 2(c) of the algorithm. Lemma 16 is a corollary of Lemma 10.

Lemma 16. *If (U, M, T, S) is a maximal index-$(0, m)$ Smith massager for $\mathrm{diag}(A, I_n)$ with Smith form $\mathrm{diag}(I_n, S', S)$, then the Smith form of the massaged matrix $B \in \mathbb{Z}^{2n\times 2n}$ as in (21) is $\mathrm{diag}(I_n, I_m, S')$.*

Lemma 17. *If (U, M, T, S) is a maximal index-$(0, m)$ Smith massager for $\mathrm{diag}(A, I_n)$ and $B \in \mathbb{Z}^{2n\times 2n}$ the massaged matrix as in (21), then the last n rows and columns of B^{-1} are integral.*

Proof. Notice that the augmenting operation of Theorem 15 can also be reversed to separate a Smith massager. We will use induction on m. The base case, for $m = 0$, holds vacuously. Next, assume that the statement of the lemma holds for a maximal index-$(0, m)$ Smith massager (U, M, T, S). This means that: (1) the largest invariant factor of the massaged matrix B is s_{n-m} according to Lemma 16, and: (2) the last n rows of B^{-1} are integral according to the induction hypothesis. Now, let (U', M', T', S') be a maximal index-$(m, 1)$ Smith massager for B. The product of the trio of matrices defined by (U, M, T, S) and the product defined by (U', M', T', S') correspond to a trio of matrices defined by a maximal index-$(0, m + 1)$ Smith massager. The inverse of the massaged matrix C will be

$$\begin{bmatrix} I_n & & & \\ & I & & \\ & & s_{n-m} & \\ & & & I_m \end{bmatrix} \begin{bmatrix} I_n & & -M' & \\ & I & & \\ & & 1 & \\ & & & I_m \end{bmatrix} \begin{bmatrix} I_n & & & \\ & I & & \\ -U' & & 1 & \\ & & & I_m \end{bmatrix} B^{-1}.$$

We see that the largest invariant factor of the product of the last three matrices is still s_{n-m}. In addition, the row of the product that is multiplied with s_{n-m}, is the only one from the last n rows of B^{-1} to which elements from the non-integral part of B^{-1} are added. Of course, multiplying with the matrix's largest invariant factor ensures that the last n rows of C^{-1} remain integral.

Finally, the last n columns are necessarily integral since they are the product of integral parts. □

Theorem 18. *Algorithm* SmithMassager *shown in Figure 2 is correct. The algorithm returns* FAIL *with probability less than* $1/2$.

Proof. The correctness of the algorithm is certified by the unimodularity check in phase 3. Regarding the probability of success, we define the following events.

- E_0: In phase 1, s is not the largest invariant factor s_n of A.
- E_i: At iteration $i = 1, \ldots, \lceil \log_2(n+1) \rceil$ of phase 2, massager (U, M, T, S) is not maximal.

In other words, in order to prove the theorem, it is enough to show that $\Pr[E_{\lceil \log_2(n+1) \rceil}] < 1/2$. From the specification of phase 1, the routine IndexMassager, and Lemma 17, we obtain that

$$\Pr[E_0] \le \frac{1}{4} \quad \text{and} \quad \Pr[E_i | \neg E_{i-1}] \le 2^{-(i+2)}.$$

Furthermore,

$$\begin{aligned} \Pr[E_i] &= \Pr[E_i|\neg E_{i-1}] \Pr[\neg E_{i-1}] + \Pr[E_i|E_{i-1}] \Pr[E_{i-1}] \\ &\le \Pr[E_i|\neg E_{i-1}] \cdot 1 + 1 \cdot \Pr[E_{i-1}] \\ &\le 2^{-(i+2)} + \Pr[E_{i-1}]. \end{aligned}$$

So, Algorithm SmithMassager returns FAIL with probability less than

$$\Pr[E_{\lceil \log_2(n+1) \rceil}] \le \sum_{i=1}^{\lceil \log_2(n+1) \rceil} 2^{-(i+2)} + \Pr[E_0] < \sum_{i=1}^{\infty} \frac{1}{2^{i+2}} + \frac{1}{4} = \frac{1}{2}.$$

□

6.4 Complexity

We begin by bounding the cost of phases 2(b) and 2(c). Lemma 19 presents a subroutine that computes matrix B in phase 2(c), and Lemma 20 a subroutine that realizes the construction of Theorem 15 in phase 2(b).

Lemma 19. *There exists a procedure that takes a reduced index-$(0,m)$ Smith massager (U,M,T,S) for $\mathrm{diag}(A,I_n)$, and returns a matrix B as in (21). The running time of the procedure is $O(n^\omega\,\mathsf{M}(\log n+\log||A||))$.*

Proof. It is enough to prove the claim for $m=n$. We have that

$$B=\begin{bmatrix} A & AMS^{-1} \\ U & (T+UM)S^{-1} \end{bmatrix}.$$

The cost-dominating operation is the product $UM\in\mathbb{Z}^{n\times n}$.

Recall that $(\det S)\mid(\det A)$, and that the entries in row i of matrix U and in column i of matrix M are reduced modulo S_{ii}. Thus, for an $X:=2^h\geq\lceil n^{1/2}||A||\rceil$, the matrices U and M can be written as their X-adic expansions (of length n),

$$U=U_0+\cdots+U_{n-1}X^{n-1}\text{ and }\ M=M_0+\cdots+M_{n-1}X^{n-1},$$

with U_i and M_i reduced modulo X. This gives the following expression for their product: $UM=\sum_{i,j=0}^{n-1}U_iM_jX^{i+j}$. Of course, it is impossible to perform n^2 matrix multiplications within our target complexity. Instead, we observe that since the rows or columns of both matrices are reduced modulo the invariant factors of A, then the higher-order coefficients of the expansion must be sparser.

According to [1, Lemma 17], if we remove the top zero rows of each $U_i\in\mathbb{Z}/(X)^{n\times n}$ to obtain a $\bar{U}_i\in\mathbb{Z}/(X)^{*\times n}$, then the matrix

$$\bar{U}:=\begin{bmatrix} \bar{U}_0 \\ \hline \vdots \\ \hline \bar{U}_{n-1} \end{bmatrix}$$

has at most $2n$ rows. The same holds for the number of columns of the matrix

$$\bar{M}:=\left[\begin{array}{c|c|c} \bar{M}_0 & \cdots & \bar{M}_{n-1} \end{array}\right],$$

where each $\bar{M}_i\in\mathbb{Z}/(X)^{n\times *}$ is produced by removing the leading zero columns from each component of the X-adic expansion of M. We can multiply $\bar{U}$ and $\bar{M}$ in $O(n^\omega\,\mathsf{M}(\log n+\log||A||)$ and obtain the (at most) $2n\times 2n$ matrix

$$\begin{bmatrix} \bar{U}_0\bar{M}_0 & \cdots & \bar{U}_0\bar{M}_{n-1} \\ \vdots & \ddots & \vdots \\ \bar{U}_{n-1}\bar{M}_0 & \cdots & \bar{U}_{n-1}\bar{M}_{n-1} \end{bmatrix}. \tag{24}$$

The above matrix contains the result of all the U_iM_j products, as they are equal to $\bar{U}_i\bar{M}_j$ along with some additional zero rows and columns. Finally, after multiplying each $\bar{U}_i\bar{M}_j$ with X^{i+j}, we compute UM by adding all the products together, while taking into account their additional zero rows and columns. □

Lemma 20. *The reduced index-$(0,m+r)$ massager of Theorem 15 can be computed in time $O(n^\omega\,\mathsf{M}(\log n+\log||A||))$.*

Proof. The only nontrivial computation of Theorem 15 is the product $U'M\in\mathbb{Z}^{r\times m}$, and the required complexity can be achieved by following the same technique as in Lemma 19. □

Theorem 21. *The running time of the Algorithm* `SmithMassager` *shown in Figure 2 is $O(n^\omega\,\mathrm{B}(\log n+\log||A||)\,(\log n)^2)$.*

Proof. Phase 1 is done in time $O(n^\omega\,\mathrm{B}(\log n+\log||A||)\log n)$ using the Monte Carlo approach of [4, Theorem 2.1] combined with fast linear system solving [1, Corollary 7] and rational number reconstruction. Phase 2 consists of $O(\log n)$ iterations of the `IndexMassager` algorithm. Since matrix U is always reduced row modulo S, the average of the lengths of the rows of U, and consequently of B, is $O(\log n+\log||A||)$. Hence, phase 2 requires time $O(n^\omega\,\mathrm{B}(\log n+\log||A||)(\log n)^2)$. Finally, according to [11, Section 4], the unimodularity check in phase 3 can be performed in time $O(n^\omega\,\mathsf{M}(\log n+\log||A||)\log n)$. □

7 AN ALGORITHM FOR SMITH FORM

Given a nonsingular input matrix $A\in\mathbb{Z}^{n\times n}$, we first compute [1] the 2-Smith form $S_{\mathsf{even}}=\mathrm{diag}(2^{e_1},\ldots,2^{e_n})$ of A, where 2^{e_i} is the largest power of 2 that divides the i-th invariant factor of A, together with A_{odd} such that $\det A_{\mathsf{odd}}\perp 2$, $||A_{\mathsf{odd}}||\leq n||A||$, and $A_{\mathsf{odd}}S_{\mathsf{even}}\equiv_R A$. Then, we compute the Smith form S_{odd} of A_{odd} with Algorithm `SmithMassager` and return $S_{\mathsf{odd}}S_{\mathsf{even}}$.

We remark that the algorithms in [1] were analysed under a more restrictive cost model than the one used in this paper. Replacing the subroutine in [1, Section 6] with the integer analogue of the algorithm supporting [9, Theorem 7] allows A_{odd} and S_{even} to be computed in $O(n^\omega\,\mathsf{M}(\log n+\log||A||)\,(\log n)^2)$ bit operations.

Theorem 22. *There exists a Las Vegas probabilistic algorithm that computes the Smith form of a nonsingular $A\in\mathbb{Z}^{n\times n}$ using $O(n^\omega\,\mathrm{B}(\log n+\log||A||)\,(\log n)^2)$ bit operations.*

REFERENCES

[1] S. Birmpilis, G. Labahn, and A. Storjohann. Deterministic reduction of integer nonsingular linear system solving to matrix multiplication. In *Proc. Int'l. Symp. on Symbolic and Algebraic Computation: ISSAC'19*. ACM Press, New York, 2019.

[2] J. Blömer, R. Karp, and E. Welzl. The rank of sparse random matrices over finite fields. *Random Structures and Algorithms*, 10(4):407–419, July 1997.

[3] C. Cooper. On the distribution of rank of a random matrix over a finite field. *Random Structures and Algorithms*, 17(3-4):197–212, oct 2000.

[4] W. Eberly, M. Giesbrecht, and G. Villard. Computing the determinant and Smith form of an integer matrix. In *Proc. 31st Ann. IEEE Symp. Foundations of Computer Science*, pages 675–685, 2000.

[5] F. L. Gall. Powers of tensors and fast matrix multiplication. In *Proc. Int'l. Symp. on Symbolic and Algebraic Computation: ISSAC'14*. ACM Press, New York, 2014.

[6] F. L. Gall and F. Urrutia. Improved rectangular matrix multiplication using powers of the Coppersmith-Winograd tensor. In A. Czumaj, editor, *Proceedings of the Twenty-Ninth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2018, New Orleans, LA, USA, January 7-10, 2018*, pages 1029–1046. SIAM, 2018.

[7] J. von zur Gathen and J. Gerhard. *Modern Computer Algebra*. Cambridge University Press, 3rd edition, 2013.

[8] M. Giesbrecht. Fast computation of the Smith form of a sparse integer matrix. *Computational Complexity*, 10(1):41–69, 11 2001.

[9] S. Gupta, S. Sarkar, A. Storjohann, and J. Valeriote. Triangular x-basis decompositions and derandomization of linear algebra algorithms over $\mathsf{K}[x]$. *Journal of Symbolic Computation*, 47(4), 2012. Festschrift for the 60th Birthday of Joachim von zur Gathen.

[10] E. Kaltofen and G. Villard. On the complexity of computing determinants. *Computational Complexity*, 13(3–4):91–130, 2004.

[11] C. Pauderis and A. Storjohann. Deterministic unimodularity certification. In *Proc. Int'l. Symp. on Symbolic and Algebraic Computation: ISSAC'12*, page 281–288. ACM Press, New York, 2012.

[12] A. Storjohann. *Algorithms for Matrix Canonical Forms*. PhD thesis, Swiss Federal Institute of Technology, ETH–Zurich, 2000.

[13] A. Storjohann. The shifted number system for fast linear algebra on integer matrices. *Journal of Complexity*, 21(4):609–650, 2005. Festschrift for the 70th Birthday of Arnold Schönhage.

Computing the N-th Term of a q-Holonomic Sequence

Alin Bostan
Inria, France

ABSTRACT

In 1977, Strassen invented a famous baby-step / giant-step algorithm that computes the factorial $N!$ in arithmetic complexity quasi-linear in $\sqrt{N}$. In 1988, the Chudnovsky brothers generalized Strassen's algorithm to the computation of the N-th term of any holonomic sequence in the same arithmetic complexity. We design q-analogues of these algorithms. We first extend Strassen's algorithm to the computation of the q-factorial of N, then Chudnovskys' algorithm to the computation of the N-th term of any q-holonomic sequence. Both algorithms work in arithmetic complexity quasi-linear in $\sqrt{N}$. We describe various algorithmic consequences, including the acceleration of polynomial and rational solving of linear q-differential equations, and the fast evaluation of large classes of polynomials, including a family recently considered by Nogneng and Schost.

CCS CONCEPTS

• **Computing methodologies** → **Algebraic algorithms**.

KEYWORDS

Algorithms, complexity, q-factorial, q-holonomic sequences.

ACM Reference Format:
Alin Bostan. 2020. Computing the N-th Term of a q-Holonomic Sequence. In *International Symposium on Symbolic and Algebraic Computation (ISSAC '20), July 20–23, 2020, Kalamata, Greece.* ACM, New York, NY, USA, 8 pages. https://doi.org/10.1145/3373207.3404060

1 INTRODUCTION

A classical question in algebraic complexity theory is: how fast can one evaluate a univariate polynomial at one point? The precise formulation of this question depends on the model of computation. We will mainly focus on the *arithmetic complexity* model, in which one counts base field operations at unit cost.

Horner's rule evaluates a polynomial P in $O(\deg(P))$ operations. Ostrowski [55] conjectured in 1954 that this is *optimal for generic polynomials* (*i.e.*, whose coefficients are algebraically independent). This optimality result was proved a few years later by Pan [57].

However, most polynomials that one might wish to evaluate have coefficients which are not algebraically independent. Paterson and Stockmeyer [58] showed that for any field $\mathbb{K}$, an arbitrary polynomial $P \in \mathbb{K}[x]$ of degree n can be evaluated using $O(\sqrt{n})$ *nonscalar* multiplications; however, their algorithm uses a linear amount of scalar multiplications, so it is not well adapted to the evaluation at points from the base field $\mathbb{K}$, since in this case the total arithmetic complexity remains linear in n.

On the other hand, for some families of polynomials, one can do much better. Typical examples are x^n and $P_n(x) := x^{n-1} + \cdots + x + 1$, which can be evaluated by repeated squaring in $O(\log n)$ operations. (Note that for $P_n(x)$ such a fast algorithm needs to perform division.) By contrast, a family $F_n(x)$ of univariate polynomials is called *hard to compute* if the complexity of the evaluation of F_n grows at least like a power in $\deg(F_n)$, whatever the algorithm used.

Paterson and Stockmeyer [58] proved the existence of polynomials in $\mathbb{K}[x]$ which are hard to compute. Specific families of hard-to-compute polynomials were first exhibited by Strassen [70]. The techniques were refined and improved by Borodin and Cook [13], Lipton [52] and Schnorr [66], who produced explicit examples of degree-n polynomials whose evaluation requires a number of operations linear in $\sqrt{n}$. Subsequently, various methods have been developed to produce similar results on *lower bounds*, *e.g.*, by Heintz and Sieveking [42] using algebraic geometry, and by Aldaz et al. [4] using a combinatorial approach. The topic is vast and very well summarized in the book by Bürgisser, Clausen and Shokrollahi [23].

In this article, we focus on *upper bounds*, that is on the design of fast algorithms for special families of polynomials, which are hard to compute, but easier to evaluate than generic polynomials. For instance, for the degree-$\binom{n}{2}$ polynomial $Q_n(x) := P_1(x) \cdots P_n(x)$, a complexity in $O(n)$ is clearly achievable. We will see in §2.1 that one can do better, and attain a cost which is almost linear in $\sqrt{n}$ (up to logarithmic factors in n). Another example is $R_n(x) := \sum_{k=0}^{n} x^{k^2}$, of degree n^2, and whose evaluation can also be performed in complexity quasi-linear in $\sqrt{n}$, as shown recently by Nogneng and Schost [54] (see §2.2). In both cases, these complexities are obtained by clever although somehow ad-hoc algorithms. The starting point of our work was the question whether these algorithms for $Q_n(x)$ and $R_n(x)$ could be treated in a unified way, which would allow to evaluate other families of polynomials in a similar complexity.

The answer to this question turns out to be positive. The key idea, very simple and natural, is to view both examples as particular cases of the following general question: given a q-holonomic sequence, that is, a sequence satisfying a linear recurrence with polynomial coefficients in q and q^n, how fast can one compute its N-th term?

In the more classical case of holonomic sequences (satisfying linear recurrences with polynomial coefficients in the index n), fast algorithms exist for the computation of the N-th term. They rely on a basic block, which is the computation of the factorial term $N!$ in arithmetic complexity quasi-linear in $\sqrt{N}$, using an algorithm due to Strassen [71]. The Chudnovsky brothers extended in [26] Strassen's algorithm to the computation of the N-th term of any holonomic sequence in arithmetic complexity quasi-linear in $\sqrt{N}$.

Our main contribution consists in transferring these results to the q-holonomic framework. It turns out that the resulting algorithms are actually simpler in the q-holonomic case than in the usual holonomic setting, essentially because multipoint evaluation on

ISSAC '20, July 20–23, 2020, Kalamata, Greece

ACM ISBN 978-1-4503-7100-1/20/07…$15.00
https://doi.org/10.1145/3373207.3404060

arithmetic progressions used as a subroutine in Strassen's and Chudnovskys' algorithms is replaced by multipoint evaluation on geometric progressions, which is considerably simpler [21].

A consequence of our results is that the following apparently unrelated polynomials / rational functions can be evaluated fast (note the change in notation, with the variable x denoted now by q):

- $A_n(q)$, the generating function of the number of partitions into n positive integers each occurring *at most twice* [75], *i.e.*, the coefficient of t^n in the product $\prod_{k\geq 1}(1+q^k t+q^{2k}t^2)$.
- $B_n(q) := \prod_{i=1}^{\infty}(1-q^i) \bmod q^n$; by Euler's pentagonal theorem [56, §5], $B_n(q) = 1 + \sum_{i(3i+1)<2n} (-1)^i \left(q^{\frac{i(3\,i-1)}{2}} + q^{\frac{i(3\,i+1)}{2}}\right)$.
- The number $C_n(q)$ of $2n \times 2n$ upper-triangular matrices over $\mathbb{F}_q$ (the finite field with q elements), whose square is the zero matrix; by [47], $C_n(q)$ is equal to

$$C_n(q) = \sum_j \left[\binom{2n}{n-3j} - \binom{2n}{n-3j-1}\right] \cdot q^{n^2-3j^2-j}.$$

The common feature, exploited by the new algorithm, is that the sequences $(A_n(q))_{n\geq 0}, (B_n(q))_{n\geq 0}, (C_n(q))_{n\geq 0}$ are all q-holonomic. Actually, q-holonomic sequences are ubiquitous, so the range of application of our results is quite broad. This stems from the fact that they are coefficient sequences of power series satisfying q-differential equations, or equivalently, q-shift (or, q-difference) equations. From that perspective, our topic becomes intimately connected with q-calculus. The roots of q-calculus are in works of famous mathematicians such as Rothe, Gauss and Heine. The topic gained renewed interest in the first half of the 20th century, with the work, both on the formal and analytic aspects, of Tanner, Jackson, Carmichael, Mason, Adams, Trjitzinsky, Le Caine and Hahn, to name just a few. Modern accounts of the various aspects of the theory (including historical ones) can be found in [30, 32, 48].

One of the reasons for interest in q-difference equations is that, formally, as q tends to 1, the q-derivative $\frac{f(qx)-f(x)}{(q-1)x}$ tends to $f'(x)$, thus to every differential equation corresponds a q-differential equation which goes formally to the differential equation as $q \to 1$. In nice cases, (some of) the solutions of the q-differential equation go to solutions of the associated differential equation as $q \to 1$. An early example of such a good deformation behavior is given by the basic hypergeometric equation of Heine [48, §1.10].

In computer algebra, q-holonomic sequences were considered starting from the early nineties, in the context of computer-generated proofs of identities in the seminal paper by Wilf and Zeilberger [74], notably in Section 5 ("Generalization to q-sums and q-multisums") and in Section 6.4 ("q-sums and integrals"). Creative telescoping algorithms for (proper) q-hypergeometric sequences are discussed in various references [12, 25, 61]; several implementations of those algorithms are described for instance in [45, 60, 64, 69]. Algorithms for computing polynomial, rational and q-hypergeometric solutions of q-differential equations were designed by Abramov and collaborators [1–3, 46]. These algorithms are important for several reasons. One is that they lie at the heart of the vast generalization by Chyzak [27, 28] of the Wilf and Zeilberger algorithmic theory, for the treatment of general q-holonomic (not only q-hypergeometric) symbolic summation and integration via creative telescoping. In that context, a multivariate notion of q-holonomy is needed; the foundations of the theory were laid by Zeilberger [77] and Sabbah [65] (in the language of D-modules), see also [25, § 2.5] and [37].

The simplest non-trivial holonomic sequence is $n!$, which combinatorially counts the number of permutations of n objects. If instead of direct counting, one assigns to every permutation π its number of inversions $\mathrm{inv}(\pi)$, *i.e.*, the number of pairs $1 \leq i < j \leq n$ with $\pi(i) > \pi(j)$, the refined count (by size and number of inversions) is $[n]_q! := (1+q)(1+q+q^2)\cdots(1+q+\cdots+q^{n-1})$. This is the q-analogue of $n!$, the simplest non-trivial q-holonomic sequence.

There is also a natural q-analog of the binomial coefficients, called the *Gaussian coefficients*, defined by $\binom{n}{k}_q := \frac{[n]_q!}{[k]_q![n-k]_q!}$. They have many counting interpretations, *e.g.,* they count the k-dimensional subspaces of $\mathbb{F}_q^n$ (points on Grassmannians over $\mathbb{F}_q$). There are q-analogs to (almost) everything. To select just two basic examples, the q-analog [5, Thm. 3.3] of the binomial theorem is

$$\prod_{k=1}^{n}(1+q^{k-1}x) = \sum_{k=0}^{n}\binom{n}{k}_q q^{\binom{k}{2}}x^k \tag{1}$$

and the q-version [5, Thm. 3.4] of the Chu-Vandermonde identity is

$$\sum_{k=0}^{n} q^{k^2}\binom{m}{k}_q\binom{n}{k}_q = \binom{m+n}{n}_q. \tag{2}$$

The ubiquity of q-holonomic sequences is manifest in plenty of fields: partition theory [5, 56] and other subfields of combinatorics [33, 47]; theta functions and modular forms [51, 59, 76]; special functions [48] and in particular orthogonal polynomials [49]; algebraic geometry [31], representation theory [44]; knot theory [35–37]; Galois theory [43]; number theory [29].

The main message of this article is that for any example of q-holonomic sequence occurring in those various fields, *one can compute selected coefficients faster than by a direct algorithm.*

Complexity basics. We estimate the complexities of algorithms by counting arithmetic operations $(+,-,\times,\div)$ in the base field $\mathbb{K}$ at unit cost. We use standard complexity notation, such as $\mathbf{M}(d)$ for the cost of degree-d multiplication in $\mathbb{K}[x]$ and θ for feasible exponents of matrix multiplication. The best known upper bound is $\theta < 2.3729$ [34]. Most arithmetic operations on univariate polynomials of degree d in $\mathbb{K}[x]$ can be performed in quasi-linear complexity $\tilde{O}(d)$: multiplication, shift, interpolation, gcd, resultant, *etc*[*]. A key feature of these results is the reduction to fast polynomial multiplication, which can be performed in time $\mathbf{M}(d) = O(d \log d \log\log d)$ [24, 68]. An excellent general reference for these questions is the book by von zur Gathen and Gerhard [38].

2 TWO MOTIVATING EXAMPLES

Before presenting our main results in Section 3, we describe in this section the approach and main ideas on two basic examples. Both examples concern the fast evaluation of special families of univariate polynomials. In §2.1, we consider polynomials of the form $\prod_\ell (x-q^\ell)$, and in §2.2 sparse polynomials of the form $\sum_\ell p^\ell x^{a\ell^2+b\ell}$. In both cases, we first present fast ad-hoc algorithms, then introduce equally fast alternative algorithms, which have the nice feature that they will be generalizable to a broader setting.

[*]As usual, the notation $\tilde{O}(\cdot)$ is used to hide polylogarithmic factors in the argument.

2.1 De Feo's question

Here is our first example, emerging from a question asked to the author by Luca De Feo*; this was the starting point of the article.

Let q be an element of the field $\mathbb{K}$, and consider the polynomial

$$F(x) := \prod_{i=0}^{N-1} (x - q^i) \ \in \mathbb{K}[x]. \tag{3}$$

Given another element $\alpha \in \mathbb{K}$, how fast can one evaluate $F(\alpha)$?

If $q = 0$, then $F(\alpha) = \alpha^N$ can be computed in $O(\log N)$ operations in $\mathbb{K}$, by binary powering. We assume in what follows that q is nonzero. Obviously, a direct algorithm consists in computing the successive powers $q, q^2, \ldots, q^{N-1}$ using $O(N)$ operations in $\mathbb{K}$, then computing the elements $\alpha - q, \alpha - q^2, \ldots, \alpha - q^{N-1}$ in $O(N)$ more operations in $\mathbb{K}$, and finally returning their product. The total arithmetic cost of this algorithm is $O(N)$, linear in the degree of F.

Is it possible to do better? The answer is positive, as one can use the following *baby-step / giant-step* strategy, in which, in order to simplify things, we assume that N is a perfect square**, $N = s^2$:

Algorithm 1

(1) (Baby-step) Compute the values of $q, q^2, \ldots, q^{s-1}$, and deduce the coefficients of the polynomial $G(x) := \prod_{j=0}^{s-1}(x - q^j)$.
(2) (Giant-step) Compute $Q := q^s, Q^2, \ldots, Q^{s-1}$, and deduce the coefficients of the polynomial $H(x) := \prod_{k=0}^{s-1}(\alpha - Q^k \cdot x)$.
(3) Return the resultant $\mathrm{Res}(G, H)$.

By the basic property of resultants, the output of this algorithm is

$$\mathrm{Res}(G,H) = \prod_{j=0}^{s-1} H(q^j) = \prod_{j=0}^{s-1}\prod_{k=0}^{s-1} \left(\alpha - q^{sk+j}\right) = \prod_{i=0}^{N-1} (\alpha - q^i) = F(\alpha).$$

Using the fast subproduct tree algorithm [38, Algorithm 10.3], one can perform the baby-step (1) as well as the giant-step (2) in $O(\mathbf{M}(\sqrt{N})\log N)$ operations in $\mathbb{K}$, and by [38, Corollary 11.19] the same cost can be achieved for the resultant computation in step (3). Using fast polynomial multiplication, we conclude that $F(\alpha)$ can be computed in arithmetic complexity quasi-linear in $\sqrt{N}$.

It is possible to speed up the previous algorithm by a logarithmic factor in N using a slightly different scheme, still based on a *baby-step / giant-step* strategy, but exploiting the fact that the roots of F are in geometric progression. Again, we assume that $N = s^2$ is a perfect square. This alternative algorithm goes as follows. Note that it is very close in spirit to Pollard's algorithm [62, p. 523].

Algorithm 2

(1) (Baby-step) Compute $q, q^2, \ldots, q^{s-1}$, and deduce the coefficients of the polynomial $P(x) := \prod_{j=0}^{s-1}(\alpha - q^j \cdot x)$.
(2) (Giant-step) Compute $Q := q^s, Q^2, \ldots, Q^{s-1}$, and evaluate P simultaneously at $1, Q, \ldots, Q^{s-1}$.
(3) Return the product $P(1)P(Q)\cdots P(Q^{s-1})$.

Obviously, the output of this algorithm is

$$\prod_{k=0}^{s-1} P(Q^k) = \prod_{k=0}^{s-1}\prod_{j=0}^{s-1} (\alpha - q^j \cdot q^{sk}) = \prod_{i=0}^{N-1} (\alpha - q^i) = F(\alpha).$$

*Private (email) communication, 10 January, 2020.

**If N is not a perfect square, then one can compute $F(\alpha)$ as $F(\alpha) = F_1(\alpha)F_2(\alpha)$, where $F_1(\alpha) := \prod_{i=0}^{\lfloor\sqrt{N}\rfloor^2-1}(\alpha - q^i)$ is computed as in Algorithm 1, while $F_2(\alpha) := \prod_{i=\lfloor\sqrt{N}\rfloor^2}^{N-1}(\alpha - q^i)$ can be computed naively, since $N - \lfloor\sqrt{N}\rfloor^2 = O(\sqrt{N})$.

As pointed out in the remarks after the proof of [21, Lemma 1], one can compute $P(x) = P_s(x) = \prod_{j=0}^{s-1}(\alpha - q^j \cdot x)$ in step (1) without computing the subproduct tree, by using a divide-and-conquer scheme which exploits the fact that $P_{2t}(x) = P_t(x) \cdot P_t(q^t x)$ and $P_{2t+1}(x) = P_t(x) \cdot P_t(q^t x) \cdot (\alpha - q^{2t}x)$. The cost of this algorithm is $O(\mathbf{M}(\sqrt{N}))$ operations in $\mathbb{K}$. As for step (2), one can use the fast chirp transform algorithms of Rabiner, Schafer and Rader [63] and of Bluestein [11]. These algorithms rely on the following observation: writing $Q^{ij} = Q^{\binom{i+j}{2}} \cdot Q^{-\binom{i}{2}} \cdot Q^{-\binom{j}{2}}$ and $P(x) = \sum_{j=0}^{s} c_j x^j$ implies that the needed values $P(Q^i) = \sum_{j=0}^{s} c_j Q^{ij}, 0 \le i < s$, are

$$P(Q^i) = Q^{-\binom{i}{2}} \cdot \sum_{j=0}^{s} c_j Q^{-\binom{j}{2}} \cdot Q^{\binom{i+j}{2}}, \ 0 \le i < s,$$

in which the sum is simply the coefficient of x^{s+i} in the product

$$\left(\sum_{j=0}^{s} c_j Q^{-\binom{j}{2}} x^{s-j}\right)\left(\sum_{\ell=0}^{2s} Q^{\binom{\ell}{2}} x^{\ell}\right).$$

This polynomial product can be computed in $2\mathbf{M}(s)$ operations (and even in $\mathbf{M}(s) + O(s)$ using the transposition principle [20, 40], since only the median coefficients $x^s, \ldots, x^{2s-1}$ are actually needed). In conclusion, step (2) can also be performed in $O(\mathbf{M}(\sqrt{N}))$ operations in $\mathbb{K}$, and thus $O(\mathbf{M}(\sqrt{N}))$ is the total cost of this second algorithm.

We have chosen to detail this second algorithm for several reasons: not only because it is faster by a factor $\log(N)$ compared to the first one, but more importantly because it has a simpler structure, which will be generalizable to the general q-holonomic setting.

2.2 Evaluation of some sparse polynomials

Let us now consider the sequence of sparse polynomial sums

$$v_N^{(p,a,b)}(q) = \sum_{n=0}^{N-1} p^n q^{an^2+bn},$$

where $p \in \mathbb{K}$ and $a, b \in \mathbb{Q}$ such that $2a, a+b$ are both integers. Typical examples are (truncated) modular forms [59], which are ubiquitous in number theory [76] and combinatorics [5]. For instance, the *Jacobi theta function* ϑ_3 depends on two complex variables $z \in \mathbb{C}$, and $\tau \in \mathbb{C}$ with $\Im(\tau) > 0$, and it is defined by

$$\vartheta_3(z;\tau) = \sum_{n=-\infty}^{\infty} e^{\pi i(n^2\tau + 2nz)} = 1 + 2\sum_{n=1}^{\infty} \eta^n q^{n^2},$$

where $q = e^{\pi i \tau}$ is the nome ($|q| < 1$) and $\eta = e^{2\pi i z}$. Here, $\mathbb{K} = \mathbb{C}$. Another example is the *Dedekind eta function*, appearing in Euler's famous *pentagonal theorem* [56, §5], which has a similar form

$$q^{\frac{1}{24}} \cdot \left(1 + \sum_{n=1}^{\infty} (-1)^n \left(q^{\frac{n(3n-1)}{2}} + q^{\frac{n(3n+1)}{2}}\right)\right), \quad \text{with } q = e^{2\pi i \tau}.$$

Moreover, sums of the form $v_N^{(1,a,b)}(q) = \sum_{n=0}^{N-1} q^{an^2+bn}$, over $\mathbb{K} = \mathbb{Q}$ or $\mathbb{K} = \mathbb{F}_2$, crucially occur in a recent algorithm by Tao, Crott and Helfgott [72] for the efficient construction of prime numbers in given intervals, *e.g.*, in the context of effective versions of Bertrand's postulate. Actually, (the proof of) Lemma 3.1 in [72] contains the first sublinear complexity result for the evaluation of the sum $v_N^{(p,a,b)}(q)$ at an arbitrary point q; namely, the cost is $O(N^{\theta/3})$, where $\theta \in [2,3]$ is any feasible exponent for matrix multiplication.

Subsequently, Nogneng and Schost [54] designed a faster algorithm, and lowered the cost down to $\tilde{O}(\sqrt{N})$. Our algorithm is similar in spirit to theirs, as it also relies on a *baby-step / giant-step* strategy.

Let us first recall the principle of the Nogneng-Schost algorithm [54]. Assume as before that N is a perfect square, $N = s^2$. The starting point is the remark that

$$v_N^{(p,a,b)}(q) = \sum_{n=0}^{N-1} p^n q^{an^2+bn} = \sum_{k=0}^{s-1}\sum_{j=0}^{s-1} p^{j+sk} q^{a(j+sk)^2+b(j+sk)}$$

can be written

$$\sum_{k=0}^{s-1} p^{sk} q^{as^2k^2+bsk} \cdot P(q^{2ask}), \text{ where } P(y) := \sum_{j=0}^{s-1} p^j q^{aj^2+bj} y^j.$$

Therefore, the computation of $v_N^{(p,a,b)}(q)$ can be reduced essentially to the simultaneous evaluation of the polynomial P at $s = 1+\deg(P)$ points (in geometric progression), with arithmetic cost $O(\mathbf{M}(\sqrt{N}))$.

We now describe an alternative algorithm, of similar complexity $O(\mathbf{M}(\sqrt{N}))$, with a slightly larger constant in the big-Oh estimate, but whose advantage is its potential of generality.

Let us denote by $u_n(q)$ the summand $p^n q^{an^2+bn}$. Clearly, the sequence $(u_n(q))_{n\geq 0}$ satisfies the recurrence relation

$$u_{n+1}(q) = A(q, q^n) \cdot u_n(q), \quad \text{where} \quad A(x, y) := px^{a+b}y^{2a}.$$

As an immediate consequence, the sequence with general term $v_n(q) := \sum_{k=0}^{n-1} u_k(q)$ satisfies a similar recurrence relation

$$v_{n+2}(q) - v_{n+1}(q) = A(q, q^n) \cdot (v_{n+1}(q) - v_n(q)),$$

with initial conditions $v_0(q) = 0$ and $v_1(q) = 1$. This scalar recurrence of order two is equivalent to the first-order matrix recurrence

$$\begin{bmatrix} v_{n+2} \\ v_{n+1} \end{bmatrix} = \begin{bmatrix} A(q, q^n) + 1 & -A(q, q^n) \\ 1 & 0 \end{bmatrix} \times \begin{bmatrix} v_{n+1} \\ v_n \end{bmatrix}.$$

By unrolling this matrix recurrence, we deduce that

$$\begin{bmatrix} v_{n+1} \\ v_n \end{bmatrix} = M(q^{n-1}) \begin{bmatrix} v_n \\ v_{n-1} \end{bmatrix} = M(q^{n-1}) \cdots M(q)M(1) \begin{bmatrix} 1 \\ 0 \end{bmatrix},$$

where

$$M(x) := \begin{bmatrix} pq^{a+b}x^{2a} + 1 & -pq^{a+b}x^{2a} \\ 1 & 0 \end{bmatrix},$$

hence $v_N = \begin{bmatrix} 0 & 1 \end{bmatrix} \times M(q^{N-1}) \cdots M(q)M(1) \times \begin{bmatrix} 1 \\ 0 \end{bmatrix}$. Therefore, the computation of v_N reduces to the computation of the "matrix q-factorial" $M(q^{N-1}) \cdots M(q)M(1)$, which can be performed fast by using a *baby-step / giant-step* strategy similar to the one of the second algorithm in §2.1. Again, we assume for simplicity that $N = s^2$ is a perfect square. The algorithm goes as follows.

Algorithm 3 (matrix q-factorial)

(1) (Baby-step) Compute $q, q^2, \ldots, q^{s-1}$; deduce the coefficients of the polynomial matrix $P(x) := M(q^{s-1}x) \cdots M(qx)M(x)$.
(2) (Giant-step) Compute $Q := q^s, Q^2, \ldots, Q^{s-1}$, and evaluate (the entries of) $P(x)$ simultaneously at $1, Q, \ldots, Q^{s-1}$.
(3) Return the product $P(Q^{s-1}) \cdots P(Q)P(1)$.

By proceeding as in Algorithm 2 in §2.1, the complexity of Algorithm 3 already is quasi-linear in $\sqrt{N}$. However, its dependence in a, b is quite high (quasi-linear in a and b). If a and b are fixed and considered as $O(1)$ this dependence is invisible, but otherwise the following variant has the same complexity with respect to N, and a much better cost with respect to a and b. It is based on the simple observation that, if $\tilde{M}(x)$ denotes the polynomial matrix

$$\tilde{M}(x) := \begin{bmatrix} prx + 1 & -prx \\ 1 & 0 \end{bmatrix}, \text{ with } r := q^{a+b}, \tag{4}$$

and if $\tilde{q} := q^{2a}$, then the following matrix q-factorials coincide:

$$M(q^{N-1}) \cdots M(q)M(1) = \tilde{M}(\tilde{q}^{N-1}) \cdots \tilde{M}(\tilde{q})\tilde{M}(1).$$

Algorithm 4 (matrix q-factorial, variant)

(0) (Precomputation) Compute $r := q^{a+b}$, $\tilde{q} := q^{2a}$, and $\tilde{M}$ in (4).
(1) (Baby-step) Compute $\tilde{q}, \tilde{q}^2, \ldots, \tilde{q}^{s-1}$; deduce the coefficients of the polynomial matrix $\tilde{P}(x) := \tilde{M}(\tilde{q}^{s-1}x) \cdots \tilde{M}(\tilde{q}x)\tilde{M}(x)$.
(2) (Giant-step) Compute $\tilde{Q} := \tilde{q}^s, \tilde{Q}^2, \ldots, \tilde{Q}^{s-1}$, and evaluate (the entries of) $\tilde{P}(x)$ simultaneously at $1, \tilde{Q}, \ldots, \tilde{Q}^{s-1}$.
(3) Return the product $\tilde{P}(\tilde{Q}^{s-1}) \cdots \tilde{P}(\tilde{Q})\tilde{P}(1)$.

Using binary powering, the cost of the additional precomputation in step (0) is only logarithmic in a and b. In exchange, the new steps (2) and (3) are performed on matrices whose degrees do not depend on a and b anymore (in the previous, unoptimized, version the degrees of the polynomial matrices were linear in a and b). The total arithmetic cost with respect to N is still quasi-linear in $\sqrt{N}$.

3 MAIN RESULTS

In this section, we generalize the algorithms from §2, and show that they apply to the general setting of q-holonomic sequences.

3.1 Preliminaries

A sequence is q-holonomic if it satisfies a nontrivial q-recurrence, that is, a linear recurrence with coefficients polynomials in q and q^n.

Definition 3.1 (q-holonomic sequence). Let $\mathbb{K}$ be a field, and $q \in \mathbb{K}$. A sequence $(u_n(q))_{n\geq 0}$ in $\mathbb{K}^{\mathbb{N}}$ is called *q-holonomic* if there exist $r \in \mathbb{N}$ and polynomials $c_0, \ldots, c_r$ in $\mathbb{K}[x, y]$, with $c_r \neq 0$, such that

$$c_r(q, q^n)u_{n+r}(q) + \cdots + c_0(q, q^n)u_n(q) = 0, \quad \text{for all } n \geq 0. \tag{5}$$

The integer r is called the *order* of the q-recurrence (5). When $r = 1$, we say that $(u_n(q))_{n\geq 0}$ is q-hypergeometric.

The most basic examples are the *q-bracket* and the *q-factorial*,

$$[n]_q := 1 + q + \cdots + q^{n-1} \quad \text{and} \quad [n]_q! := \prod_{k=1}^{n} [k]_q. \tag{6}$$

They are clearly q-holonomic, and even q-hypergeometric.

The sequences $(u_n) = (q^n)$, $(v_n) = (q^{n^2})$ and $(w_n) = (q^{\binom{n}{2}})$ are also q-hypergeometric, since they satisfy the recurrence relations

$$u_{n+1} - qu_n = 0, \quad v_{n+1} - q^{2n+1}v_n = 0, \quad w_{n+1} - q^n w_n = 0.$$

However, the sequence (q^{n^3}) is not q-holonomic [37, Ex. 2.2(b)].

Another basic example is the q-Pochhammer symbol

$$(x; q)_n := \prod_{k=0}^{n-1} (1 - xq^k) \tag{7}$$

which is also q-hypergeometric, since $(x; q)_{n+1} - (1 - xq^n)(x; q)_n = 0$. In particular, the sequence $(q; q)_n := \prod_{k=1}^{n}(1 - q^k)$, also denoted $(q)_n$, is q-hypergeometric and satisfies $(q)_{n+1} - (1 - q^{n+1})(q)_n = 0$.

As mentioned in the introduction, q-holonomic sequences show up in various contexts. As an example, in (quantum) knot theory, the ("colored") Jones function of a (framed oriented) knot (in 3-space) is a powerful knot invariant, related to the Alexander polynomial [6]; it is a q-holonomic sequence of Laurent polynomials [36]. Its recurrence equations are themselves of interest, as they are closely related to the A-polynomial of a knot, via the *AJ conjecture*, verified in some cases using massive computer algebra calculations [35].

It is well known that the class of q-holonomic sequences is closed under several operations, such as addition, multiplication, Hadamard product and monomial substitution [37, 45]. All these closure properties are effective, *i.e.*, they can be executed algorithmically on the level of q-recurrences. Several computer algebra packages are available for the manipulation of q-holonomic sequences, *e.g.*, the Mathematica packages qGeneratingFunctions [45] and HolonomicFunctions [50], and the Maple packages qsum [12], qFPS [69], qseries and QDifferenceEquations.

A simple but useful fact is that the order-r scalar q-recurrence (5) can be translated into a first-order recurrence on $r \times 1$ vectors:

$$\begin{bmatrix} u_{n+r} \\ \vdots \\ u_{n+1} \end{bmatrix} = \begin{bmatrix} -\frac{c_{r-1}}{c_r} & \cdots & -\frac{c_1}{c_r} & -\frac{c_0}{c_r} \\ 1 & \cdots & 0 & 0 \\ \vdots & \ddots & \vdots & \vdots \\ 0 & \cdots & 1 & 0 \end{bmatrix} \times \begin{bmatrix} u_{n+r-1} \\ \vdots \\ u_n \end{bmatrix}. \quad (8)$$

In particular, the N-th term of the q-holonomic sequence (u_n) is simply expressible in terms of the *matrix q-factorial*

$$M(q^{N-1}) \cdots M(q) M(1), \quad (9)$$

where $M(q^n)$ denotes the companion matrix from equation (8).

3.2 Computation of the q-factorial

We now give the promised q-analogue of Strassen's result on the computation of $N!$ in $O(\mathbf{M}(\sqrt{N}) \log N)$ arithmetic operations. Note that Strassen's case $q = 1$ is also covered by [19, §6], where the cost $O(\mathbf{M}(\sqrt{N}))$ is reached under some invertibility assumptions.

Theorem 3.2. *Let $\mathbb{K}$ be a field, let $q \in \mathbb{K} \setminus \{1\}$ and $N \in \mathbb{N}$. The q-factorial $[N]_q!$ can be computed using $O(\mathbf{M}(\sqrt{N}))$ operations in $\mathbb{K}$. The same is true for the q-Pochhammer symbol $(\alpha; q)_N$ for any $\alpha \in \mathbb{K}$.*

Proof. If $\alpha = 0$, then $(\alpha; q)_N = 1$. If $q = 0$, then $[N]_q! = 1$ and $(\alpha; q)_N = 1 - \alpha$. We can assume that $q \in \mathbb{K} \setminus \{0, 1\}$ and $\alpha \in \mathbb{K} \setminus \{0\}$. We have $[N]_q! = r^N \cdot F(q^{-1})$ and $(\alpha; q)_N = \alpha^N \cdot F(\alpha^{-1})$, where $r := q/(1-q)$ and $F(x) := \prod_{i=0}^{N-1}(x - q^i)$. Algorithm 2 can be used to compute $F(q^{-1})$ and $F(\alpha^{-1})$ in $O(\mathbf{M}(\sqrt{N}))$ operations in $\mathbb{K}$. The cost of computing r^N and α^N is $O(\log N)$, and thus it is negligible. □

Corollary 3.3. *Under the assumptions of Theorem 3.2 and for any $n \in \mathbb{N}$, one can compute in $O(\mathbf{M}(\sqrt{N}))$ operations in $\mathbb{K}$:*

- *the q-binomial coefficient $\binom{N}{n}_q$;*
- *the coefficient of x^n in the polynomial $\prod_{k=1}^{N}(1 + q^{k-1}x)$;*
- *the sum $\binom{N-n}{0}_q \binom{n}{0}_q + q\binom{N-n}{1}_q \binom{n}{1}_q + \cdots + q^{n^2}\binom{N-n}{n}_q \binom{n}{n}_q$.*

Proof. The first assertion is a direct consequence of Theorem 3.2. The second assertion is a consequence of the first one, and of (1). The third assertion is a consequence of the first one, and of (2). □

3.3 N-th term of a q-holonomic sequence

We give the promised q-analogue of Chudnovskys' result on the computation of the N-th term of an arbitrary holonomic sequence in $O(\mathbf{M}(\sqrt{N}) \log N)$ arithmetic operations. Note that Chudnovskys' case $q = 1$ is also covered by [19, §6], where the improved cost $O(\mathbf{M}(\sqrt{N}))$ is reached under additional invertibility assumptions.

Theorem 3.4. *Let $\mathbb{K}$ be a field, $q \in \mathbb{K} \setminus \{1\}$ and $N \in \mathbb{N}$. Let $(u_n(q))_{n \geq 0}$ be a q-holonomic sequence satisfying recurrence (5), and assume that $c_r(q, q^k)$ is nonzero for $k = 0, \ldots, N-1$. Then, $u_N(q)$ can be computed in $O(\mathbf{M}(\sqrt{N}))$ operations in $\mathbb{K}$.*

Proof. Using equation (8), it is enough to show that the matrix q-factorial $M(q^{N-1}) \cdots M(q)M(1)$ can be computed in $O(\mathbf{M}(\sqrt{N}))$, where $M(q^n)$ denotes the companion matrix from equation (8). Algorithms 3 and 4 adapt *mutatis mutandis* to this effect. □

Corollary 3.5. *Let $\mathbb{K}$ be a field, $q \in \mathbb{K}$ not a root of unity, and $N \in \mathbb{N}$. Let $e_q(x)$ be the q-exponential series*

$$e_q(x) := \sum_{n \geq 0} \frac{x^n}{[n]_q!}$$

and let $E_q^{(N)}(x) := e_q(x) \bmod x^N$ be its truncation of degree $N-1$. If $\alpha \in \mathbb{K}$, one can compute $E_q^{(N)}(\alpha)$ in $O(\mathbf{M}(\sqrt{N}))$ operations in $\mathbb{K}$.

Proof. Denote the summand $\frac{\alpha^n}{[n]_q!}$ by $u_n(q)$. Then $(u_n(q))_n$ is q-hypergeometric, and satisfies the recurrence $[n+1]_q u_{n+1}(q) - \alpha u_n = 0$, therefore $v_N(q) := \sum_{i=0}^{N-1} u_i(q)$ satisfies the second-order recurrence $[n+1]_q(v_{n+2}(q) - v_{n+1}(q)) - \alpha(v_{n+1}(q) - v_n(q)) = 0$. Applying Theorem 3.4 to $v_N(q)$ concludes the proof. □

Remark that the same result holds if $e_q(x)$ is replaced by any power series satisfying a q-differential equation. For instance, one can evaluate fast all truncations of Heine's q-hypergeometric series

$${}_2\phi_1([a, b], [c]; q; x) := \sum_{n \geq 0} \frac{(a; q)_n (b; q)_n}{(c; q)_n} \cdot \frac{x^n}{(q)_n}.$$

Remark that Theorem 3.4 can be adapted to the computation of *several coefficients* of a q-holonomic sequence. We omit the proof, which is similar to that of Theorem 15 in [19].

Theorem 3.6. *Under the assumptions of Theorem 3.4, let $N_1 < N_2 < \cdots < N_s = N$ be positive integers, where $s < N^{\frac{1}{2}-\varepsilon}$ for some $0 < \varepsilon < \frac{1}{2}$. Then, the terms $u_{N_1}(q), \ldots, u_{N_s}(q)$ can be computed altogether in $O(\mathbf{M}(\sqrt{N}))$ operations in $\mathbb{K}$.*

3.4 The case q is an integer: bit complexity

Until now, we only considered the arithmetic complexity model. We briefly discuss here the case where q is an integer (or rational) number. The arithmetic complexity model needs to be replaced by the bit-complexity model, and the matrix q-factorials from §3.1 are computed by *binary splitting* rather than by baby-steps / giant-steps.

As an illustrative example, consider the computation of the term $u_N(q) = \sum_{n=0}^{N-1} q^{n^2}$, where q is assumed to be an integer of B bits. The integer $u_N(q)$ is bounded in absolute value by Nq^{N^2}, so its bitsize is of magnitude $N^2 B$. The "naive" algorithm consisting of computing the summands q^{n^2} one after the other, before summing

them, has bit complexity $\tilde{O}(N^3B)$. This is not (quasi-)optimal with respect to the output size. Can one do better? The answer is "yes". It is sufficient to use the q-holonomic character of $u_N(q)$, and to reduce its computation to that of a q-factorial matrix (9) as in §2.2. Now the point is that, instead of using baby-steps / giant-steps, it is a better idea to use binary splitting. The complexity of this approach becomes then quasi-optimal, that is $\tilde{O}(N^2B)$, which is quasi-linear in the bitsize of the output. The following general result can be proved along the same lines.

Theorem 3.7. *Under the assumptions of Theorem 3.4, with $\mathbb{K} = \mathbb{Q}$, the term $u_N(q)$ can be computed in $\tilde{O}(N^2B)$ bit operations, where B is the bitsize of q.*

As a corollary, (truncated) solutions of q-differential equations can be evaluated using the same (quasi-linear) bit complexity. This result should be viewed as the q-analogue of the classical fact that holonomic functions can be evaluated fast using binary splitting, a 1988 result by the Chudnovsky brothers [26, §6], anticipated a decade earlier (without proof) by Schroeppel and Salamin in Item 178 of [7]; see [8, §12] for a good survey on binary splitting.

4 APPLICATIONS

4.1 Combinatorial q-holonomic sequences

As already mentioned, many q-holonomic sequences arise in combinatorics, for example in connection with the enumeration of lattice polygons, where q-analogues of the Catalan numbers $\frac{1}{n+1}\binom{2n}{n}$ occur naturally [33, 39], or in the enumeration of special families of matrices with coefficients in the finite field $\mathbb{F}_q$ [47], where sequences related to the Gaussian coefficients $\binom{n}{k}_q$ also show up.

A huge subfield of combinatorics is the theory of partitions [5], where q-holonomic sequences occur as early as in the famous Roger-Ramanujan identities [5, Ch. 7], *e.g.*,

$$1 + \sum_{n\geq 1} \frac{q^{n^2}}{(1-q)\cdots(1-q^n)} = \prod_{n\geq 0} \frac{1}{(1-q^{5n+1})(1-q^{5n+4})}$$

which translates the fact that the number of partitions of n into parts that differ by at least 2 is equal to the number of partitions of n into parts congruent to 1 or 4 modulo 5. Andrews [5, Chapter 8] laid the foundations of a theory able to capture the q-holonomy of any generating function of a so-called *linked partition ideal*.

As a consequence, a virtually infinite number of special families of polynomials coming from partitions can be evaluated fast. For instance, the family of truncated polynomials

$$F_n(x) := \prod_{k=1}^{\infty} (1-x^k)^3 \bmod x^n,$$

can be evaluated fast due to our results and to the identity [56, §6]

$$F_N(q) = \sum_{\binom{n+1}{2}<N} (-1)^n (2n+1) q^{\binom{n+1}{2}}.$$

4.2 Evaluation of q-orthogonal polynomials

In the theory of special functions, *orthogonal polynomials* play a fundamental role. There exists an extension to the q-framework of the theory, see *e.g.*, Chapter 9 in Ernst's book [32]. Amongst the most basic examples, the *discrete q-Hermite polynomials* are defined by their q-exponential generating function

$$\sum_{n\geq 0} F_{n,q}(x) \frac{t^n}{[n]_q!} = \frac{e_q(xt)}{e_q(t)e_q(-t)},$$

and therefore they satisfy the second-order linear q-recurrence

$$F_{n+1,q}(x) = xF_{n,q}(x) - (1-q^n)q^{n-1}F_{n-1,q}(x), \quad n \geq 1,$$

with initial conditions $F_{0,q}(x) = 1, F_{1,q}(x) = x$. From there, it follows that for any $\alpha \in \mathbb{K}$, the sequence $(F_{n,q}(\alpha))_{n\geq 0}$ is q-holonomic, thus the evaluation of the n-th polynomial at $x = \alpha$ can be computed fast. The same is true for the *continuous q-Hermite polynomials*, for which $2\alpha H_{n,q}(\alpha) = H_{n+1,q}(\alpha) + (1-q^n)H_{n-1,q}(\alpha)$ for $n \geq 1$, and $H_{0,q}(\alpha) = 1, H_{1,q}(\alpha) = 2\alpha$. More generally, our results in §3 imply that any family of q-orthogonal polynomials can be evaluated fast.

4.3 Polynomial and rational solutions of q-differential equations

The computation of polynomial and rational solutions of linear differential equations lies at the heart of several important algorithms, for computing hypergeometric and Liouvillian solutions, for factoring and for computing differential Galois groups [73]. Creative telescoping algorithms (of second generation) for multiple integration with parameters [28, 50] also rely on computing rational solutions, or deciding their existence. The situation is completely similar for q-differential equations: improving algorithms for polynomial and rational solutions of such equations is important in finding q-hypergeometric solutions [3], in computing q-differential Galois groups [43], and in performing q-creative telescoping [28, 49, 50].

In both differential and q-differential cases, algorithms for computing polynomial solutions proceed in two distinct phases: (i) compute a degree bound N, potentially exponentially large in the equation size; (ii) reduce the problem of computing polynomial solutions of degree at most N to linear algebra. Abramov, Bronstein and Petkovšek showed in [1] that, in step (ii), linear algebra in size N can be replaced by solving a much smaller system, of polynomial size. However, setting up this smaller system still requires linear time in N, essentially by unrolling a (q-)linear recurrence up to terms of indices close to N. For differential (and difference) equations, this step has been improved in [17, 18], by using Chudnovskys' algorithms for computing fast the N-th term of a holonomic sequence. This allows for instance to decide (non-)existence of polynomial solutions in sublinear time $\tilde{O}(\sqrt{N})$. Moreover, when polynomial solutions exist, one can represent / manipulate them in *compact form* using the recurrence and initial terms as a compact data structure.

The same improvements can be transferred to q-differential equations, in order to improve the existing algorithms [1, 2, 46]. In this case, setting up the smaller system in phase (ii) amounts to computing the N-th term of a q-holonomic sequence, and this can be done fast using our results in §3*.

*A technical subtlety is that, as pointed out in [1, §4.3], it is not obvious in the q-differential case how to guarantee the non-singularity of the q-recurrence on the coefficients of the solution. This induces potential technical complications similar to the ones for polynomial solutions of differential equations in small characteristic, which can nevertheless be overcome by adapting the approach described in [22, §3.2].

4.4 q-hypergeometric creative telescoping

In the case of differential and difference hypergeometric creative telescoping, it was demonstrated in [17] that the compact representation for polynomial solutions can be used as an efficient data structure, and can be applied to speed up the computation of Gosper forms and Zeilberger's classical summation algorithm [61, §6]. The key to these improvements lies in the fast computation of the N-th term of a holonomic sequence, together with the close relation between Gosper's algorithm and the algorithms for rational solutions.

Similarly, in the q-differential case, Koornwinder's q-Gosper algorithm [49, §5] is closely connected to Abramov's algorithm for computing rational solutions [2, §2], and this makes it possible to transfer the improvements for rational solutions to the q-Gosper algorithm. This leads in turn to improvements upon Koornwinder's algorithm for q-hypergeometric summation [49], along the same lines as in the differential and difference cases [17].

5 EXPERIMENTS

A preliminary implementation in Magma of Algorithms 1 and 2 in §2.1 delivers some encouraging timings. Of course, since these algorithms are designed to be fast in the *arithmetic model*, it is natural to make experiments over a finite field $\mathbb{K}$, or over truncations of real/complex numbers, as was done in [54] for the problem in §2.2. Recall that both Algorithms 1 and 2 compute $\prod_{i=0}^{N-1}(\alpha - q^i) \in \mathbb{K}$, given α, q in a field $\mathbb{K}$, and $N \in \mathbb{N}$. In our experiments, $\mathbb{K}$ is the finite field $\mathbb{F}_p$ with $p = 2^{30} + 3$ elements. Timings are given in Table 1. We compare the straightforward iterative algorithm (column Naive), to the fast baby-step / giant-step algorithms, one based on subproduct trees and resultants (column Algorithm 1), the other based on multipoint evaluation on geometric sequences (column Algorithm 2).

Some conclusions can be drawn by analyzing these timings:

- The theoretical complexities are perfectly reflected in practice: timings are multiplied (roughly) by 4 in column Naive, and (roughly) by 2 in columns Algorithm 1 and Algorithm 2.
- The asymptotic regime is reached from the very beginning.
- Algorithm 2 is always faster than Algorithm 1, which is itself much faster than the Naive algorithm, as expected.
- A closer look into the timings shows that for Algorithm 1, $\approx 80\%$ of the time is spent in step (3) (resultant computation), the other steps taking $\approx 10\%$ each; for Algorithm 2, step (1) takes $\approx 25\%$, step (2) takes $\approx 75\%$, and step (3) is negligible.

degree N	Naive algorithm	Algorithm 1	Algorithm 2
2^{16}	0.04	0.03	0.00
2^{18}	0.18	0.03	0.01
2^{20}	0.72	0.06	0.01
2^{22}	2.97	0.14	0.02
2^{24}	11.79	0.32	0.04
2^{26}	47.16	0.73	0.08
2^{28}	188.56	1.68	0.15
2^{30}	755.65	3.84	0.31
2^{32}	3028.25	8.65	0.64
2^{34}		19.65	1.41
2^{36}		44.42	2.96
2^{38}		101.27	6.36
2^{40}		228.58	14.99
2^{42}		515.03	29.76
2^{44}		1168.51	61.69
2^{46}		2550.28	137.30
2^{48}			297.60
2^{50}			731.63
2^{52}			1395.33
2^{54}			3355.39

Table 1 Comparative timings (in seconds) for the computation of $\prod_{i=0}^{N-1}(\alpha - q^i) \in \mathbb{F}_p$, with $p = 2^{30} + 3$ and (α, q) randomly chosen in $\mathbb{F}_p \times \mathbb{F}_p$. All algorithms were executed on the same machine, running Magma v. 2.24. For each target degree N, each execution was limited to one hour. Naive algorithm could reach degree $N = 2^{32}$, Algorithm 1 degree $N = 2^{46}$, and Algorithm 2 degree $N = 2^{54} = 8\,014\,398\,509\,481\,984$. By extrapolation, the Naive algorithm would have needed $\approx 4^{11} \times 3028.25$ sec. ≈ 400 years on the same instance, and Algorithm 2 approximately 18 hours.

6 CONCLUSION AND FUTURE WORK

We have shown that selected terms of q-holonomic sequences can be computed fast, both in theory and in practice, the key being the extension of classical algorithms in the holonomic ("$q = 1$") case. We have demonstrated through several examples that this basic algorithmic improvement has many other algorithmic implications, notably on the faster evaluation of many families of polynomials and on the acceleration of algorithms for q-differential equations.

Here are some questions that we plan to investigate in the future.

1. (Computing curvatures of q-differential equations) In the differential case, p-curvatures can be computed fast [14–16, 22]. What about the q-differential analogue? One strong motivation comes from the fact that the q-analogue [10] of Grothendieck's conjecture (relating equations over $\mathbb{Q}$ with their reductions modulo primes p) is proved [29]. This could be used to improve the computation of rational solutions.
2. (Counting points on q-curves) Counting efficiently points on (hyper-)elliptic curves leads to questions like: for $a, b \in \mathbb{Z}$, compute the coeff. of $x^{\frac{p-1}{2}}$ in $G_p(x) := (x^2 + ax + b)^{\frac{p-1}{2}}$ modulo p, for one [19] or several [41] primes p. A natural extension is to ask the same with $G_p(x)$ replaced by $\prod_{k=1}^{\frac{p-1}{2}}(q^{2k}x^2 + aq^k x + b)$. This might have applications related to Question 1, or to counting points on q-deformations [67].
3. (Computing q-deformed real numbers) Recently, Morier-Genoud and Ovsienko [53] introduced q-analogues of real numbers. How fast can one compute (truncations / evaluations of) quantized versions of numbers like e or π?
4. (Evaluating more polynomials) Is it possible to evaluate fast polynomials of the form $\sum_{n=0}^{N} x^{n^s}$, for $s \geq 3$, and many others that escape the q-holonomic class? *E.g.*, [9] presents a beautiful generalization of Algorithm 1 to the fast evaluation of isogenies between elliptic curves, by using *elliptic resultants*, with applications in isogeny-based cryptography.

Acknowledgements. I thank Luca De Feo for his initial question, who motivated this work, and for the very interesting subsequent discussions. My friendly thanks go to Lucia Di Vizio, Kilian Raschel and Sergey Yurkevich for their careful reading of the manuscript. I am indebted to the three referees for many helpful remarks. This work was supported in part by DeRerumNatura ANR-19-CE40-0018.

REFERENCES

[1] S. A. Abramov, M. Bronstein, and M. Petkovšek. On polynomial solutions of linear operator equations. In *ISSAC'95*, pages 290–296. ACM, 1995.
[2] S. A. Abramov. Rational solutions of linear difference and q-difference equations with polynomial coefficients. *Programmirovanie*, (6):3–11, 1995.
[3] S. A. Abramov, P. Paule, and M. Petkovšek. q-hypergeometric solutions of q-difference equations. *Discrete Math.*, 180(1-3):3–22, 1998.
[4] M. Aldaz, G. Matera, J. L. Montaña, and L. M. Pardo. A new method to obtain lower bounds for polynomial evaluation. *TCS*, 259(1-2):577–596, 2001.
[5] G. E. Andrews. *The theory of partitions.* Addison-Wesley, Reading, 1976.
[6] D. Bar-Natan and S. Garoufalidis. On the Melvin-Morton-Rozansky conjecture. *Invent. Math.*, 125(1):103–133, 1996.
[7] M. Beeler, R. Gosper, and R. Schroeppel. *HAKMEM.* Artificial Intelligence Memo No. 239. MIT, 1972. http://www.inwap.com/pdp10/hbaker/hakmem/algorithms.
[8] D. J. Bernstein. Fast multiplication and its applications. In *Algorithmic number theory: lattices, number fields, curves and cryptography*, *MSRIP* 44:325–384. 2008.
[9] D. J. Bernstein, L. De Feo, A. Leroux, and B. Smith. Faster computation of isogenies of large prime degree. Preprint, 2020. https://eprint.iacr.org/2020/341.
[10] J.-P. Bézivin. Les suites q-récurrentes linéaires. *Comp. Math.*, 80:285–307, 1991.
[11] L. I. Bluestein. A linear filtering approach to the computation of the discrete Fourier transform. *IEEE Trans. Electroacoustics*, AU-18:451–455, 1970.
[12] H. Böing and W. Koepf. Algorithms for q-hypergeometric summation in computer algebra. *J. Symbolic Comput.*, 28(6):777–799, 1999.
[13] A. Borodin and S. Cook. On the number of additions to compute specific polynomials. *SIAM J. Comput.*, 5(1):146–157, 1976.
[14] A. Bostan, X. Caruso, and É. Schost. A fast algorithm for computing the characteristic polynomial of the p-curvature. In *ISSAC'14*, pages 59–66. ACM, 2014.
[15] A. Bostan, X. Caruso, and É. Schost. A fast algorithm for computing the p-curvature. In *ISSAC'15*, pages 69–76. ACM, 2015.
[16] A. Bostan, X. Caruso, and É. Schost. Computation of the similarity class of the p-curvature. In *ISSAC'16*, pages 111–118. ACM, 2016.
[17] A. Bostan, F. Chyzak, T. Cluzeau, and B. Salvy. Low complexity algorithms for linear recurrences. In *ISSAC'06*, pages 31–38. ACM, 2006.
[18] A. Bostan, T. Cluzeau, and B. Salvy. Fast algorithms for polynomial solutions of linear differential equations. In *ISSAC'05*, pages 45–52. ACM, 2005.
[19] A. Bostan, P. Gaudry, and É. Schost. Linear recurrences with polynomial coefficients and application to integer factorization and Cartier-Manin operator. *SIAM J. Comput.*, 36(6):1777–1806, 2007.
[20] A. Bostan, G. Lecerf, and É. Schost. Tellegen's principle into practice. In *ISSAC'03*, pages 37–44. ACM, 2003.
[21] A. Bostan and É. Schost. Polynomial evaluation and interpolation on special sets of points. *J. Complexity*, 21(4):420–446, 2005.
[22] A. Bostan and É. Schost. Fast algorithms for differential equations in positive characteristic. In *ISSAC'09*, pages 47–54. ACM, 2009.
[23] P. Bürgisser, M. Clausen, and M. A. Shokrollahi. *Algebraic complexity theory*, volume 315 of *Grundlehren der Mathematischen Wissenschaften.* Springer, 1997.
[24] D. G. Cantor and E. Kaltofen. On fast multiplication of polynomials over arbitrary algebras. *Acta Inform.*, 28(7):693–701, 1991.
[25] P. Cartier. Démonstration "automatique" d'identités et fonctions hypergéométriques (d'après D. Zeilberger). *Astérisque*, (206): 41–91, 1992. S. Bourbaki.
[26] D. V. Chudnovsky and G. V. Chudnovsky. Approximations and complex multiplication according to Ramanujan. In *Ramanujan revisited (Urbana-Champaign, Ill., 1987)*, pages 375–472. Academic Press, Boston, MA, 1988.
[27] F. Chyzak. Gröbner bases, symbolic summation and symbolic integration. In *Gröbner bases and applications*, volume LMS LN 251:32–60. CUP, 1998.
[28] F. Chyzak. An extension of Zeilberger's fast algorithm to general holonomic functions. *Discrete Math.*, 217(1-3):115–134, 2000.
[29] L. Di Vizio. Arithmetic theory of q-difference equations: the q-analogue of Grothendieck-Katz's conjecture on p-curvatures. *Invent. Math.*, 150:517–578, 2002.
[30] L. Di Vizio, J.-P. Ramis, J. Sauloy, and C. Zhang. Équations aux q-différences. *Gaz. Math.*, (96):20–49, 2003.
[31] T. Ekedahl and G. van der Geer. Cycle classes on the moduli of K3 surfaces in positive characteristic. *Selecta Math. (N.S.)*, 21(1):245–291, 2015.
[32] T. Ernst. *A comprehensive treatment of q-calculus.* Birkhäuser/Springer, 2012.
[33] J. Fürlinger and J. Hofbauer. q-Catalan numbers. *JCTA*, 40(2):248–264, 1985.
[34] F. Le Gall. Powers of tensors and fast matrix multiplication. In *ISSAC'14*, pages 296–303. ACM, 2014.
[35] S. Garoufalidis and C. Koutschan. Irreducibility of q-difference operators and the knot 7_4. *Algebr. Geom. Topol.*, 13(6):3261–3286, 2013.
[36] S. Garoufalidis and T. T. Q. Lê. The colored Jones function is q-holonomic. *Geom. Topol.*, 9:1253–1293, 2005.
[37] S. Garoufalidis and T. T. Q. Lê. A survey of q-holonomic functions. *Enseign. Math.*, 62(3-4):501–525, 2016.
[38] J. von zur Gathen and J. Gerhard. *Modern computer algebra.* CUP, 3rd ed., 2013.
[39] I. Gessel. A noncommutative generalization and q-analog of the Lagrange inversion formula. *Trans. Amer. Math. Soc.*, 257(2):455–482, 1980.
[40] G. Hanrot, M. Quercia, and P. Zimmermann. The middle product algorithm. I. *Appl. Algebra Engrg. Comm. Comput.*, 14(6):415–438, 2004.
[41] D. Harvey. Counting points on hyperelliptic curves in average polynomial time. *Ann. of Math. (2)*, 179(2):783–803, 2014.
[42] J. Heintz and M. Sieveking. Lower bounds for polynomials with algebraic coefficients. *TCS*, 11(3):321–330, 1980.
[43] P. A. Hendriks. An algorithm for computing a standard form for second-order linear q-difference equations. *J. Pure Appl. Algebra*, 117/118:331–352, 1997.
[44] J. Hua. Counting representations of quivers over finite fields. *J. Algebra*, 226(2):1011–1033, 2000.
[45] M. Kauers and C. Koutschan. A Mathematica package for q-holonomic sequences and power series. *Ramanujan J.*, 19(2):137–150, 2009.
[46] D. E. Khmel′ nov. Improved algorithms for solving difference and q-difference equations. *Programmirovanie*, (2):70–78, 2000.
[47] A. A. Kirillov and A. Melnikov. On a remarkable sequence of polynomials. In *Algèbre non commutative, groupes quantiques et invariants (Reims, 1995)*, volume 2 of *Sémin. Congr.*, pages 35–42. Soc. Math. France, Paris, 1997.
[48] R. Koekoek, P. A. Lesky, and R. F. Swarttouw. *Hypergeometric orthogonal polynomials and their q-analogues.* Monographs in Mathematics. Springer, 2010.
[49] T. H. Koornwinder. On Zeilberger's algorithm and its q-analogue. *J. Comput. Appl. Math.*, 48(1-2):91–111, 1993.
[50] C. Koutschan. A fast approach to creative telescoping. *Math. Comput. Sci.*, 4(2-3):259–266, 2010.
[51] H. Labrande. Computing Jacobi's theta in quasi-linear time. *Math. Comp.*, 87(311):1479–1508, 2018.
[52] R. J. Lipton. Polynomials with $0-1$ coefficients that are hard to evaluate. *SIAM J. Comput.*, 7(1):61–69, 1978.
[53] S. Morier-Genoud and V. Ovsienko. On q-deformed real numbers. *Exp. Math.*, pages 1–9, 2019. To appear.
[54] D. Nogneng and É. Schost. On the evaluation of some sparse polynomials. *Math. Comp.*, 87(310):893–904, 2018.
[55] A. Ostrowski. On two problems in abstract algebra connected with Horner's rule. In *Studies in mathematics and mechanics presented to Richard von Mises*, pages 40–48. Academic Press Inc., 1954.
[56] I. Pak. Partition bijections, a survey. *Ramanujan J.*, 12(1):5–75, 2006.
[57] V. Y. Pan. Methods of computing values of polynomials. *Russian Mathematical Surveys*, 21(1):105–136, 1966.
[58] M. S. Paterson and L. J. Stockmeyer. On the number of nonscalar multiplications necessary to evaluate polynomials. *SIAM J. Comput.*, 2:60–66, 1973.
[59] P. Paule and S. Radu. Rogers-Ramanujan functions, modular functions, and computer algebra. In *Advances in computer algebra*, PROMS 226, 229–280, 2018.
[60] P. Paule and A. Riese. A Mathematica q-analogue of Zeilberger's algorithm based on an algebraically motivated approach to q-hypergeometric telescoping. In *Special functions, q-series and related topics*, FIC 14:179–210. AMS, 1997.
[61] M. Petkovšek, H. S. Wilf, and D. Zeilberger. $A = B$. A K Peters, 1996.
[62] J. M. Pollard. Theorems on factorization and primality testing. *Proc. Cambridge Philos. Soc.*, 76:521–528, 1974.
[63] L. R. Rabiner, R. W. Schafer, and C. M. Rader. The chirp z-transform algorithm and its application. *Bell System Tech. J.*, 48:1249–1292, 1969.
[64] A. Riese. qMultiSum—a package for proving q-hypergeometric multiple summation identities. *J. Symbolic Comput.*, 35(3):349–376, 2003.
[65] C. Sabbah. Systèmes holonomes d'équations aux q-différences. In *D-modules and microlocal geometry (Lisbon, 1990)*, pages 125–147. de Gruyter, 1993.
[66] C.-P. Schnorr. improved lower bounds on the number of multiplications / divisions which are necessary to evaluate polynomials. *TCS*, 7(3):251–261, 1978.
[67] P. Scholze. Canonical q-deformations in arithmetic geometry. *Ann. Fac. Sci. Toulouse Math. (6)*, 26(5):1163–1192, 2017.
[68] A. Schönhage. Schnelle Multiplikation von Polynomen über Körpern der Charakteristik 2. *Acta Informatica*, 7:395–398, 1977.
[69] T. Sprenger and W. Koepf. Algorithmic determination of q-power series for q-holonomic functions. *J. Symbolic Comput.*, 47(5):519–535, 2012.
[70] V. Strassen. Polynomials with rational coefficients which are hard to compute. *SIAM J. Comput.*, 3:128–149, 1974.
[71] V. Strassen. Einige Resultate über Berechnungskomplexität. *Jber. Deutsch. Math.-Verein.*, 78(1):1–8, 1976/77.
[72] T. Tao, E. Croot, III, and H. Helfgott. Deterministic methods to find primes. *Math. Comp.*, 81(278):1233–1246, 2012.
[73] M. van der Put and M. F. Singer. *Galois theory of linear differential equations*, volume 328 of *Grundlehren der Mathematischen Wissenschaften.* Springer, 2003.
[74] H. S. Wilf and D. Zeilberger. An algorithmic proof theory for hypergeometric (ordinary & q) multisum/integral identities. *Invent. Math.*, 108(3):575–633, 1992.
[75] K.-W. Yang. On the product $\prod_{n\geq 1}(1+q^n x+q^{2n}x^2)$. *J. Austral. Math. Soc. Ser. A*, 48(1):148–151, 1990.
[76] D. Zagier. Elliptic modular forms and their applications. In *The 1-2-3 of modular forms*, Universitext, pages 1–103. Springer, 2008.
[77] D. Zeilberger. A holonomic systems approach to special functions identities. *J. Comput. Appl. Math.*, 32(3):321–368, 1990.

Separating Variables in Bivariate Polynomial Ideals

Manfred Buchacher*
Institute for Algebra
Johannes Kepler University
Linz, Austria
manfred.buchacher@jku.at

Manuel Kauers†
Institute for Algebra
Johannes Kepler University
Linz, Austria
manuel.kauers@jku.at

Gleb Pogudin‡
Department of Computer Science
Higher School of Economics
Moscow, Russia
LIX, CNRS
École Polytechnique,
Institut Polytechnique de Paris
France
pogudin.gleb@gmail.com

ABSTRACT

We present an algorithm which for any given ideal $I \subseteq \mathbb{K}[x, y]$ finds all elements of I that have the form $f(x) - g(y)$, i.e., all elements in which no monomial is a multiple of xy.

ACM Reference Format:
Manfred Buchacher, Manuel Kauers, and Gleb Pogudin. 2020. Separating Variables in Bivariate Polynomial Ideals. In *International Symposium on Symbolic and Algebraic Computation (ISSAC '20), July 20–23, 2020, Kalamata, Greece.* ACM, New York, NY, USA, 8 pages. https://doi.org/10.1145/3373207.3404028

1 INTRODUCTION

One of the fundamental problems in computer algebra and applied algebraic geometry is the problem of elimination. Here, we are given a polynomial ideal $I \subseteq \mathbb{K}[x_1, \ldots, x_n, y_1, \ldots, y_m]$ and the task is to compute generators of the ideal $I \cap \mathbb{K}[x_1, \ldots, x_n]$. The resulting ideal of $\mathbb{K}[x_1, \ldots, x_n]$ consists of all elements of I that do not contain any terms that are a multiple of any of the variables y_i. It is well-known that this problem can be solved by computing a Gröbner basis with respect to an elimination order that assigns higher weight to terms involving $y_1, \ldots, y_m$ than to terms not involving these variables.

It is less clear how to use Gröbner bases (or any other standard elimination techniques) for finding ideal elements that do not contain any terms which are a multiple of certain prescribed terms rather than certain prescribed variables. The problem considered in this paper is an elimination problem of this kind. Here, given an ideal $I \subseteq \mathbb{K}[x_1, \ldots, x_n, y_1, \ldots, y_m]$, we are interested in all elements of I that do not involve any terms which are multiples of any of the terms $x_i y_j$ ($i = 1, \ldots, n$, $j = 1, \ldots, m$). Note that, these are precisely the elements of I which can be written as the sum of a polynomial in $x_1, \ldots, x_n$ only and a polynomial in $y_1, \ldots, y_m$ only, so the problem under consideration is as follows.

Problem 1.1 (Separation).
Input *An ideal* $I \subseteq \mathbb{K}[x_1, \ldots, x_n, y_1, \ldots, y_m]$*;*
Output *Description of all* $f - g \in I$ *such that*

$$f \in \mathbb{K}[x_1, \ldots, x_n] \text{ and } g \in \mathbb{K}[y_1, \ldots, y_m].$$

At first glance, it may seem that there should be a simple way to solve this problem with Gröbner bases, similarly as for the classical elimination problem. However, we were not able to come up with such an algorithm. The obstruction seems to be that there is no term order that ranks the term xy higher than both x^2 and y^2.

We ran into the need for such an algorithm when we tried to automatize an interesting non-standard elimination step which appears in Bousquet-Mélou's "elementary" solution of Gessel's walks [9]. Dealing with certain power series, say $u \in \mathbb{K}[x][[t]]$ and $v \in \mathbb{K}[x^{-1}][[t]]$, she finds polynomials f, g such that $f(u) - g(v) = 0$, and then concludes that $f(u)$ and $g(v)$ must in fact belong to $\mathbb{K}[[t]]$. Deriving a pair (f, g) automatically from known relations among u, v amounts to the problem under consideration.

The problem also arises when one wants to compute the intersection of two $\mathbb{K}$-algebras. For example, suppose that for given $u, v \in \mathbb{K}[t_1, \ldots, t_n]$ one wants to compute $\mathbb{K}[u] \cap \mathbb{K}[v]$. This can be done by finding all pairs (f, g) such that $f(u) = g(v)$, i.e., all pairs (f, g) with $f(x) - g(y) \in \langle x - u, y - v \rangle \cap \mathbb{K}[x, y]$. See [3, 13] for a discussion of this and similar problems.

Definition 1.2. *Let* $p \in \mathbb{K}[x_1, \ldots, x_n, y_1, \ldots, y_m]$.
(1) p *is called* separated *if there exist* $f \in \mathbb{K}[x_1, \ldots, x_n]$ *and* $g \in \mathbb{K}[y_1, \ldots, y_m]$ *such that* $p = f - g$.
(2) p *is called* separable *if there is a* $q \in \mathbb{K}[x_1, \ldots, x_n, y_1, \ldots, y_m]$ *such that* qp *is separated.*

Proposition 1.3. *Let* I *be an ideal in* $\mathbb{K}[x_1, \ldots, x_n, y_1, \ldots, y_m]$. *Then*

$$A(I) := \{ (f, g) \in \mathbb{K}[x_1, \ldots, x_n] \times \mathbb{K}[y_1, \ldots, y_m] : f - g \in I \}$$

is a unital $\mathbb{K}$*-algebra with respect to component-wise addition and multiplication and component-wise multiplication by elements of* $\mathbb{K}$. *We refer to* $A(I)$ *as the* algebra of separated polynomials *of* I.

*Supported by the Austrian FWF grant F5004. Part of this work was done during the visit of MB to HSE University. MB would like to thank the Faculty of Computer Science of HSE for its hospitality.
†Supported by the Austrian FWF grants F5004 and P31571-N32
‡Supported by NSF grants CCF-1564132, CCF-1563942, DMS-1853482, DMS-1853650, and DMS-1760448, by PSC-CUNY grants #69827-0047 and #60098-0048.

ISSAC '20, July 20–23, 2020, Kalamata, Greece

ACM ISBN 978-1-4503-7100-1/20/07…$15.00
https://doi.org/10.1145/3373207.3404028

Proof. We just note that $A(I)$ is clearly a $\mathbb{K}$-vector space, and that it is closed under component-wise multiplication, as for any $(f,g),(f',g') \in A(I)$ we have $f-g \in I$ and $f'-g' \in I$, so $(f-g)f' + g(f'-g') = ff' - gg' \in I$. It is unital, because we always have $(1,1) \in A(I)$. □

Given ideal generators of I, we want to determine $\mathbb{K}$-algebra generators of $A(I)$. This is in general too much to be asked for, because, as shown in Example 5.1, $A(I)$ may not be finitely generated. On the positive side, it is known that $A(I)$ is finitely generated if I is a principal ideal in the ring of bivariate polynomials (see [15]).

The main result of the paper is Algorithm 4.3 for computing generators of the algebra $A(I)$ for a given bivariate ideal $I \subseteq \mathbb{K}[x,y]$. In particular, it implies that such an algebra is always finitely generated and yields an algorithm to compute a minimal separated multiple of a bivariate polynomial [15, Definition 4.1]. An implementation of the algorithm in Mathematica can be found on the website of the second author.

The general structure of the algorithm is the following. Every bivariate ideal is the intersection of a zero-dimensional ideal and a principal ideal. We solve the separation problem for the zero-dimensional case (Section 2) and for the principal case (Section 3) separately. Then we show how to compute the intersection of the resulting algebras in Section 4. We conclude with discussing the case of more than two variables in Section 5.

In the context of separated polynomials, many deep results have been obtained for some kind of "inverse problem" to the problem considered here, i.e., the study of the shape of factors of polynomials of the form $f(x)-g(y)$, see [6, 7, 10–12, 14, 15] and references therein. We use techniques developed in [10] in our proofs (see Section 3).

We assume throughout that the ground field $\mathbb{K}$ has characteristic zero and that for a given element of an algebraic extension of $\mathbb{K}$ we can decide whether it is a root of unity. This is true, for example, for every number field (see Section 3.3).

It is an open question whether the assumption on the characteristic of $\mathbb{K}$ can be eliminated. In positive characteristic, additional phenomena have to be taken into account. For example, separable polynomials need not be squarefree, as the example $(x+y)^2 \in \mathbb{Z}_3[x,y]$ shows, which is separable because $(x+y)(x+y)^2 = (x+y)^3 = x^3+y^3$.

2 ZERO-DIMENSIONAL IDEALS

When $I \subseteq \mathbb{K}[x,y]$ has dimension zero, it is easy to separate variables. In this case, there are nonzero polynomials p, q with $I \cap \mathbb{K}[x] = \langle p\rangle$ and $I \cap \mathbb{K}[y] = \langle q\rangle$. Clearly, these univariate polynomials p and q are separated. Also all $\mathbb{K}[x]$-multiples of p and all $\mathbb{K}[y]$-multiples of q are separated elements of I.

An arbitrary pair $(f,g) \in \mathbb{K}[x]\times\mathbb{K}[y]$ belongs to $A(I)$ if and only if $(f+up, g+vq)$ belongs to $A(I)$ for all $u \in \mathbb{K}[x]$ and $v \in \mathbb{K}[y]$. In particular, we have $(f,g) \in A(I) \iff (\mathrm{rem}_x(f,p), \mathrm{rem}_y(g,q)) \in A(I)$. It is therefore sufficient to find all pairs $(f,g) \in A(I)$ with $\deg_x f < \deg_x p$ and $\deg_y g < \deg_y q$. These pairs can be found with linear algebra.

Algorithm 2.1. *Input: $I \subseteq \mathbb{K}[x,y]$ of dimension zero.*
Output: generators of the $\mathbb{K}$-algebra $A(I) \subseteq \mathbb{K}[x]\times\mathbb{K}[y]$

1 *if $I = \langle 1\rangle$, return $\{(1,0),(x,0),(0,1),(0,y)\}$.*
2 *compute $p \in \mathbb{K}[x]$ and $q \in \mathbb{K}[y]$ such that*
$$I \cap \mathbb{K}[x] = \langle p\rangle \quad \text{and} \quad I \cap \mathbb{K}[y] = \langle q\rangle.$$
3 *make an ansatz $h = \sum_{i=0}^{\deg_x p-1} a_i x^i - \sum_{j=0}^{\deg_y q-1} b_j y^j$ with undetermined coefficients a_i, b_j.*
4 *compute the normal form of h with respect to a Gröbner basis of I and equate its coefficients to zero.*
5 *solve the resulting linear system over $\mathbb{K}$ for the unknowns a_i, b_j and let $(f_1,g_1),\dots,(f_d,g_d)$ be the pairs of polynomials corresponding to a basis of the solution space.*
6 *return $(f_1,g_1),\dots,(f_d,g_d),(p,0),\dots,(x^{\deg_x p-1}p,0)$, $(0,q),\dots,(0,y^{\deg_y q-1}q)$.*

Proposition 2.2. *Algorithm 2.1 is correct.*

Proof. It is clear by construction that all returned elements belong to $A(I)$. It remains to show that they generate $A(I)$ as $\mathbb{K}$-algebra. This is clear if $I = \langle 1\rangle$, because then $A(I) = \mathbb{K}[x]\times\mathbb{K}[y]$. Now suppose that $I \neq \langle 1\rangle$ and let $(f,g) \in A(I)$. Because of $I \neq \langle 1\rangle$, we have $\deg_x p, \deg_y q > 0$. Then $\langle p\rangle \subseteq \mathbb{K}[x]$ is generated as a $\mathbb{K}$-algebra by $p, xp, \dots, x^{\deg_x p-1}p$. To see this, we just note that, by performing repeatedly division by p on a polynomial and the resulting quotients, any $u \in \langle p\rangle$ can be written

$$u = \sum_{i=1}^{k} r_i p^i$$

where r_i are polynomials with $\deg r_i < \deg p$. Hence, $\langle p\rangle$ is a subset of the algebra generated by $p, xp, \dots, x^{\deg_x p-1}p$, and clearly, the reverse inclusion holds as well. For the same reason, $\langle q\rangle$ is generated as $\mathbb{K}$-algebra by $q, xq, \dots, x^{\deg_x q-1}q$.

Hence (f,g) can be expressed in terms of the given generators if and only if $(\mathrm{rem}_x(f,p), \mathrm{rem}_y(g,q))$ can be expressed in terms of the given generators. Because of $\deg_x(\mathrm{rem}_x(f,p)) < \deg_x(p)$ and $\deg_y(\mathrm{rem}_y(g,q)) < \deg_y(q)$, the pair $(\mathrm{rem}_x(f,p), \mathrm{rem}_y(g,q))$ is a $\mathbb{K}$-linear combination of $(f_1,g_1),\dots,(f_d,g_d)$, as required. □

Example 2.3. *Consider the 0-dimensional ideal $I = \langle x^2y^2-1, y^5+y^3+xy^2+x\rangle$. We have*

$$I \cap \mathbb{K}[x] = \langle x^{10}+x^8-x^2-1\rangle \quad \text{and} \quad I \cap \mathbb{K}[y] = \langle y^{10}+y^8-y^2-1\rangle.$$

Every separated polynomial of I therefore has the form

$$f(x) + u(x)(x^{10}+x^8-x^2-1) - g(y) - v(y)(y^{10}+y^8-y^2-1)$$

for certain $f(x), g(y)$ of degree less than 10 and some $u(x), v(y)$. To find the pairs (f,g), compute the normal form of $h = \sum_{i=0}^{9} a_i x^i - \sum_{i=0}^{9} b_j y^j$ with respect to a Gröbner basis of I. Taking a degrevlex Gröbner basis, this gives

$$(a_0+a_8-b_0) + (a_6-b_2)y^2 + (a_7+b_5)xy^2 + \cdots.$$

Equate the coefficients with respect to x, y to zero and solve the resulting linear system for the unknowns $a_0,\dots,a_9,b_0,\dots,b_9$. The following pairs of polynomials (f,g) correspond to a basis of the solution space:

$$(1,1),\ (x-x^9, y^9-y),\ (x^2, y^8+y^6-1),\ (x^9+x^3, -y^9-y^3)$$
$$(x^4, -y^8+y^4+1),\ (x^5-x^9, y^3-y^7),\ (x^6, y^8+y^2-1)$$
$$(x^9+x^7, -y^5-y^3),\ (x^8, 2-y^8).$$

These pairs together with the pairs $(x^i(x^{10}+x^8-x^2-1),0)$ *and* $(0,y^i(y^{10}+y^8-y^2-1))$ *for* $i=0,\dots,9$ *form a set of generators of* $A(I)$.

For an ideal $I\subseteq\mathbb{K}[x,y]$ to be zero-dimensional means that its codimension as $\mathbb{K}$-subspace of $\mathbb{K}[x,y]$ is finite. Note that, in this case, also $A(I)$ has finite codimension as $\mathbb{K}$-subspace of $\mathbb{K}[x]\times\mathbb{K}[y]$. Since we will need this feature later, let us record it as a lemma.

Lemma 2.4. *If* $I\subseteq\mathbb{K}[x,y]$ *has dimension zero, then there is a finite-dimensional* $\mathbb{K}$*-subspace* V *of* $\mathbb{K}[x]\times\mathbb{K}[y]$ *such that the direct sum* $V\oplus A(I)$ *is equal to* $\mathbb{K}[x]\times\mathbb{K}[y]$. *Moreover, we can compute a basis of such a* V*, and for every* $(f,g)\in\mathbb{K}[x]\times\mathbb{K}[y]$ *we can compute a* $(\tilde f,\tilde g)\in V$ *such that* $(f,g)-(\tilde f,\tilde g)\in A(I)$.

Proof. Let $p,q,(f_1,g_1),\dots,(f_d,g_d)$ be as in Algorithm 2.1. Note that, as a $\mathbb{K}$-vector space, $A(I)$ has the basis

$$\{(f_1,g_1),\dots,(f_d,g_d)\}\cup\{(x^kp,0):k\in\mathbb{N}\}\cup\{(0,y^kq):k\in\mathbb{N}\}.$$

Using row-reduction, it can be arranged that the f_i have pairwise distinct degrees. Note that, all f_i are nonzero by the choice of q. Let V be the $\mathbb{K}$-subspace of $\mathbb{K}[x]\times\mathbb{K}[y]$ generated by the pairs $(x^k,0)$ for all $k<\deg_x(p)$ which are not the degree of some f_i and the pairs $(0,y^k)$ for all $k<\deg_y(q)$. We have $V\oplus A(I)=\mathbb{K}[x]\times\mathbb{K}[y]$.

Given $(f,g)\in\mathbb{K}[x]\times\mathbb{K}[y]$, we compute $(\operatorname{rem}_x(f,p),\operatorname{rem}_y(g,q))$, and then eliminate all terms from the first component whose exponent is the degree of an f_i. The resulting pair $(\tilde f,\tilde g)$ is an element of V with $(f,g)-(\tilde f,\tilde g)\in A(I)$. □

3 PRINCIPAL IDEALS

We now consider the case where $I=\langle p\rangle$ is a principal ideal of $\mathbb{K}[x,y]$. If $p\in\mathbb{K}[x]\cup\mathbb{K}[y]$, the algebra $A(I)$ of separated polynomials is finitely generated, as we have seen in the proof of Proposition 2.2. It was shown in [15, Theorem 4.2] that, if p is separable, there is a separated multiple $f(x)-g(y)$ of p that divides any other separated multiple of it. We refer to $f(x)-g(y)$ as *the minimal separated multiple* of p. Moreover, [15, Theorem 2.3] implies that if $p\notin\mathbb{K}[x]\cup\mathbb{K}[y]$, then (f,g) is an algebra generator for $A(I)$. We note that, [15, Theorem 2.3] was reproven in [8], and generalized further in [1, 19]. The proof of [15, Theorem 4.2] was not constructive. In the following we provide a criterion that allows to decide if p is separable, and if it is, to compute its minimal separated multiple.

Our criterion is based on considering the highest graded component of the polynomial with respect to a certain grading. The separability of the highest component is a necessary but not a sufficient condition for the separability of a polynomial itself. Surprisingly, there is a weaker converse, that is, the minimal separated multiple of the highest component is equal to the highest component of the minimal separated multiple of p *if the latter exists* (see Theorem 3.5). This allows us to reduce the problem for a general not necessarily homogeneous polynomial to the same problem for a homogeneous polynomial (which is solved in Section 3.1) and solving a linear system. The resulting algorithm is presented in Section 3.3.

Since the case $p\in\mathbb{K}[x]\cup\mathbb{K}[y]$ is trivial, for the rest of the section, we assume that $p\in\mathbb{K}[x,y]\setminus(\mathbb{K}[x]\cup\mathbb{K}[y])$.

3.1 Homogeneous case

Definition 3.1.

(1) *A function* ω *from the set of monomials in* x *and* y *to* $\mathbb{R}$ *is called a* weight function *if there exist* $\omega_x,\omega_y\in\mathbb{Z}_{>0}$ *such that* $\omega(x^iy^j)=\omega_xi+\omega_yj$ *for every* $i,j\in\mathbb{Z}_{\geq0}$.
(2) *Two weight functions are considered to be* equivalent *if they differ by a constant non-zero factor.*
(3) *For a weight function* ω *and a nonzero polynomial* $p\in\mathbb{K}[x,y]$, $\omega(p)$ *is defined to be the maximum of the weights of the monomials of* p.
(4) *For a weight function* ω *and a polynomial* $p\in\mathbb{K}[x,y]$, *we define* the ω-leading part of p *(denoted by* $\operatorname{lp}_\omega(p)$*) as the sum of the terms of* p *of weight* $\omega(p)$.

In this subsection, we consider the case of p being homogeneous with respect to some weight function ω, that is, $\operatorname{lp}_\omega(p)=p$.

Proposition 3.2. *Let* ω *be a weight function, and let* $p\in\mathbb{K}[x,y]\setminus(\mathbb{K}[x]\cup\mathbb{K}[y])$ *satisfy* $\operatorname{lp}_\omega(p)=p$. *Then* p *is separable if and only if*

(1) p *involves a monomial only in* x, *and*
(2) *all the roots of* $p(x,1)$ *in the algebraic closure* $\overline{\mathbb{K}}$ *of* $\mathbb{K}$ *are distinct and the ratio of every two of them is a root of unity.*

Moreover, if p *is separable and* N *is the minimal number such that the ratio of every pair of roots of* $p(x,1)$ *is an* N*-th root of unity, then the weight of the minimal separated multiple of* p *is* $N\omega_x$.

Proof. Assume that p is separable, and let P be a separated multiple. Replacing P with $\operatorname{lp}_\omega(P)$ if necessary, we will further assume that $P=\operatorname{lp}_\omega(P)$. Since $P\notin\mathbb{K}[x]\cup\mathbb{K}[y]$ and is separated, P involves a monomial in x only, and hence, so does p.

Since P is ω-homogeneous and separated, it is of the form ax^m-by^n for some $a,b\in\mathbb{K}\setminus\{0\}$, so $p(x,1)\mid ax^m-b$. All roots of the latter are distinct and the ratio of each of them is an m-th root of unity. Hence, the same is true for $p(x,1)$. This proves the only-if part of the proposition.

To prove the remaining part of the proposition, let N be as in the statement of the proposition, and $\gamma\in\overline{\mathbb{K}}$ be a root of $p(x,1)$. Consider the ω-homogeneous Puiseux polynomial

$$P:=x^N-\gamma^Ny^{N\omega_x/\omega_y}.$$

We perform Euclidean division of P by p over the field F of Puiseux series in y over $\overline{\mathbb{K}}$. This will yield a representation $P=qp+r$, where q and r are also ω-homogeneous. Since $P(x,1)$ is divisible by $p(x,1)$, we see that $r(x,1)=0$. However, the ω-homogeneity of r implies that each of its coefficients with respect to x is a Puiseux monomial in y. Thus, $r=0$. Next, assume that $N\omega_x/\omega_y$ is not an integer. Then there is an automorphism σ of the Galois group of F over $\overline{\mathbb{K}}(y)$ that moves $y^{N\omega_x/\omega_y}$. Then

$$p\mid P-\sigma(P)\in F,$$

which is impossible. Therefore, P is a separated polynomial divisible by p of weight $N\omega_x$. □

Of course, because of symmetry, the statements of Proposition 3.2 also hold for y instead of x.

3.2 Reduction to the homogeneous case

We will start with a necessary condition for p being separable.

Lemma 3.3. *Let $p \in \mathbb{K}[x,y] \setminus (\mathbb{K}[x] \cup \mathbb{K}[y])$ be separable.*

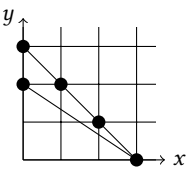
(1) There exists a unique (up to a constant factor) weight function ω such that $\mathrm{lp}_\omega(p)$ involves at least two monomials.
(2) The polynomial $\mathrm{lp}_\omega(p)$ is separable.

Proof. Let $q \in \mathbb{K}[x,y] \setminus \{0\}$ be such that qp is separated. Let $\deg_x qp = m$ and $\deg_y qp = n$. Define $\omega(x^i y^j) = ni + mj$. If $\mathrm{lp}_\omega(p)$ contains only one monomial, then every monomial in $\mathrm{lp}_\omega(qp)$ is divisible by it. This is impossible since $\mathrm{lp}_\omega(qp)$ involves both x^m and y^n.

To prove the uniqueness, assume that there are two nonequivalent weight functions ω_1 and ω_2 with this property. Since $\mathrm{lp}_{\omega_i}(qp) = \mathrm{lp}_{\omega_i}(q)\,\mathrm{lp}_{\omega_i}(p)$ for $i = 1, 2$, we have that both $\mathrm{lp}_{\omega_1}(qp)$ and $\mathrm{lp}_{\omega_2}(qp)$ contain at least two monomials. However, the only monomials of qp that can appear in the leading part are x^m and y^n, and there is a unique weight function so that they have the same weight.

The second claim of the lemma follows from $\mathrm{lp}_\omega(q)\,\mathrm{lp}_\omega(p) = \mathrm{lp}_\omega(qp)$. □

There is an analogous version of Lemma 3.3 with the lowest homogeneous part in place of the leading homogeneous part. However, even when both the lowest and the leading homogeneous part are separable, the whole polynomial need not be separable, as the following example shows.



Example 3.4. *For $p = (x^3 + x^2y + xy^2 + y^3) + y^2 \in \mathbb{Q}[x,y]$, the relevant weight function for the leading homogeneous part as in Lemma 3.3 is given by $\omega_x = \omega_y = 1$. It leads to the leading homogeneous part $x^3 + x^2y + xy^2 + y^3$. Analogously, the relevant weight function for the lowest homogeneous part is given by $\omega_x = 2, \omega_y = 3$. It leads to the lowest homogeneous part $x^3 + y^2$. Both the leading and the lowest homogeneous part are separable. We claim that p is not separable.*

Let ω be the weight function defined by $\omega(x^iy^j) = 2i + 3j$, so that the lowest homogeneous part of p is $x^3 + y^2$ (weight 6), and the next-to-lowest part is x^2y (weight 7). With respect to ω, any separated polynomial involving both variables only consists of homogeneous parts $ax^n + by^m$ whose weight $2n = 3m$ is a multiple of 6.

Assume that p is separable and let $q \in \mathbb{Q}[x,y] \setminus \{0\}$ be such that qp is separated. Write $q = q_0 + q_1 + \cdots$, where $q_0, q_1, \ldots$ are the lowest, the next-to-lowest, etc. homogeneous parts of q with respect to ω. The lowest homogeneous part of pq is then $q_0(x^3 + y^2)$, and since it must be separated and involve both variables, we have $\omega(q_0) = 0 \bmod 6$.

Because of $\omega(q_0x^2y) = \omega(q_0(x^3 + y^2)) + 1 = 1 \bmod 6$, none of the terms of q_0x^2y can appear in qp, so they must all be canceled by something. We must therefore have $\omega(q_1) = \omega(q_0) + 1$ and $q_0x^2y + q_1(x^3 + y^2) = 0$. This implies that $x^3 + y^2$ divides q_0, which in turn implies that the lowest homogeneous part $q_0(x^3 + y^2)$ of pq has a multiple factor. On the other hand, $q_0(x^3 + y^2) = ax^n + by^m$ for some $a, b \neq 0$, and every such polynomial is squarefree. This is a contradiction.

The main result of the section is the following "partial converse" of Lemma 3.3.

Theorem 3.5. *Let $p \in \mathbb{K}[x,y] \setminus (\mathbb{K}[x] \cup \mathbb{K}[y])$ be a separable polynomial. Let ω be the weight function given by Lemma 3.3, and let P be the minimal separated multiple of p. Then $\mathrm{lp}_\omega(P)$ is the minimal separated multiple of $\mathrm{lp}_\omega(p)$.*

Before proving the theorem, we will establish some combinatorial tools for dealing with divisors of separated polynomials extending the results of Cassels [10].

Notation 3.6. *Consider a separated polynomial $f(x) - g(y)$ with $\deg_x f = m$ and $\deg_y g = n$, where $m, n > 0$, and a weight function $\omega(x^iy^j) = in + jm$. We introduce a new variable t and consider two auxiliary equations*

$$f(x) = t \quad \text{and} \quad g(y) = t.$$

We solve these equations with respect to x and y in $\overline{\mathbb{K}(t)}$, the algebraic closure of $\mathbb{K}(t)$. Let the solutions be $\alpha_0, \ldots, \alpha_{m-1}$ and $\beta_0, \ldots, \beta_{n-1}$, respectively. Then every element π of $\mathrm{Gal}(\overline{\mathbb{K}(t)}/\mathbb{K}(t))$, the Galois group of $\overline{\mathbb{K}(t)}$ over $\mathbb{K}(t)$, acts on $\mathbb{Z}_m \times \mathbb{Z}_n$ by

$$\pi(i,j) := (i',j') \iff (\pi(\alpha_i), \pi(\beta_j)) = (\alpha_{i'}, \beta_{j'}).$$

Let $G \subseteq \mathrm{S}_m \times \mathrm{S}_n$ be the group of permutations induced on $\mathbb{Z}_m \times \mathbb{Z}_n$ by this action.

Notation 3.7. *For a subset $T \subseteq \mathbb{Z}_m \times \mathbb{Z}_n$, and $(i,j) \in \mathbb{Z}_m \times \mathbb{Z}_n$, we introduce*

$$T_{i,*} := \{k \mid (i,k) \in T\} \text{ and } T_{*,j} := \{k \mid (k,j) \in T\}.$$

Lemma 3.8. *Let $T \subseteq \mathbb{Z}_m \times \mathbb{Z}_n$ be a G-invariant subset. Then $|T_{0,*}| = |T_{1,*}| = \ldots = |T_{m-1,*}|$ and $|T_{*,0}| = |T_{*,1}| = \ldots = |T_{*,n-1}|$.*

Proof. We show that $|T_{0,*}| = |T_{1,*}|$, the rest is analogous. First, we observe that $f(x) - t$ is irreducible over $\mathbb{K}(t)$. If it was not, it would be reducible over $\mathbb{K}[t]$ due to Gauss's lemma. The latter is impossible because $f(x) - t$ is linear in t and does not have factors in $\mathbb{K}[x]$. The irreducibility of $f(x) - t$ implies that its Galois group acts transitively on the roots. In particular, there exists $\pi \in \mathrm{Gal}(\overline{\mathbb{K}(t)}/\mathbb{K}(t))$ such that $\pi(\alpha_0) = \alpha_1$. Hence, π maps $T_{0,*}$ to $T_{1,*}$, and we have $|T_{0,*}| \leqslant |T_{1,*}|$. The reverse inequality is analogous. □

Lemma 3.9 (cf. [10, p. 9-10]). *Let $T \subseteq \mathbb{Z}_m \times \mathbb{Z}_n$ be a G-invariant subset. There exists a divisor p of $f(x) - g(y)$, unique up to a multiplicative constant, such that*

$$T = \{(i,j) \in \mathbb{Z}_m \times \mathbb{Z}_n \mid p(\alpha_i, \beta_j) = 0\}. \tag{1}$$

Proof. *Existence.* Let $T_{0,*} = \{j_1, \ldots, j_s\}$. Since $f(\alpha_0) = t$, we have $\mathbb{K}(\alpha_0) \supseteq \mathbb{K}(t)$, so every element of $\mathrm{Gal}(\overline{\mathbb{K}(t)}/\mathbb{K}(\alpha_0))$ leaves T invariant. If α_0 is fixed, then $\beta_{j_1}, \ldots, \beta_{j_s}$ are permuted. Therefore, the polynomial $(y - \beta_{j_1})(y - \beta_{j_2}) \ldots (y - \beta_{j_s})$ is invariant under the action of $\mathrm{Gal}(\overline{\mathbb{K}(t)}/\mathbb{K}(\alpha_0))$. Hence, by the fundamental theorem of Galois theory, it is a polynomial in $\mathbb{K}(\alpha_0)[y]$. Since, by construction, it divides $f(\alpha_0) - g(y)$ over $\mathbb{K}(\alpha_0)$, and α_0 and y are algebraically independent, it in fact belongs to $\mathbb{K}[\alpha_0, y]$. Replacing α_0 by x, we find a polynomial $p \in \mathbb{K}[x,y]$, which divides $f(x) - g(y)$ in $\mathbb{K}[x,y]$.

Let $(i,j) \in \mathbb{Z}_m \times \mathbb{Z}_n$. Since $\mathrm{Gal}(\overline{\mathbb{K}(t)}/\mathbb{K}(t))$ acts transitively on the roots of $f(x) - t$ (see the proof of Lemma 3.8), there is an automorphism π with $\pi(\alpha_i) = \alpha_0$. Let $\beta_{j'} = \pi(\beta_j)$. We then have

$$p(\alpha_i, \beta_j) = 0 \iff p(\alpha_0, \beta_{j'}) = 0 \iff j' \in T_{0,*} \iff (i,j) \in T.$$

Uniqueness. It remains to prove that p is unique up to a multiplicative constant. Assume that $\tilde{p}$ is another divisor of $f(x) - g(y)$ such that $\tilde{p}(\alpha_i, \beta_j) = 0$ for all $(i, j) \in T$. The same argument which proved that p is a divisor of $f(x) - g(y)$ applies to show that p is a divisor of $\tilde{p}$ in $\mathbb{K}[x, y]$, and vice versa. Hence, they only differ by a multiplicative constant. □

Lemma 3.10. *Let $T \subseteq \mathbb{Z}_m \times \mathbb{Z}_n$ be a G-invariant subset. The unique factor p corresponding to $T \subseteq \mathbb{Z}_m \times \mathbb{Z}_n$ (see Lemma 3.9) is separated if and only if*

$$\forall\, i, j \in \mathbb{Z}_m : (T_{i,*} \cap T_{j,*} = \varnothing) \text{ or } (T_{i,*} = T_{j,*}) \tag{2}$$

Proof. Assume that T satisfies (2), and let $T_{0,*} = \{j_1, \ldots, j_s\}$. Consider the corresponding polynomial p constructed in the proof of Lemma 3.9, which is of the form

$$p(x, y) = y^s + a_{s-1}(x)y^{s-1} + \cdots + a_0(x),$$

where, for every $0 \leqslant i < s$ and $0 \leqslant j < m$, $a_i(\alpha_j)$ is (up to sign) the $s - i$-th elementary symmetric polynomial in $\{\beta_k \mid k \in T_{j,*}\}$.

Since $p \mid f(x) - g(y)$, we have $\mathrm{lp}_\omega(p) \mid \mathrm{lp}_\omega(f(x) - g(y)) = ax^m - by^n$, with $a, b \in \mathbb{K} \setminus \{0\}$. Hence, y^s belongs to $\mathrm{lp}_\omega(p)$, and so $\omega(a_i(x)y^i) \leqslant \omega(y^s) = ms$ for all $i \in \{0, \ldots, s-1\}$, This implies

$$\deg_x a_i(x) \leqslant \frac{ms - mi}{n} = (s - i)\frac{m}{n}.$$

Since T is the disjoint union of the $T_{i,*}$'s and of the $T_{*,j}$'s, respectively, whose cardinality, by Lemma 3.8, does not depend on i and j, and $T_{0,*}$, by definition, consists of s elements, we find that $ms = |T| = n|T_{*,j_1}|$, in particular $\ell := |T_{*,j_1}| = \frac{ms}{n}$. Hence there exist $0 = i_1 < i_2 < \ldots < i_\ell < m$ such that $j_1 \in T_{i_1,*} \cap \ldots \cap T_{i_\ell,*}$ and so, by (2), $T_{i_1,*} = \ldots = T_{i_\ell,*}$. This shows that the polynomial $a_j(x) - a_j(\alpha_0)$ has at least ℓ pairwise distinct roots, $\alpha_{i_1}, \ldots, \alpha_{i_\ell}$, while it has degree less than ℓ for $0 < j < s$. Hence, it is the zero polynomial, and $a_j(x)$ is a constant (which we denote by a_j). Therefore, p is separated and of the form $p(x, y) = f_0(x) - g_0(y)$ with $f_0(x) = a_0(x)$ and $g_0(y) = -(y^s + a_{s-1}y^{s-1} + \cdots + a_1 y)$.

To prove the other implication, let $p(x, y) = f_0(x) - g_0(y)$ be a separated factor of $f(x) - g(y)$. It is sufficient to show that

$$(i, j),\ (i', j),\ (i, j') \in T \implies (i', j') \in T.$$

Indeed, $(i, j),\ (i', j) \in T$ implies that $f_0(\alpha_i) = f_0(\alpha_{i'})$, so that $f_0(\alpha_i) - g_0(\beta_{j'}) = 0$ implies that $f_0(\alpha_{i'}) - g_0(\beta_{j'}) = 0$, i.e. $(i', j') \in T$. □

Lemma 3.10 motivates the following definition.

Definition 3.11. *(1) A subset $T \subseteq \mathbb{Z}_m \times \mathbb{Z}_n$ is called* separated *if it satisfies* (2)*, that is*

$$\forall\, i, j \in \mathbb{Z}_m : (T_{i,*} \cap T_{j,*} = \varnothing) \text{ or } (T_{i,*} = T_{j,*}).$$

(2) The intersection of all separated subsets containing $T \subseteq \mathbb{Z}_m \times \mathbb{Z}_n$ is called the separated closure *of T and denoted by T^{sep}. Notice that the separated closure is separated.*

Example 3.12. *(1) Let $f(x) = x^4$ and $g(y) = y^4 + 2y^2 + 1$. The group of permutations on pairs of roots of $f(x) - t$ and $g(y) - t$ is generated by $((0123), (0123))$, $((0321), (03)(12))$ and $(id, (02))$. According to $f(x) - g(y)$ having two separated irreducible factors, $x^2 - y^2 - 1$ and $x^2 + y^2 + 1$, we find that there are two orbits, each of them forming a separated set (Figure 1).*

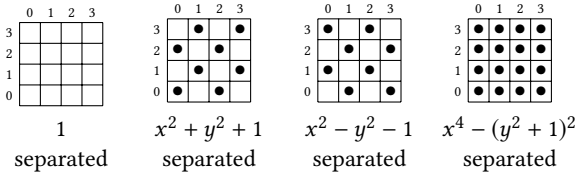


Figure 1: The factors of $x^4 - (y^2 + 1)^2$ in $\mathbb{Q}[x, y]$ and the sets $T \subseteq \mathbb{Z}_4^2$ corresponding to them.

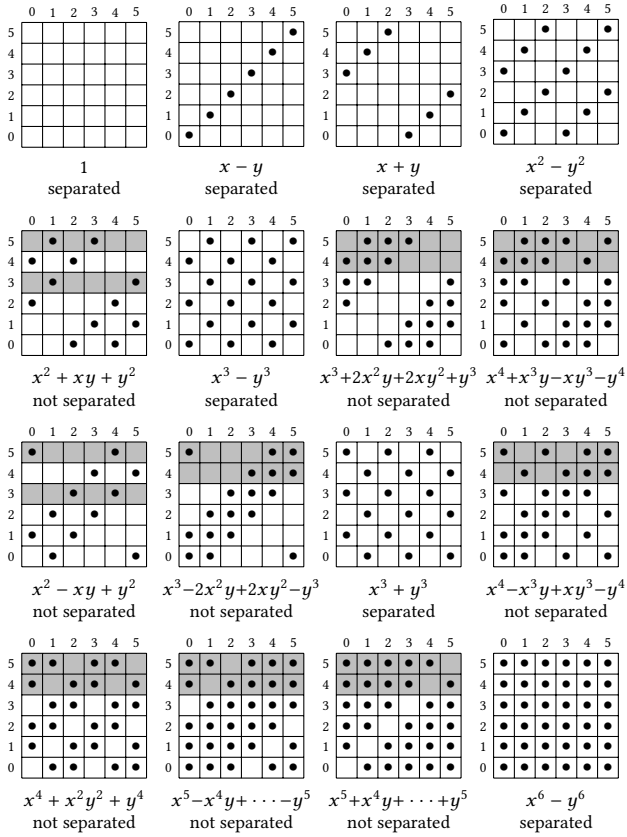


Figure 2: The factors of $x^6 - y^6$ in $\mathbb{Q}[x, y]$ and the sets $T \subseteq \mathbb{Z}_6^2$ corresponding to them. For the unseparated cases, we highlight one choice of two incompatible rows.

(2) Let $f(x) - g(y) = x^6 - y^6$. Let $t^{1/6} \in \overline{\mathbb{C}(t)}$ be any 6th root of t, and let ϵ be a primitive 6th root of unity. Then the polynomials $f(x) - t$ and $g(y) - t$ have the same roots, namely:

$$\alpha_i = \beta_i = \epsilon^i t^{1/6}, \quad i \in \{0, \ldots, 5\}.$$

The Galois group of $\overline{\mathbb{C}(t)}$ permutes these elements cyclically, so the induced action on $\mathbb{Z}_6^2$ is generated by $((012345), (012345))$. Figure 2 shows the sets T for the various factors of $x^6 - y^6$. Observe that T is separated if and only if the corresponding factor is separated. Observe also that multiplying two factors corresponds to taking the union of the corresponding sets T.

Lemma 3.13. *Let $T \subseteq \mathbb{Z}_m \times \mathbb{Z}_n$ be invariant with respect to $G \subseteq \mathrm{S}_m \times \mathrm{S}_n$. Then T^{sep} is also G-invariant.*

Proof. Let $\pi = (\sigma, \tau) \in \mathrm{S}_m \times \mathrm{S}_n$, and let $S \subseteq \mathbb{Z}_m \times \mathbb{Z}_n$ be a separated set. Since $\pi(S)_{i,*} = \tau(S_{\sigma(i),*})$, we find that $\pi(S)$ is separated as well.

Assume that T^{sep} is not G-invariant, that is, there exists a $\pi \in G$ such that $\pi(T^{\mathrm{sep}}) \neq T^{\mathrm{sep}}$. As we have shown, $\pi(T^{\mathrm{sep}})$ is separated, hence so is $S := T^{\mathrm{sep}} \cap \pi(T^{\mathrm{sep}})$. Observe that, since $\pi(T^{\mathrm{sep}}) \neq T^{\mathrm{sep}}$, $S \subsetneq T^{\mathrm{sep}}$. Since T is G-invariant, $T \subseteq \pi(T^{\mathrm{sep}})$, so $T \subseteq S$. This contradicts the minimality of T^{sep}. □

Proof of Theorem 3.5. We use Notation 3.6 with $\overline{\mathbb{K}(t)}$ being identified with a subfield of the field F of Puiseux series in t^{-1} over $\overline{\mathbb{K}}$. Let $\alpha_0, \ldots, \alpha_{m-1}$ and $\beta_0, \ldots, \beta_{n-1}$ denote the roots of $f(x) - t$ and $g(y) - t$ and $\overline{\alpha}_0, \ldots, \overline{\alpha}_{m-1}$ and $\overline{\beta}_0, \ldots, \overline{\beta}_{n-1}$ their highest degree terms. Observe that the highest degree terms are proportional to $t^{1/n}$ and $t^{1/m}$, and hence they are the roots of $\mathrm{lp}_\omega(f(x)) - t$ and $\mathrm{lp}_\omega(g(y)) - t$, respectively. We define

$$T = \{(i,j) \in \mathbb{Z}_m \times \mathbb{Z}_n \mid p(\alpha_i, \beta_j) = 0\},$$
$$\overline{T} = \{(i,j) \in \mathbb{Z}_m \times \mathbb{Z}_n \mid \mathrm{lp}_\omega(p)(\overline{\alpha}_i, \overline{\beta}_j) = 0\}.$$

If $\mathrm{lp}_\omega(P)$ were not the minimal separated multiple of $\mathrm{lp}_\omega(p)$, by Lemma 3.10, we would have $\overline{T}^{\mathrm{sep}} \subsetneq \mathbb{Z}_m \times \mathbb{Z}_n$. Therefore, it is sufficient to show that $\overline{T}^{\mathrm{sep}} = \mathbb{Z}_m \times \mathbb{Z}_n$.

Since

$$p(\alpha_i, \beta_j) = 0 \implies \mathrm{lp}_\omega(p)(\overline{\alpha_i}, \overline{\beta_j}) = 0,$$

we have $T \subseteq \overline{T}$. By assumption, P is the minimal separated multiple of p, so, by Lemma 3.13, $T^{\mathrm{sep}} = \mathbb{Z}_m \times \mathbb{Z}_n$. Since $T^{\mathrm{sep}} \subseteq \overline{T}^{\mathrm{sep}}$, this implies that $\overline{T}^{\mathrm{sep}} = \mathbb{Z}_m \times \mathbb{Z}_n$, and finishes the proof. □

3.3 Algorithm

The algorithm for finding a generator of the algebra of separated polynomials of a principal ideal $\langle p \rangle$ is based on the results above. First, it uses Theorem 3.5 to reduce the situation to a homogeneous polynomial for a suitable grading, then, it uses Proposition 3.2 to find a degree bound for the minimal separated multiple, and finally, it uses linear algebra to determine if such a multiple exists.

Algorithm 3.14. *Input: $p \in \mathbb{K}[x,y] \setminus (\mathbb{K}[x] \cup \mathbb{K}[y])$.*
Output: $a \in \mathbb{K}[x] \times \mathbb{K}[y]$ such that $\mathbb{K}[a] = A(\langle p \rangle)$. The algorithm returns $a = (1,1)$ iff $A(\langle p \rangle) \cong \mathbb{K}$.

1 *let $\omega_x, \omega_y \in \mathbb{N}$ be maximal such that p contains monomials $x^{\omega_y} y^0$ and $x^0 y^{\omega_x}$. Such parameters exist because p is not univariate.*
2 *set $h = \mathrm{lp}_\omega(p)$ with $\omega(x^i y^j) := \omega_x i + \omega_y j$.*
3 *if h does not contain x^{ω_y}, return $(1,1)$.*
4 *let $\{\zeta_1, \ldots, \zeta_m\} \subseteq \overline{\mathbb{K}}$ be the roots of $h(x,1) \in \mathbb{K}[x]$. If any of them is not a simple root, return $(1,1)$.*
5 *let $N \in \mathbb{N}$ be minimal such that $(\zeta_i/\zeta_j)^N = 1$ for all i,j. If no such N exists, return $(1,1)$.*
6 *make an ansatz*

$$f = \sum_{i=0}^{N} a_i x^i, \quad g = \sum_{j=0}^{N\omega_x/\omega_y} b_j y^j,$$

compute $\mathrm{rem}_x(f-g,p)$ in $\mathbb{K}(a_0, \ldots, a_N, b_0, \ldots, b_{N\omega_x/\omega_y}, y)[x]$. The result lives in $\mathbb{K}[a_0, \ldots, a_N, b_0, \ldots, b_{N\omega_x/\omega_y}, y, x]$ because the leading coefficient of p is in $\mathbb{K}$.
7 *equate the coefficients of $\mathrm{rem}_x(f-g,p)$ with respect to x, y to zero and solve the resulting linear system for the unknowns a_i, b_j.*
8 *if there is a nonzero solution, return the corresponding pair (f,g), otherwise return $(1,1)$.*

When $\mathbb{K}$ is a number field, Step 5 can be carried out as follows: for each ratio ζ_i/ζ_j, one should check whether the minimal polynomial of this ratio over $\mathbb{Q}$ is a cyclotomic polynomial Φ_n and, if yes, return such n. This check can be performed using a bound from [18, Theorem 15] that yields the upper bound on n based on the degree of the polynomial.

Proposition 3.15. *Algorithm 3.14 is correct.*

Proof. The algorithm consists of an application of the results of the previous section and a handling of degenerate cases not covered by these results. In Steps 3–5, it is correct to return $(1,1)$ in the indicated situations because Proposition 3.2 implies that h is not separable in these cases, which in combination with Lemma 3.3 implies that p is not separable either.

By Proposition 3.2, when h has a separated multiple at all, it has one of weight $N\omega_x$, and by Theorem 3.5, when p has a separated multiple at all, it also has one of weight $N\omega_x$. Therefore, if p has a separated multiple, it will have one of the shape set up Step 6. For $f - g$ to be a separated multiple of p is equivalent to $\mathrm{rem}_x(f-g,p) = 0$, which we can safely view as univariate division with respect to the variable x because the leading coefficient of p with respect to x does not contain y (nor any of the undetermined coefficients). It is checked in Step 7 whether there is a way to instantiate the undetermined coefficients in such a way that this remainder becomes zero. If so, any such way translates into a separated multiple, and by [15, Theorem 2.3], it is a generator of $A(I)$. If there is no non-zero solution, it is correct to return $(1,1)$. □

4 ARBITRARY BIVARIATE IDEALS

The case of an arbitrary ideal $I \subseteq \mathbb{K}[x,y]$ is reduced to the two cases discussed in Sections 2 and 3. Every ideal $I \subseteq \mathbb{K}[x,y]$ can be written as $I = \bigcap_{i=1}^{k} P_i$, where the P_i's are primary ideals. Unless $I = \{0\}$ or $I = \langle 1 \rangle$, these primary ideals have dimensions zero or one. Primary ideals in $\mathbb{K}[x,y]$ of dimension 1 must be principal ideals, because $\dim(P_i) = 1$ together with Bezout's theorem implies that P_i cannot contain any elements p, q with $\gcd(p,q) = 1$, and then P_i being primary implies that P_i is generated by some power of an irreducible polynomial.

The intersection of zero-dimensional ideals is zero-dimensional and the intersection of principal ideals is principal, so there exists a zero-dimensional ideal I_0 and a principal ideal I_1 such that $I = I_0 \cap I_1$. These ideals are obtained as the intersections of the respective primary components of I. When $I_0 = \langle 1 \rangle$ or $I_1 = \langle 1 \rangle$, we have $I = I_1$ or $I = I_0$, respectively, and are in one of the cases already considered. Assume now that I_1, I_0 are both different from $\langle 1 \rangle$.

In order to use the results of Section 3, we have to make sure that the generator of I_1 contains both variables. If this is not the case, say if $I_1 = \langle h \rangle$ for some $h \in \mathbb{K}[x] \setminus \mathbb{K}$, then the separated polynomials in I are precisely the elements of $I \cap \mathbb{K}[x]$. If p is such

that $\langle p\rangle = I \cap \mathbb{K}[x]$, then the pairs $(x^i p, 0)$ for $i = 0, \dots, \deg_x p - 1$ are generators of $A(I)$ (see the proof of Proposition 2.2), so this case is settled. Therefore, from now on we assume that the generator of I_1 contains both the variables.

We can compute generators of the algebra $A(I_0) \subseteq \mathbb{K}[x]\times\mathbb{K}[y]$ of separated polynomials in I_0 as described in Section 2 and a generator of the algebra $A(I_1) \subseteq \mathbb{K}[x] \times \mathbb{K}[y]$ of separated polynomials in I_1 as described in Section 3. Clearly, the algebra $A(I) \subseteq \mathbb{K}[x]\times\mathbb{K}[y]$ of separated polynomials in I is $A(I) = A(I_0)\cap A(I_1)$. It thus remains to compute generators for this intersection. In order to do so, we will exploit that the codimension of $A(I_0)$ as $\mathbb{K}$-subspace of $\mathbb{K}[x]\times\mathbb{K}[y]$ is finite (Lemma 2.4), and that $A(I_1) = \mathbb{K}[a]$ for some $a \in \mathbb{K}[x] \times \mathbb{K}[y]$. We have to find all polynomials p such that $p(a) \in A(I_0)$. Polynomials p with a prescribed finite set of monomials can be found with the help of Lemma 2.4 as follows.

Algorithm 4.1. *Input: $a \in \mathbb{K}[x] \times \mathbb{K}[y]$, $A(I_0)$ and V as in Lemma 2.4, and a finite set $S = \{s_1, \dots, s_m\} \subseteq \mathbb{N}$.*
Output: a $\mathbb{K}$-vector space basis of the space of all polynomials p with $p(a) \in A(I_0)$ such that p involves only monomials with exponents in S.

1 *for $i = 1, \dots, m$, compute $r_i \in V$ such that $a^{s_i} - r_i \in A(I_0)$*
2 *compute a basis B of the space of all $(c_1, \dots, c_m) \in \mathbb{K}^m$ with $c_1 r_1 + \cdots + c_m r_m = 0$*
3 *for every element $(c_1, \dots, c_m) \in B$, return $c_1 t^{s_1} + \cdots + c_m t^{s_m}$.*

Proposition 4.2. *Algorithm 4.1 is correct.*

Proof. If $(c_1, \dots, c_m) \in \mathbb{K}^m$ is such that $\sum_{i=1}^m c_i a^{s_i} \in A(I_0)$, then $\sum_{i=1}^m c_i r_i \in A(I_0)$, and since $r_i \in V$ for all i and $A(I_0)\cap V = \{0\}$, we have $\sum_{i=1}^m c_i r_i = 0$. Therefore $(c_1, \dots, c_m)$ is among the vectors computed in step 2, so the algorithm does not miss any solutions. Conversely, if $(c_1, \dots, c_m) \in \mathbb{K}^m$ is such that $\sum_{i=1}^m c_i r_i = 0$, then $\sum_{i=0}^m c_i a^{s_i} = \sum_{i=0}^m c_i (a^{s_i} - r_i) \in A(I_0)$, so the algorithm does not return any wrong solutions. □

To find a set of generators of $A(I_0)\cap A(I_1)$, we apply Algorithm 4.1 repeatedly. First call it with $S = \{1, \dots, \dim V + 1\}$. Since $|S| > \dim V$, the output must contain at least one nonzero polynomial p_1. If d_1 is its degree, we can restrict the search for further generators to subsets S of $\mathbb{N} \setminus d_1\mathbb{N}$, because when q is such that $q(a) \in A(I_0)$, then we can subtract a suitable linear combination of powers of p_1 to remove from q all monomials whose exponents are multiples of d_1. When $d_1 = 1$, we have $A(I_0)\cap A(I_1) = \mathbb{K}[a]$ and are done. Otherwise, $\mathbb{N} \setminus d_1\mathbb{N}$ is still an infinite set, so we can choose $S \subseteq \mathbb{N} \setminus d_1\mathbb{N}$ with $|S| > \dim V$ and call Algorithm 4.1 to find another nonzero polynomial p_2, say of degree d_2. The search for further generators can be restricted to polynomials consisting of monomials whose exponents belong to $\mathbb{N}\setminus(d_1\mathbb{N}+d_2\mathbb{N})$. We can continue to find further generators of degrees $d_3, d_4, \dots$ with $d_i \in \mathbb{N} \setminus (d_1\mathbb{N} + \cdots + d_{i-1}\mathbb{N})$ for all i. Since the monoid $(\mathbb{N}, +)$ has the ascending chain condition, this process must come to an end.

The end is clearly not reached as long as $g := \gcd(d_1, \dots, d_m) \neq 1$, because then $\mathbb{N} \setminus g\mathbb{N}$ is an infinite subset of $\mathbb{N} \setminus (d_1\mathbb{N} + \cdots + d_m\mathbb{N})$. Once we have reached $g = 1$, it is well known [2, 17] that $\mathbb{N} \setminus (d_1\mathbb{N} + \cdots + d_m\mathbb{N})$ is a finite set, and there are algorithms [5] for computing its largest element (known as the Frobenius number of $d_1, \dots, d_m$). We can therefore constructively decide when all generators have been found.

Putting all steps together, our algorithm for computing the separated polynomials in an arbitrary ideal of $\mathbb{K}[x, y]$ works as follows. We use the notation $\langle d_1, \dots, d_m\rangle$ for the submonoid $d_1\mathbb{N}+\cdots+d_m\mathbb{N}$ generated by $d_1, \dots, d_m$ in $\mathbb{N}$.

Algorithm 4.3. *Input: an ideal $I \subseteq \mathbb{K}[x, y]$, given as a finite set of ideal generators*
Output: a finite set of generators for the algebra $A(I)$ of separated polynomials of I

1 *if $\dim I = 0$, call Algorithm 2.1, return the result.*
2 *compute a zero-dimensional ideal I_0 and a principal ideal $I_1 = \langle h\rangle$ with $I = I_0 \cap I_1$ (for example, using Gröbner bases [4] and the remarks at the beginning of this section).*
3 *if $h \in \mathbb{K}[x]$, compute p such that $\langle p\rangle = I \cap \mathbb{K}[x]$, return the pairs $(x^i p, 0)$ for $i = 0, \dots, \deg_x p - 1$. Likewise if $h \in \mathbb{K}[y]$.*
4 *call Algorithm 2.1 to get generators of $A(I_0)$, and let V be as in Lemma 2.4.*
5 *call Algorithm 3.14 to get an $a \in \mathbb{K}[x] \times \mathbb{K}[y]$ with $A(I_1) = \mathbb{K}[a]$. If $A(I_1) \cong \mathbb{K}$, return $(1, 1)$.*
6 *$G = \emptyset$, $\Delta = \emptyset$.*
7 *while $\gcd(\Delta) \neq 1$, do:*
8 *select a set $S \subseteq \mathbb{N}\setminus\langle\Delta\rangle$ with $|S| > \dim V$ and call Algorithm 4.1 to find a nonzero polynomial p with $p(a) \in A(I_0)$ consisting only of monomials with exponents in S.*
9 *$G = G \cup \{p\}$, $\Delta = \Delta \cup \{\deg_x p\}$*
10 *call Algorithm 4.1 with $S = \mathbb{N} \setminus \langle\Delta\rangle$ (which is now a computable finite set) and add the resulting polynomials to G.*
11 *return G*

An implementation of the algorithm in Mathematica can be found on the website of the second author. Incidentally, the algorithm also shows that $A(I)$ is always a finitely generated $\mathbb{K}$-algebra.

Example 4.4. *For the ideal*

$$I = \langle (x^2 - xy + y^2)(x^3 - 2xy^2 - 1), (x^2 - xy + y^2)(y^3 - 2x^2y - 1)\rangle$$

we have $I_0 = \langle x^3 - 2xy^2 - 1, y^3 - 2x^2y - 1\rangle$ and $I_1 = \langle x^2 - xy + y^2\rangle$. Algorithm 2.1 yields a somewhat lengthy list of generators for $A(I_0)$ from which it can be read off that a suitable choice for V is the $\mathbb{K}$-vector space generated by $(0, y^i)$ for $i = 0, \dots, 8$. In particular, $\dim V = 9$. Algorithm 3.14 yields $A(I_1) = \mathbb{K}[(x^3, -y^3)]$.

Making an ansatz for a polynomial p of degree at most 10 such that $p(a) \in A(I_0)$, we find a solution space of dimension 7. Its lowest degree element is $t^4 - 2t^2$, giving rise to the element $(x^{12} - 2x^6, y^{12} - 2y^6)$ of $A(I_0) \cap A(I_1)$. If we discard the other solutions and continue with the next iteration, we search for polynomials p whose support is contained $\{x^s : s \in S\}$ for $S = \{1, 2, 3, 5, 6, 7, 9, 10, 11, 13\}$. Again, the solution space turns out to have dimension 7. The lowest degree element is now $9t^5 - 26t^3 + 17$. Since $\gcd(4, 5) = 1$, we can exit the while loop. In step 10 of the algorithm, we get $S = \{1, 2, 3, 6, 7, 11\}$, and this exponent set leads to a solution space of dimension three, generated by the polynomials $81t^6 - 323t^3$, $81t^7 - 539t^3 + 458$, and $6561t^{11} - 191125t^3 + 184564$. The resulting generators of $A(I) = A(I_0) \cap A(I_1)$ are therefore the pairs $p((x^3, -y^3))$ where p runs through the five polynomials found by the algorithm.

5 MORE THAN TWO VARIABLES

It is a natural question whether anything more can be said about the case of several variables. Incidentally, a multivariate version would be needed in order to solve the combinatorial problem that motivated this research in the first place.

Algorithm 2.1 for bivariate zero-dimensional ideals also holds for zero-dimensional ideals of $\mathbb{K}[x_1, \ldots, x_n, y_1, \ldots, y_m]$ for arbitrary n, m. Also Lemma 2.4 generalizes without problems. We believe that with some further work, our results for principal ideals can also be generalized to the case of several variables. However, in general, not every polynomial ideal with more than two variables is the intersection of a principal ideal and a zero-dimensional ideal, so the route taken in Section 4 is blocked. Also, as the next example shows we cannot expect an algorithm that finds the algebra of separated polynomials for an arbitrary ideal $I \subseteq \mathbb{K}[x_1, \ldots, x_n, y_1, \ldots, y_m]$, since it does not need to be finitely generated.

Example 5.1 ($A(I)$ is not necessarily finitely generated). *It is shown in [16, Example 1.3] that the algebra*

$$R := \mathbb{C}[t_1^2, t_1^3, t_2] \cap \mathbb{C}[t_1^2, t_2 - t_1] \subset \mathbb{C}[t_1, t_2]$$

is not finitely generated. Consider the ideal

$$\begin{aligned} I = \langle & x_1 - t_1^2, x_2 - t_1^3, x_3 - t_2, \\ & y_1 - t_1^2, y_2 - (t_2 - t_1) \rangle \cap \mathbb{C}[x_1, x_2, x_3, y_1, y_2] \\ = \langle & x_1 - y_1, -x_2 + x_3 y_1 - y_1 y_2, x_3^2 - y_1 - 2x_3 y_2 + y_2^2 \rangle. \end{aligned}$$

We claim that $A(I) \cong R$ as $\mathbb{C}$-algebras, implying that $A(I)$ is not finitely generated. We show that $\phi\colon A(I) \to R$ defined by $\phi(f, g) = f(t_1^2, t_1^3, t_2)$ is an isomorphism:

- *ϕ is well-defined (the image is contained in $R \subseteq \mathbb{C}[t_1^2, t_1^3, t_2]$). To see this, note that, $(f, g) \in A(I)$ means $f - g \in I$, which by definition of I means $f(t_1^2, t_1^3, t_2) = g(t_1^2, t_2 - t_1)$. Therefore, $f(t_1^2, t_1^3, t_2) \in \mathbb{C}[t_1^2, t_2^3, t_2] \cap \mathbb{C}[t_1^2, t_2 - t_1] = R$.*
- *ϕ is surjective. For every $p \in R$ there exist polynomials f, g with $p = f(t_1^2, t_1^3, t_2) = g(t_1^2, t_2 - t_1)$. By definition of I we have $f(x_1, x_2, x_3) - g(y_1, y_2) \in I$, hence $(f, g) \in A(I)$. Now $\phi(f) = p$, so p is in the image of ϕ.*
- *ϕ is injective. This follows from $I \cap \mathbb{C}[y_1, y_2] = \{0\}$.* □

It would still make sense to ask for an algorithm that decides whether $A(I)$ is nontrivial. We do not have such an algorithm, but being able to solve the problem in the bivariate case gives rise to a necessary condition.

Proposition 5.2. *Let*

$$\xi\colon \mathbb{K}[x_1, \ldots, x_n] \to \mathbb{K}[x] \quad \text{and} \quad \eta\colon \mathbb{K}[y_1, \ldots, y_m] \to \mathbb{K}[y]$$

be two homomorphisms, and let $I \subseteq \mathbb{K}[x_1, \ldots, x_n, y_1, \ldots, y_m]$ be an ideal such that

$$I \cap \mathbb{K}[y_1, \ldots, y_m] = \{0\} \quad \text{and} \quad (\mathrm{id} \otimes \eta)(I) \cap \mathbb{K}[x_1, \ldots, x_n] = \{0\}.$$

If the algebra of separated polynomials of I is non-trivial, then so is the algebra of separated polynomials of $J := (\xi \otimes \eta)(I) \subseteq \mathbb{K}[x, y]$.

Proof. Let (f, g) be an arbitrary, non-constant element of $A(I)$. If $(\xi(f), \eta(g)) \in A(J)$ were a $\mathbb{K}$-multiple of $(1, 1)$, we would find that $f - \eta(g)$ were an element of $(\mathrm{id} \otimes \eta)(I) \cap \mathbb{K}[x_1, \ldots, x_n]$, and hence, by our assumption, that f itself were a constant. So $f - g \in I \cap \mathbb{K}[y_1, \ldots, y_m]$, and hence, by assumption, $g = f$ is a constant as well, contradicting that (f, g) is not a constant. □

The examples below show different reasonable choices for homomorphisms ξ and η.

Example 5.3. *Consider the polynomial $p = x^2 + xy_1y_2 + y_1^2 + y_2^2$. Let $\xi = \mathrm{id}$ and let η be defined by $\eta(y_1) = y$, $\eta(y_2) = 2$. Notice that η is just the evaluation of y_2 at 2. Then $(\xi \otimes \eta)(p) = x^2 + 2xy_1 + y_1^2 + 4$, a polynomial that is not separable. Hence p is not separable.*

Example 5.4. *Consider the polynomial $p = x^2 + xy_1 + y_1^2 + y_2^4$. We cannot use the same strategy as in the previous example because any evaluation of y_1 or y_2 results in a separable polynomial. Nevertheless, the homomorphism defined by $\xi(x) = x$, $\eta(y_1) = y^2$, and $\eta(y_2) = y$ maps p to $(\xi \otimes \eta)(p) = x^2 + xy^2 + 2y^4$, a polynomial which is not separable. So p is not separable either.*

Acknowledgements. We thank Erhard Aichinger and Josef Schicho for sharing their thoughts on the topic and for providing pointers to the literature. We also thank the referees for their careful reading and their valuable suggestions.

REFERENCES

[1] Erhard Aichinger and Stefan Steinerberger. A proof of a theorem by Fried and MacRae and applications to the composition of polynomial functions. *Archiv der Mathematik*, 97:115–124, 2011.
[2] Jorge L. Ramírez Alfonsín. *The Diophantine Frobenius Problem.* Oxford University Press, 2006.
[3] Robert M. Beals, Joseph L. Wetherell, and Michael E. Zieve. Polynomials with a common composite. *Israel Journal of Mathematics*, 174(1):93–117, 2009.
[4] Thomas Becker, Volker Weispfenning, and Heinz Kredel. *Gröbner Bases.* Springer, 1993.
[5] Dale Beihoffer, Jemimah Hendry, Albert Nijenhuis, and Stan Wagon. Faster algorithms for Frobenius numbers. *Electronic Journal of Combinatorics*, 12:R27, 2005.
[6] Yuri F. Bilu. Quadratic factors of $f(x) - g(y)$. *Acta Arithmetica*, 90(4):341–355, 1999.
[7] Yuri F. Bilu and Robert Tichy. The diophantine equation $f(x) = g(y)$. *Acta Arithmetica*, 95(3):261–288, 2000.
[8] Franz Binder. Fast computations in the lattice of polynomial rational function fields. In *Proceedings of ISSAC'96*, pages 43–48, 1996.
[9] Mireille Bousquet-Mélou. An elementary solution of Gessel's walks in the quadrant. *Advances in Mathematics*, 303:1171–1189, 2016.
[10] J. W. S. Cassels. Factorization of polynomials in several variables. In K. E. Aubert and W. Ljunggren, editors, *Proceedings of the 15th Scandinavian Congress Oslo 1968*, pages 1–17. Springer Berlin Heidelberg, 1969.
[11] Pierrette Cassou-Noguès and Jean-Marc Couveignes. Factorisations explicites de $g(y) - h(z)$. *Acta Arithmetica*, 87(4):291–317, 1999.
[12] H. Davenport, D. J. Lewis, and A. Schinzel. Equations of the form $f(x) = g(y)$. *The Quarterly Journal of Mathematics*, 12(1):304–312, 1961.
[13] H. T. Engstrom. Polynomial substitutions. *American Journal of Mathematics*, 63(2):249–255, 1941.
[14] Michael D. Fried. The field of definition of function fields and a problem in the reducibility of polynomials in two variables. *Illinois Journal of Mathematics*, pages 128–146, 1973.
[15] Michael D. Fried and R. E. MacRae. On curves with separated variables. *Mathematische Annalen*, 180:220–226, 1969.
[16] Pinaki Mondal. When is the intersection of two finitely generated subalgebras of a polynomial ring also finitely generated? *Arnold Mathematical Journal*, 3(3):333–350, 2017.
[17] R. W. Owens. An algorithm to solve the Frobenius problem. *Mathematics Magazine*, 76(4):264–275, 2003.
[18] J. Barkley Rosser and Lowell Schoenfeld. Approximate formulas for some functions of prime numbers. *Illinois Journal of Mathematics*, 6(1):64–94, 1962.
[19] Josef Schicho. A note on a theorem of Fried and MacRae. *Archiv der Mathematik*, 65:239–243, 1995.

Robots, Computer Algebra and Eight Connected Components

Jose Capco
Innsbruck University
Innsbruck, Austria
jose.capco@uibk.ac.at

Mohab Safey El Din
Sorbonne Université, CNRS, LIP6
F-75252, Paris Cedex 05, France
mohab.safey@lip6.fr

Josef Schicho
JKU University
Linz, Austria
Josef.Schicho@risc.jku.at

ABSTRACT

Answering connectivity queries in semi-algebraic sets is a long-standing and challenging computational issue with applications in robotics, in particular for the analysis of kinematic singularities. One task there is to compute the number of connected components of the complementary of the singularities of the kinematic map. Another task is to design a continuous path joining two given points lying in the same connected component of such a set. In this paper, we push forward the current capabilities of computer algebra to obtain computer-aided proofs of the analysis of the kinematic singularities of various robots used in industry.

We first show how to combine mathematical reasoning with easy symbolic computations to study the kinematic singularities of an infinite family (depending on paramaters) modelled by the UR-series produced by the company "Universal Robots". Next, we compute roadmaps (which are curves used to answer connectivity queries) for this family of robots. We design an algorithm for "solving" positive dimensional polynomial system depending on parameters. The meaning of solving here means partitioning the parameter's space into semi-algebraic components over which the number of connected components of the semi-algebraic set defined by the input system is invariant. Practical experiments confirm our computer-aided proof and show that such an algorithm can already be used to analyze the kinematic singularities of the UR-series family. The number of connected components of the complementary of the kinematic singularities of generic robots in this family is 8.

KEYWORDS

kinematic singularity avoidance, roadmaps, semialgebraic sets

ACM Reference Format:
Jose Capco, Mohab Safey El Din, and Josef Schicho. 2020. Robots, Computer Algebra and Eight Connected Components. In *International Symposium on Symbolic and Algebraic Computation (ISSAC '20), July 20–23, 2020, Kalamata, Greece.* ACM, New York, NY, USA, 8 pages. https://doi.org/10.1145/3373207.3404048

The three authors are supported by the joint ANR-FWF ANR-19-CE48-0015, FWF I 4452-N ECARP project. Mohab Safey El Din is supported by the ANR grants ANR-18-CE33-0011 Sesame and ANR-19-CE40-0018 De Rerum Natura, the PGMO grant CAMiSAdo and the European Union's Horizon 2020 research and innovation programme under the Marie Sklodowska-Curie grant agreement N. 813211 (POEMA). .

ISSAC '20, July 20–23, 2020, Kalamata, Greece

ACM ISBN 978-1-4503-7100-1/20/07…$15.00
https://doi.org/10.1145/3373207.3404048

1 INTRODUCTION

The individual parts of a serial robot, called links, are moved by controlling the angle of each joint connecting two links. The inverse kinematics problem asks for the values of all angles producing a desired position of the end effector, where "position" includes not just the location of the end effector in 3-space but also the orientation. In a sense, this means inverting a function which is called the forward kinematics map in robotics, which determines the position of the end effector for given angles by a well-known formula (see Section 2). In robot controlling, the inverse kinematics problem is often solved incrementally: starting from some known initial angle configuration and its corresponding end effector position, we want to compute the change of the angles required to achieve a desired small change in the end effector position.

Kinematic singularities are defined as critical points of the forward map, i.e. angle configurations where the Jacobian matrix of the forward map is rank deficient. There are two known facts that make rather difficulty to control a robot in a singular or near a singular configuration. First, if an end effector velocity or force outside the image of the singular Jacobian is desired, then the necessary joint velocity or torque is either not defined or very large (see [23] §4.3 and [31] §5.9). The second reason is that industrial controllers are based on Newton's method for the incremental solution of the inverse problem, and this method is not guaranteed to converge if it is used with a starting point close to the singular set. For these reasons, engineers prefer to plan the robot movements avoiding kinematic singularities.

For a general serial robot with six joints, the singular set is a hypersurface defined locally by the Jacobian determinant of the forward map. Its complementary, a real manifold, is not connected. Counting the number of connected components of this manifold and answering connectivity queries in this set is then of crucial importance in this application domain.

Answering connectivity queries in semi-algebraic sets is a classical problem of algorithmic semi-algebraic geometry which has attracted a lot of attention through the development of the so-called ROADMAP algorithms (see e.g. [6, 8, 9, 11, 12, 21, 27, 28]). Up to our knowledge, such algorithms had never been developed enough and implemented efficiently to tackle real-life applications.

In this paper, we push forward the capabilities of computer algebra in this application domain by solving connectivity queries for the non singular configuration sets of industrial robots from the UR series of the company "Universal Robots". For a particular robot in this series, the UR5, the number of components of the non singular configuration set is 8 (see Section 3). For two points in the same component, we show how to construct a connecting path, in two ways: either by an ad hoc way (which has its own algorithmic interest) taking advantage of the specialty of the geometric parameters of UR5, and by using the ROADMAP algorithm (see Section 5). Next,

we go further and extend our analysis of UR5 to the whole UR series and prove that outside a proper Zariski closed set (UR5 is outside this closed set) the number of connected components of the non singular configuration set is constant. These are computer-aided mathematical proofs involving «easy» symbolic computations
The next contribution is based on the fact that the family of UR robots is determined by a finite list of real parameters. Hence, an algorithmic way of tackling the problem of analyzing kinematic singularities of the whole UR family is to «solve» a *positive dimensional* system *depending on parameters* (i.e. after specialization of the parameters, the specialized system is positive dimensional). We design an algorithm that decomposes the parameter's space into semi-algebraic subsets, such that the number of connected components of the non singular configurations is constant in each of these subsets. As far as we know, this is the first algorithm of that type which is designed.
We also implemented this algorithm and used it for the analysis of the kinematic singularities of the UR series. Computations are heavy but already doable (on a standard laptop) within 10 hours. This is a computational way to retrieve the same results as our computer-aided mathematical proofs. These computations show that computer algebra today is efficient enough to solve connectivity queries that are of practical interest in industrial robot applications.

2 ROBOTICS PROBLEM FORMULATION

We define a *manipulator* or *robot* as follows: we have finite ordered rigid bodies called *links* which are connected by n revolute *joints* that are also ordered. To each joint we associate a coordinate system or a *frame*. The links are connected in a serial manner i.e. if we consider the robot as a graph such that the vertices are joints and the edges are links then this graph is a path (the first and last joint has degree 1 and all other joints have degree 2) and the joints allow rotation about its axes, so that if a joint rotates then all other subsequent links rotate about the axes of this joint. A reference coordinate system is chosen for the final joint which is called the *end-effector*[1]. See Fig. 1 for an illustration.

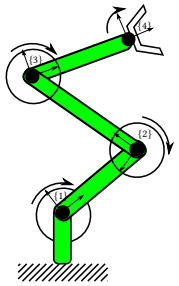


Figure 1: Green objects are links. Joints 1,2,3,4 and their corresponding frames are shown.

In theoretical kinematics one may forget that the links are rigid bodies so that *collision* between links are disregarded. In this case we may as well think of a robot as a differentiable map $F : \mathrm{SO}(2)^n \longrightarrow \mathrm{SE}(3)$ where SO(2) is the one-dimensional group of rotations around a fixed line, parameterised by the rotation angle, and SE(3) is the six-dimensional group of Euclidean congruence transformations. This map is defined in the following way:

- The i-th coordinate of an element in $\mathrm{SO}(2)^n$ is associated to the i-th (revolute) joint parameter.
- For joint values $\vec{\theta} := (\theta_1, \ldots, \theta_n)$ in $\mathrm{SO}(2)^n$, the image $F(\vec{\theta})$ is the transformation of the end-effector from the initial position corresponding to all angles being zero to the final position obtained by composing the n rotations.

The map F itself is called the *kinematic map* (of the robot). Its domain is called the *configuration space*, while its image is called the *work-space* or the *kinematic image*.
We use the Denavit-Hartenberg (DH) convention when describing relations between two joint frames. It is standard in robotics ; its advantages are discussed in e.g. [31, §3.2], [1, §4.2]. The transformation between the frames is given by the following rule:

- The z-axis of the reference frame will be the axis of rotation of the joint.
- To obtain the next frame, one starts with a rotation about the z-axis of the reference frame, called the *rotation*, followed by
- a translation along the z-axis of the reference frame, called the *offset*, followed by
- a translation along the x-axis, called the *distance*, followed by
- a rotation about the x-axis, called the *twist*.

The transformation between frame i to frame $i+1$ is

$$R_z(\theta_i)T_z(d_i)T_x(a_i)R_x(\alpha_i)$$

where R_z, T_z, T_x, R_x are rotations or translations with respect to z- or x-axis parameterised by the angle of rotation θ_i (the i-th joint parameter), the offset d_i, the distance a_i and the angle of twist α_i of the i-th frame. For a given robot with n joints all DH parameters except for the rotation are fixed values. So that image of F for given joint values (the rotations) $(\theta_1, \ldots, \theta_n)$ is just the multiplication of these transformations in SE(3). The parameters d_1, d_n, a_n, α_n are assumed to be 0. This is not a loss of generality, because we can freely choose the frame at the base and at the end-effector. More detailed discussion on these can be seen in [31].

Example 2.1. *The UR5 robot has the following DH parameters:*
distances (m.) $(a_1, \ldots, a_6) := (0, -\frac{425}{1000}, -\frac{39225}{100000}, 0, 0, 0)$
offsets (m.) $(d_1, \ldots, d_6) := (0, 0, 0, \frac{10915}{100000}, \frac{9465}{100000}, 0)$
twist angles (rad.) $(\alpha_1, \ldots, \alpha_6) := (\frac{\pi}{2}, 0, 0, \frac{\pi}{2}, -\frac{\pi}{2}, 0)$
For example, the following joint angles (rotations, in rad.)

$$(\theta_1, \ldots, \theta_6) := \left(\frac{1}{10}, \frac{2}{10}, \frac{3}{10}, \frac{4}{10}, \frac{5}{10}, \frac{6}{10}\right)$$

leads to the following transformation in $(R, \vec{t}) \in \mathrm{SE}(3)$ *(represented as elements in* $\mathrm{SO}(3) \rtimes \mathbb{R}^3$*) where:*

$$R \simeq \begin{pmatrix} 0.047 & -0.977 & -0.209 \\ -0.393 & 0.174 & -0.903 \\ 0.918 & 0.123 & -0.376 \end{pmatrix}, \ \vec{t} \simeq (-6.768, -1.7784, -3.336).$$

Definition. *Given the kinematic map of a manipulator* $F : \mathrm{SO}(2)^n \to \mathrm{SE}(3)$*, the* kinematic singularities *in the configuration space are the points* $P \in (\mathbb{P}^1)^n$ *such that the Jacobian of* F *at* P *is rank-deficient.*

In this paper, we will only deal with 6-jointed manipulators. Therefore the kinematic map is a differentiable map from the 6-dimensional configuration space $(\mathrm{SO}(2))^6$ to the group SE(3), which

[1] this is usually another frame, but this is just an additional fixed transformation in SE(3) and w.l.o.g. we assume that the final offset, distance and twist is 0

is also 6-dimensional. For non-singular points of the map, the Jacobian is therefore invertible, and F is a local homeomorphism. Here is a well-known geometric description of singularities.

Theorem 2.2. *Let $F : \mathrm{SO}(2)^6 \to \mathrm{SE}(3)$ be the kinematic map of a robot with 6 joints. Let $P \in \mathrm{SO}(2)^6$. Then the following are equivalent.*

(1) P is a kinematic singularity.
(2) The Jacobian of F at P is singular.
(3) If $P_1, \ldots, P_6 \in \mathbb{P}^5(\mathbb{R})$ are the Plücker representation of the axes (lines in $\mathbb{P}^3$) of the joints of the robot at the configuration point P then the matrix consisting of the Plücker coordinates $(p_{i,j})_{i,j\leq 6}$ ($P_i = (p_{i,1} : p_{i,2} : \cdots : p_{i,6})$ for $i = 1, \ldots, 6$) is singular.

Proof. The equivalence of the first two items is clear by definition. The equivalence of the first and the third item is found in [30, §4.5.1], [23, §4.1] or [1, §4.5.1.] □

Assume that we have two non-singular points in the configuration set. As explained earlier, we want to decide if these two configurations can be connected by a curve of configurations which avoids the singular hypersurface (see [33] §1.2 for some history on this question). If yes, then an explicit construction of such a curve is also of interest. In order to tackle these problems, we choose parameters for SO(2) so that the equation of the hypersurface becomes a polynomial. This is not the case when we use the angles $\theta_1, \ldots, \theta_n$, because the Jacobian contains trigonometric functions in these angles. One well-known strategy is to parametrize by points on a unit circle, i.e. by two parameters satisfying the equation of the unit circle. This has a clear disadvantage: the number of variables increases, and the singular set has co-dimension greater than one. Another well-known strategy is to replace θ_i by $v_i = \tan\frac{\theta_i}{2}$. The variable v_i ranges over the projective line, and the angle π corresponds to the point at infinity. If we set $v_i = \tan\frac{\theta_i}{2}$ for $i = 1, \ldots, n$, then we obtain, in general, a polynomial in $v_2, \ldots, v_5$. More precisely, the degree is 2 in v_2 and v_5 and degree 4 in v_3 and in v_4. The Jacobian does not depend on the joint angles θ_1 and θ_6. This is clear from the third characterization of singularities in Theorem 2.2: only the position of the axes are relevant, and a rotation along the first or the last axis does not change the position of any axis.

We define the *UR Family* to be robots having a similar DH-parameter as the known UR robots (UR5, UR10, etc.). Such *UR robots* are parameterised by the following DH parameters

distances (m.)	$(a_1, \ldots, a_6) := (0, a_2, a_3, 0, 0, 0)$
offsets (m.)	$(d_1, \ldots, d_6) := (0, 0, 0, d_4, d_5, 0)$
twist angles (rad.)	$(\alpha_1, \ldots, \alpha_6) := (\frac{\pi}{2}, 0, 0, \frac{\pi}{2}, -\frac{\pi}{2}, 0)$

For these robots, the determinant of the Jacobian (see [34]), expressed as a polynomial in $v_2, \ldots, v_5$, is $A = -Bv_3v_5$ with

$$\begin{aligned} B = {} & a_2v_2^2v_3^2v_4^2 - a_3v_2^2v_3^2v_4^2 - 2d_5v_2^2v_3^2v_4 - 2d_5v_2^2v_3v_4^2 - 2d_5v_2v_3^2v_4^2 \\ & + a_2v_2^2v_3^2 + a_2v_2^2v_4^2 - a_2v_3^2v_4^2 - a_3v_2^2v_3^2 + a_3v_2^2v_4^2 + 4a_3v_2v_3v_4^2 \\ & + a_3v_3^2v_4^2 + 2d_5v_2^2v_3 + 2d_5v_2^2v_4 + 2d_5v_2v_3^2 + 8d_5v_2v_3v_4 \\ & + 2d_5v_2v_4^2 + 2d_5v_3^2v_4 + 2d_5v_3v_4^2 + a_2v_2^2 - a_2v_3^2 - a_2v_4^2 + a_3v_2^2 \\ & + 4a_3v_2v_3 + a_3v_3^2 - a_3v_4^2 - 2d_5v_2 - 2d_5v_3 - 2d_5v_4 - a_2 - a_3 \end{aligned}$$

Note that there is a degree drop in three of the four cases: the degree in v_3 is only 3, and not 4, and the degree in v_4 is only 2, and not 4, and the degree in v_5 is only 1, and not 2. The drop in the degree means that the homogeneous form of the Jacobian has a linear factor that vanishes if and only if the value of the variable whose degree drops is infinity, or equivalently, that the corresponding angle is π. Since we are interested in the complement of the singular space, we may assume that none of these three angles is equal to π, and we can use the parameters v_3, v_4, v_5 without worrying about paths crossing infinity.

For the angle θ_2, the situation is different. There is no degree drop, hence there are configurations with $\theta_2 = \pi$ in the non singular configuration set. If we use the parametrization by half angle, then we have to take paths in the projective line into account that cross infinity, or, in other words, consider this variable in the projective space $\mathbb{P}^1(\mathbb{R})$. But, to take advantage on algorithms acting on semi-algebraic sets, one needs variables that range over $\mathbb{R}$.

Hence, we instead use parameters $s_2 = \sin(\theta_2)$ and $c_2 = \cos(\theta_2)$ and add the additional equation $s_2^2 + c_2^2 = 1$, obtaining a polynomial in the variables s_2, c_2, v_3, v_4, v_5 with coefficients depending on the parameters a_2, a_3, d_4, d_5. Of course, the reason for this more costly treatment for θ_2 is just necessary if we use the ROADMAP algorithm subsequently. For an alternative analysis not using it, it is still better to use the half tangent ranging over the projective line.

3 ANALYSIS OF THE UR5 ROBOT

We gave in Example 2.1 the Denavit-Hartenberg parameters of the UR5 robot. These values are used to instantiate a_2, a_3 and d_5 in the above polynomial B; the specialized polynomial is then denoted by $\tilde{B}$ and we let $\tilde{A} = \tilde{B}v_3v_5$. Recall that v_2 ranges over $\mathbb{P}^1$, while v_3, v_4, v_5 range only over the affine line.

We investigate the discriminant of $\tilde{B}$ with respect to the variable v_2 (thus the projection of the critical set to the (v_3, v_4)-plane). The discriminant of $\tilde{B}$ with respect to the variable v_2 we denote as $b \in \mathbb{R}[v_3, v_4]$. This discriminant is still factorisable in $\mathbb{C}[v_3, v_4]$. In fact, one checks, that it is the factor of two complex conjugates of some polynomial in $\mathbb{R}[v_3, v_4]$. This implies that $b = c^2 + d^2$ is the sum of two squares of real polynomials $c, d \in \mathbb{R}[v_3, v_4]$. These two polynomials are given by

$$c = \frac{1577212v_3 - 3561263v_4 - 14850585}{\sqrt{2006237}}$$

$$d = \frac{(\sqrt{2006237}v_4 + 1239915 - 7144712)v_3 + 16090500v_4}{\sqrt{2006237}}$$

Thus, b can have only two real roots (i.e. two pairs (v_3, v_4)), i.e the vanishing set of b in $\mathbb{R}^2$ is finite, namely they points that are the zeros of both c and d. We solve this as floating numbers to have an idea of their vicinity in an affine chart of the ambient space of the kinematic singularity. The roots are

$$\begin{aligned} q_1 &= (v_3 \simeq -9.140975564 \quad , \quad v_4 \simeq -8.218388067) \\ q_2 &= (v_3 \simeq 9.140975563 \quad , \quad v_4 \simeq -.1216783622) \end{aligned}$$

For the two special values q_1 and q_2 in the $(v3, v4)$-plane, all three coefficients of $\tilde{B}$ with respect to v_2 are zero.

Now, since the discriminant b is positive except at these two points and since $\tilde{B}$ itself is quadratic with respect to v_2 we conclude that the preimage of the projection (to (v_3, v_4)-plane) are two real points in the variety defined by $\tilde{B}$. Thus, the variety defined by $\tilde{B}$ is composed of two sheets (above any two points (v_3, v_4) except q_1 and q_2). Let X be the complement of the vanishing points of $\tilde{B}$ in $\mathbb{P}^1 \times \mathbb{A}^2$. Set

$$Y := \mathbb{A}^2 \setminus (\{q_1, q_2\} \cup \{(0, v_4) \mid v_4 \in \mathbb{R}\})$$

So we have a canonical projection (to the (v_3, v_4)-plane) from $X \cap (\mathbb{P}^1 \times Y)$ to Y. The fiber of this projection is a projective line

without two distinct points. Hence, every fiber has two components. The sign of $\tilde{B}$ is different for the two components of each fiber. Then, we have two components of X for each component of Y. Obviously, Y has two components, hence we have a total number of 4 components.

For the non singular set, which is the complement of the zero set of $\tilde{A}$, we get 8 components: for each component of X, we have one component where v_5 is positive and one where v_5 is negative.

Now assume that we have two non singular configuration points x, y in the same component, and we want to construct a path connecting them. The projections to Y have to lie in the same component of Y, and because Y is the plane without a line and two points, it is easy to connect the images of the projections in Y: in most cases, a straight line segment is fine; if the straight line segment connecting the two image points contains q_1 or q_2, we have to do a random detour via a third point. The zero set of $\tilde{B}$ is a two-sheeted covering of Y. So, for any value of Y, we have two points in the zero set of $\tilde{B}$ projecting to it. If we look at these points as points in SO(2), then it is clear that there are two "midpoints" in the zero set of $\tilde{B}$, which have equal angle distance to these two points. The value of $\tilde{B}$ is positive for one of the two midpoints and negative for the other one. The sign of $\tilde{B}(x)$ and $\tilde{B}(y)$, however, must be the same because the two points are in the same connected component. Suppose, without loss of generality, that $\tilde{B}(x)$ and $\tilde{B}(y)$ are both positive. Then we first connect x to the midpoint over the projection of x to Y with positive sign, by a curve in the fiber. Next, we lift the path in Y, connecting the projections of x and y in the same component, to a path of midpoints with positive sign, arriving at the midpoint with positive sign lying over the projection of y. Finally, we connect this midpoint to y by the other fiber.

Below, we show the sheets in Figure 2 to illustrate that:

(i) the regions above and below the sheets can be connected
(ii) the region between the two sheets is the other connected component
(iii) the two points q_2 and q_3 are points in the projection where the sheets get connected (see assymptotes in Fig. 2). Thus, the variety describing the two sheets is connected.

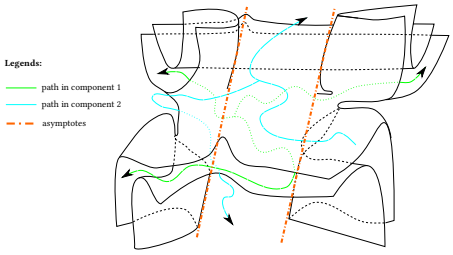


Figure 2: The two sheets of $\tilde{B}$

4 UR SERIES

We can make a general statement for robots belonging to the UR family (e.g. UR10, UR3 etc.). We define the *UR Family* to be robots which have a similar DH-parameter as the known UR robots (UR5, UR10 etc.), a robot in this family we shall call a *UR robot*. Namely they are parameterised by the following DH parameters :

distances (m.) $(a_1, \ldots, a_6) := (0, a_2, a_3, 0, 0, 0)$
offsets (m.) $(d_1, \ldots, d_6) := (0, 0, 0, d_4, d_5, 0)$
twist angles (rad.) $(\alpha_1, \ldots, \alpha_6) := (\frac{\pi}{2}, 0, 0, \frac{\pi}{2}, -\frac{\pi}{2}, 0)$

i.e. these robots are parameterised by 4 parameters: a_2, a_3, d_4, d_5.

We can write the largest (in number of terms and in degree) polynomial factor of the polynomial whose vanishing points is the kinematic singularity in configuration space of a UR robot as

$$\begin{aligned} B = {} & a_2v_2^2v_3^2v_4^2 - a_3v_2^2v_3^2v_4^2 - 2d_5v_2^2v_3^2v_4 - 2d_5v_2^2v_3v_4^2 - 2d_5v_2v_3^2v_4^2 \\ & + a_2v_2^2v_3^2 + a_2v_2^2v_4^2 - a_2v_3^2v_4^2 - a_3v_2^2v_3^2 + a_3v_2^2v_4^2 + 4a_3v_2v_3v_4^2 \\ & + a_3v_3^2v_4^2 + 2d_5v_2^2v_3 + 2d_5v_2^2v_4 + 2d_5v_2v_3^2 + 8d_5v_2v_3v_4 \\ & + 2d_5v_2v_4^2 + 2d_5v_3^2v_4 + 2d_5v_3v_4^2 + a_2v_2^2 - a_2v_3^2 - a_2v_4^2 + a_3v_2^2 \\ & + 4a_3v_2v_3 + a_3v_3^2 - a_3v_4^2 - 2d_5v_2 - 2d_5v_3 - 2d_5v_4 - a_2 - a_3 \end{aligned}$$

Note that d_4 does not affect the singularity of the robot. Taking the discriminant of B with respect to v_2 yields the sum of two squares i.e. the product of two quadratic complex conjugate polynomials $\mathrm{disc}(B, v_2) = g\overline{g}$.

$$\begin{aligned} g = {} & (-a_2v_3v_4 + a_3v_3v_4 + d_5v_3 + d_5v_4 + a_2 + a_3) \\ & + (-d_5v_3v_4 + a_2v_3 + a_2v_4 - a_3v_3 + a_3v_4 + d_5)i \end{aligned}$$

For a robot determined by some real quadruple $u \in \mathbb{R}^4$, let A_u, B_u, g_u be the polynomials obtained by instantiating in A, B, g the variables a_2, a_3, d_4, d_5 by the corresponding real values in the quadruple. Let $f : \mathbb{R}^3 \to \mathbb{R}^2$ be the projection $(v_2, v_3, v_5) \mapsto (v_3, v_5)$. Let $Y_u \subset \mathbb{R}^2$ be the complement of (the union of the line $v_3 = 0$ and the common zero set of $\mathrm{Re}\, g_u$ and $\mathrm{Im}\, g_u$). Then the real zero set of B_u in $\mathbb{R}^3$ intersected with $f^{-1}(Y_u)$ projects to surjectively to Y_u, in such a way that there are two sheets, each projecting homeomorphically to Y_u.

For general robot u, the real set of g, which is meaning the set of all points in the real (v_3, v_4)-plane such that both the real part and the imaginary part of g is equal to zero, is a finite subset of $\mathbb{R}^2$. All arguments from the previous sections work in this case as well. Hence we get 8 components for these parameters'values. Moreover, we have paths connecting points in the same component, as in the previous section.

It remains to treat the non-general robots where the real zero set of g is one-dimensional. This is the case if and only if $d_5 = a_2^2 - a_3^2 = 0$. The even more special case $d_5 = a_2 = a_3 = 0$ is easy to analyze: here, the determinant of the Jacobian A is identically zero, which means that there are no non singular configurations. Excluding that case, we have two families of robots, and in each family, up to the value of d_4, the parameters are unique up to scaling. Without loss of generality, we can reduce to exactly two non-general robots $u' = (1, 1, 0, 0)$ and $u'' = (1, -1, 0, 0)$. Then the polynomial $B_{u'}$ has a factor is $C' := v_3v_4 - v_3 - v_4 - 1$, and the polynomial $B_{u''}$ has a factor $C'' := v_3v_4 + v_3 + v_4 - 1$. Apart from that complication, the analysis proceeds similar as in the general case: the set $Y_{u'}$ is the plane minus the line $v_3 = 0$ minus the hyperbola with equation C', and the set $Y_{u''}$ is the plane minus the hyperbola with equation C''. In both cases, the number of components of Y is 5, as it can be seen in Figure 3. Consequently, we have 20 components in total. The paths between points in the same component can be constructed similarly as in the general case.

5 CONNECTIVITY AND ROADMAPS

We explain the ROADMAP algorithm for the special case where the semi-algebraic set S is given as a subset of some vector space $\mathbb{R}^N$,

$N \in \mathbb{N}$, defined by an equation $f(x_1, \ldots, x_N) = 0$ and an inequation $g(x_1, \ldots, x_N) \neq 0$. We assume that the algebraic set defined by $f = 0$ is smooth. This is sufficient for our application: the inequation is the determinant of the Jacobian of the kinematic map A, and the equality is $s_2^2 + c_2^2 - 1 = 0$.

One first reduces the problem to one where the semi-algebraic set we consider is bounded. Note that there exists $R > 0$ large enough such that the connected components of S are in one-to-one correspondence with the intersection of S with the hyper-ball defined by $\mathscr{N}_R \leq 0$ where $\mathscr{N}_R = x_1^2 + \cdots + x_n^2 - R$. We denote this intersection by S'. Note that a roadmap of S' provides a roadmap of S.

Determining such a large enough real number R is done by choosing it larger than the largest critical value of the restriction of the map $\boldsymbol{x} \to \|\boldsymbol{x}\|^2$ to each regular strata of the the Euclidean closure of S. This leads us to compute critical values of that map restricted to the hypersurface defined by $f = 0$ and next take the limits of the critical values of the sets defined by $g = \pm\varepsilon$ and $f = g \pm \varepsilon = 0$ when $\varepsilon \to 0$.

Next, we compute the critical values $\eta_1 < \cdots < \eta_s$ of the restriction of the map $\boldsymbol{x} \to g(\boldsymbol{x})$ to the semi-algebraic set defined by $f = 0$ and $\mathscr{N}_R \leq 0$. Following Thom's isotopy lemma [13], when e is chosen between 0 and $\min(|\eta_i|, 1 \leq i \leq s)$, the connected components of the semi-algebraic set S_e^+ (resp. S_e^-) defined by $\mathscr{N}_R \leq 0, f = g - e = 0$ (resp. $\mathscr{N}_R \leq 0, f = g + e = 0$) are in one-to-one correspondence with the connected components of the semi-algebraic set defined by $\mathscr{N}_R \leq 0, f = 0, g > 0$ (resp. $\mathscr{N}_R \leq 0, f = 0, g > 0$). Besides, $S_e^+ \subset S'$ (resp. $S_e^- \subset S'$). Then a roadmap of S' is obtained by taking the union of a roadmap of S_e^+ with the roadmap of S_e^-. Hence, we have performed a reduction to computing roadmaps in the compact semi-algebraic sets S_e^+ and S_e^-.

In our application, the algebraic sets defined by the vanishing of all subsets of the defining polynomials of S_e^+ and S_e^- are smooth. Hence, we can rely on a slight modification of the roadmap algorithm given in [12] where we replace computations with multivariate resultants for solving polynomial systems by computations of Gröbner bases. The algorithm in [12] then takes as input a polynomial system defining a closed and bounded semi-algebraic set S and proceeds as follows. The core idea is to start by computing a curve $\mathscr{C}$ which has a non-empty intersection with each connected component of S. That curve will be typically the critical locus on the (x_1, x_2)-plane when one is in generic coordinates (else, one just needs to change linearly generically the coordinate system). A few remarks are in order here. When S is defined by $f_1 = \cdots = f_p = 0$ and $g_1 \geq 0, \cdots, g_s \geq 0$, to define the critical locus of the projection on the (x_1, x_2)-plane restricted to S one takes the union of the critical loci of that projection restricted to the real algebraic sets defined for all $\{i_1, \ldots, i_\ell\} \subset \{1, \ldots, s\}$, by $f_1 = \cdots = f_p = g_{i_1} = \cdots = g_{i_\ell} = 0$ and intersect this union of critical loci with S (see [12]).

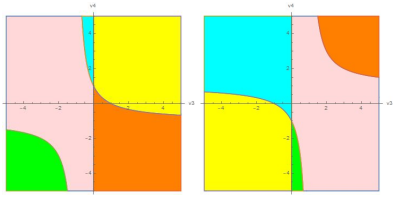


Figure 3: Left (resp. right) shows the components of $v_3(v_3v_4 + v_3 + v_4 - 1) \neq 0$ (resp. $v_3(v_3v_4 - v_3 - v_4 - 1) \neq 0$) in Y

That way, one obtains curves that intersect all connected components of S but these intersections may not be connected. To repair these connectivity failures, Canny's algorithm finds appropriate slices of S. Let π_1 be the canonical projection $(x_1, \ldots, x_n) \to x_1$. This basically consists in finding $\alpha_1 < \ldots < \alpha_k$ in $\mathbb{R}$ such that the union of $\cup_{i=1}^k S \cap \pi_1^{-1}(\alpha_i)$ with the critical curve $\mathscr{C}$ has a non-empty and connected intersection with each connected component of S. The way Canny proposes to find those α_i's is to compute the critical values of the restriction of π_1 to $\mathscr{C}$. By the algebraic Sard's theorem (see e.g. [28, Appendix B]), these values are in finite number and Canny proposes to take $\alpha_1, \cdots, \alpha_k$ as those critical values. This leads to compute with real algebraic numbers which can be encoded with their minimal polynomials and isolating intervals. Since these minimal polynomials may have large degrees (singly exponential in n), that step can be prohibitive for practical computations. We use then the technique introduced in [22] which consists in replacing $\alpha_1 < \cdots < \alpha_k$ with rational numbers $\rho_1 < \cdots < \rho_{k-1}$ with $\alpha_i < \rho_i < \alpha_{i+1}$. We refer to [22] for the rationale justifying this trick. All in all, one obtains a recursive algorithm with a decreasing number of variables at each recursive call. Combined with efficient Gröbner bases engines, we illustrate in Section 7 that the ROADMAP algorithm (with the modifications introduced above) can be used in practice to answer connectivity queries in semi-algebraic sets in concrete applications.

The concept of roadmap and the algorithm computing it, described above, may seem cumbersome and unnecessarily sophisticated, especially when compared with the much more direct CAD approach [29]. The CAD algorithm is also a recursive algorithm, producing its recursive instance by projecting the hypersurface to $\mathbb{R}^{n+1}$ and analyzing the discriminant. This leads to an iteration of discriminants, and it is easy to see that the degree of the iterated discriminants grows double exponentially in n: roughly, the degree of the discriminant is squared in every iteration. There lies the motivation for all the sophistication of the ROADMAP algorithm: for each instance in the all recursive calls, the degree of the input polynomial is exactly the same as the degree of the initially given polynomial f. This leads to an asymptotic complexity which is only single exponential in n^2. We refer to [8, 9, 27, 28] for more recent algorithms improving the complexity of roadmap computations.

6 PARAMETRIC POLYNOMIAL SYSTEMS

Let $F = (f_1, \ldots, f_p)$ and $G = (g_1, \ldots, g_q)$ in $\mathbb{Q}[\boldsymbol{x}, \boldsymbol{y}]$ with $\boldsymbol{x} = (x_1, \ldots, x_n)$ and $\boldsymbol{y} = (y_1, \ldots, y_t)$. We consider further $\boldsymbol{y}$ as a sequence of parameters and the polynomial system

$$f_1 = \cdots = f_p = 0, \qquad g_1 \ \sigma_1 \ 0, \ \ldots, \ g_q \ \sigma_q \ 0$$

with $\sigma_i \in \{>, \geq\}$. We let $S \subset \mathbb{R}^n \times \mathbb{R}^t$ be the semi-algebraic set defined by this system. For $y \in \mathbb{R}^t$, we denote by F_y and G_y the sequences of polynomials obtained after instantiating $\boldsymbol{y}$ to y in F and G respectively. Also, we denote by $S_y \subset \mathbb{R}^n$ the semi-algebraic set defined by the above system when $\boldsymbol{y}$ is specialized to y. The algebraic set defined by the simultaneous vanishing of the entries of F (resp. F_y) is denoted by $V(F) \subset \mathbb{C}^{n+t}$ (resp. $V(F_y) \subset \mathbb{C}^n$).

We describe an algorithm for solving such a parametric polynomial system *without* assuming that for a *generic* point y in $\mathbb{C}^t$, $V(F_y)$ is finite. In that situation, solving such a parametric polynomial system may consist in partitioning the parameters'space $\mathbb{R}^t$ into semi-algebraic sets $T_1, \ldots, T_r$ such that, for $1 \leq i \leq r$, the number of connected components of S_y is invariant for any choice of y in T_i. We prove below that such an algorithmic problem makes sense.

Proposition 6.1. *Let $S \subset \mathbb{R}^n \times \mathbb{R}^t$ be a semi-algebraic set and π be the canonical projection*

$$(x_1, \ldots, x_n, y_1, \ldots, y_t) \to (y_1, \ldots, y_t).$$

There exist semi-algebraic sets $T_1, \ldots, T_r$ in $\mathbb{R}^t$ such that

- $\mathbb{R}^t = T_1 \cup \cdots \cup T_r$,
- *there exists $b_i \in \mathbb{N}$ such that for any $y \in T_i$, the number of connected components of S_y is b_i.*

Proof. Observe that the restriction of π to S is semi-algebraically continuous. From Hardt's semi-algebraic triviality theorem [10, Theorem 9.3.2], there exists a finite partition of $\mathbb{R}^t$ into semi-algebraic sets $T_1, \ldots, T_r$ and for each $1 \leq i \leq r$, a trivialization $\vartheta_i : T_i \times E_i \to \pi^{-1}(T_i) \cap S$ (where E_i is a fiber $\pi^{-1}(y) \cap S$ for some $y \in T_i$). Fix i and choose an arbitrary point $y' \in T_i$. Observe that we are done once we have proved that $\pi^{-1}(y') \cap S$ and E_i have the same number of connected components. Recall that, by definition of a trivialization (see [10, Definition 9.3.1]), $\theta_i : T_i \times E_i \to \pi^{-1}(T_i) \cap S$ is a semi-algebraic homeomorphism and for any $(y', x) \in T_i \times E_i$, $\pi \circ \theta_i(y', x) = y'$. Hence, we deduce that E_i is homeomorphic $\pi^{-1}(y') \cap S$. As a consequence, they both have the same number of connected components. □

Instead of computing a partition of the parameters'space into semi-algebraic sets $T_1, \ldots, T_r$ as above, one will consider non-empty *disjoint* open semi-algebraic sets $U_1, \ldots, U_\ell$ in $\mathbb{R}^t$ such that the complement of $U_1 \cup \cdots \cup U_\ell$ in $\mathbb{R}^t$ is a semi-algebraic set of dimension less than t and such that for $1 \leq i \leq t$, there exists $b_i \in \mathbb{N}$ such that b_i is the number of connected components of S_y for any $y \in U_i$. For instance, one can take $U_1, \ldots, U_\ell$ as the non-empty interiors (for the Euclidean topology) of $T_1, \ldots, T_r$.

Our strategy to solve this problem is to first compute a polynomial Δ in $\mathbb{Q}[\boldsymbol{y}] - \{0\}$ defining a Zariski closed set $\mathscr{D} \subset \mathbb{C}^t$ such that $\mathscr{D}$ contains $\mathbb{R}^t - (U_1 \cup \cdots \cup U_\ell)$. The next lemma is immediate.

Lemma 6.2. *Let $\mathscr{E} \subset \mathbb{R}^t$ be a finite set of points which has a non-empty intersection with any of the connected components of the semi-algebraic set defined by $\Delta \neq 0$. For $1 \leq i \leq \ell$, $\mathscr{E} \cap U_i$ is not empty.*

Hence, computing sample points in each connected component of the set defined by $\Delta \neq 0$ (e.g. using the algorithm in [26] applied to the set defined by $z\Delta - 1 = 0$ where z is a new variable) is enough to obtain at least one point per connected component of $U_1, \ldots, U_\ell$. Finally, for each such a point y, it remains to count the number of connected components of the set S_y by using a roadmap algorithm. We call *partial semi-algebraic resolution* of (F, G) the data $(b_1, \eta_1), \ldots, (b_k, \eta_k)$ where b_i is the number of connected components of S_{η_i} and $\{\eta_1, \ldots, \eta_k\}$ has a non-empty intersection with each connected component of $U_1 \cup \cdots \cup U_\ell$.

Hence, our algorithm relies on three subroutines. The first one, which we call Eliminate, takes as input F and G, as well as $\boldsymbol{x}$ and $\boldsymbol{y}$ and outputs $\Delta \in \mathbb{Q}[\boldsymbol{y}]$ as above ; we let $\mathscr{D} = V(\Delta)$. The second one, which we call SamplePoints takes as input Δ and outputs a finite set of sample points $\{\eta_1, \ldots, \eta_k\}$ (with $\eta_i \in \mathbb{Q}^t$) which meets each connected component of $\mathbb{R}^t - \mathscr{D}$. The last one, which we call NumberOfConnectedComponents takes F_η and G_η and for some $\eta \in \mathbb{Q}^t$ and computes the number of connected components of the semi-algebraic set S_η. The algorithm is described hereafter.

Algorithm 1: ParametricSolve$(F, G, \boldsymbol{x}, \boldsymbol{y})$

Data: Finite sequences F and G in $\mathbb{Q}[\boldsymbol{x}, \boldsymbol{y}]$ with $\boldsymbol{x} = (x_1, \ldots, x_n)$ and $\boldsymbol{y} = (y_1, \ldots, y_t)$.
Result: a partial semi-algebraic resolution of (F, G)

```
Δ ← Eliminate(F, G, x, y)
{η_1, ..., η_k} ← SamplePoints(Δ ≠ 0)
for i from 1 to k do
    b_i = NumberOfConnectedComponents(F_{η_i}, G_{η_i})
end
return {(b_1, η_1), ..., (b_k, η_k)}.
```

While the rationale of algorithm ParametricSolve is mostly straightforward, detailing each of its subroutines is less. The easiest ones are SamplePoints and NumberOfConnectedComponents: they rely on known algorithms using the critical point method [5, 7], polar varieties [3, 4, 24, 26] and for computing roadmaps [6, 9, 27, 28]. The most difficult one is subroutine Eliminate. We provide a detailed description of it under the following regularity assumption. We say that (F, G) satisfies assumption (A)

(A) for any $\{i_1, \ldots, i_s\}$ in $\{1, \ldots, q\}$, the Jacobian matrix associated to $(f_1, \ldots, f_p, g_{i_1}, \ldots, g_{i_s})$ has maximal rank at any complex solution to

$$f_1 = \cdots = f_p = g_{i_1} = \cdots = g_{i_s} = 0$$

Note that using the Jacobian criterion [14, Chap. 16], it is easy to decide whether (A) holds. Note also that it holds generically.

For $\boldsymbol{i} = \{i_1, \ldots, i_s\} \subset \{1, \ldots, q\}$, under assumption (A), the algebraic set $V_{\boldsymbol{i}} \subset \mathbb{C}^{n+t}$ defined by

$$f_1 = \cdots = f_p = g_{i_1} = \cdots = g_{i_s} = 0.$$

are smooth and equidimensional and these systems generate radical ideals (applying the Jacobian criterion [14, Theorem 16.19]). Besides, the tangent space to $z \in V_{\boldsymbol{i}}$ coincides with the the (left) kernel of the Jacobian matrices associated to $(f_1, \ldots, f_p, g_{i_1}, \ldots, g_{i_s})$ at z.

Let I be the ideal generated by $(f_1, \ldots, f_p, g_{i_1}, \ldots, g_{i_s})$ and the maximal minors of the truncated Jacobian matrix associated to $(f_1, \ldots, f_p, g_{i_1}, \ldots, g_{i_s})$ obtained by removing the columns corresponding to the partial derivatives w.r.t. the $\boldsymbol{y}$-variables. Under assumption (A), one can compute the set of critical values of the restriction of the projection π to the algebraic set $V_{\boldsymbol{i}}$ by eliminating the variables $\boldsymbol{x}$ from I.

Hence, using elimination algorithms, which include Gröbner bases [15, 16] with elimination monomial orderings, or triangular sets (see e.g. [2, 32]) or geometric resolution algorithms [18–20], one can compute a polynomial $\Delta_{\boldsymbol{i}} \in \mathbb{Q}[\boldsymbol{y}]$ whose vanishing set is the set of critical values of the restriction of π to $V_{\boldsymbol{i}}$. By the algebraic Sard's theorem (see e.g. [28, App. A]), $\Delta_{\boldsymbol{i}}$ is not identically zero (the critical values are contained in a Zariski closed subset of $\mathbb{C}^t$).

Under assumption (A), we define the set of critical points (resp. values) of the restriction of π to the Euclidean closure of S as the union of the set of critical points (resp. values) of the restriction of π to $V_{\boldsymbol{i}} \cap \mathbb{R}^{n+t}$ when $\boldsymbol{i}$ ranges over the subsets of $\{1, \ldots, q\}$. We denote the Euclidean closure of S by $\overline{S}$, the set of critical points (resp. values) of the restriction of π to $\overline{S}$ by $\mathscr{W}(\pi, \overline{S})$ (resp. $\mathscr{D}(\pi, \overline{S})$). We say that S satisfies a *properness* assumption (P) if:

(P) the restriction of π to $\overline{S}$ is proper ($\forall y \in \mathbb{R}^t$, there exists a ball $B \ni y$ s.t. $\pi^{-1}(B) \cap \overline{S}$ is closed and bounded).

Our interest in critical points and values is motivated by the semi-algebraic version of Thom's isotopy lemma (see [13]) which states the following, under assumption (P). Take an open semi-algebraic subset $U \subset \mathbb{R}^t$ which does not meet the set of critical values of the restriction of π to $\overline{S}$, $y \in U$ and $E = \pi^{-1}(y) \cap S$. Then, there exists a semi-algebraic trivialization $\vartheta : U \times E \to \pi^{-1}(U) \cap S$.
Hence, $\cup_{\boldsymbol{i} \subset \{1,\ldots,q\}} \mathscr{D}(\pi, V_{\boldsymbol{i}})$ contains the boundaries of the open disjoint semi-algebraic set $U_1, \ldots, U_\ell$. Recall that by Sard's theorem it has co-dimension ≥ 1. This leads to the following algorithm.

Algorithm 2: EliminateProper($F, G, \boldsymbol{x}, \boldsymbol{y}$)

Data: Finite sequences F and G in $\mathbb{Q}[\boldsymbol{x}, \boldsymbol{y}]$ with $\boldsymbol{x} = (x_1, \ldots, x_n)$ and $\boldsymbol{y} = (y_1, \ldots, y_t)$, defining a semi-algebraic set $S \subset \mathbb{R}^n \times \mathbb{R}^t$.
Assumes that assumptions (A) and (P) hold.
Result: $\Delta \in \mathbb{Q}[\boldsymbol{y}]$ such that π realizes a fibration over all connected components of $\mathbb{R}^t - \{\Delta = 0\}$

```
for all subsets i in {1,...,q} do
    M ← maximal minors of jac([F, G_i], x)
    Δ_i ← AlgebraicElimination([F, G_i, M], x)
end
Δ ← ∏_i Δ_i.
return Δ.
```

Lemma 6.3. *On input (F, G) in $\mathbb{Q}[\boldsymbol{x}, \boldsymbol{y}]$ satisfying* (A)*, algorithm* EliminateProper *is correct.*

For some applications, deciding if (P) holds is easy (e.g. when the inequalities in G define a box). However, in general, one needs to generalize EliminateProper to situations where (P) does not hold. To do so, we use a classical technique from effective real algebraic geometry. Let ε be an infinitesimal and $\mathbb{R}\langle\varepsilon\rangle$ be the field of Puiseux series in ε with coefficients in $\mathbb{R}$. By [7, Chap. 2], $\mathbb{R}\langle\varepsilon\rangle$ is a real closed field and one can define semi-algebraic sets over $\mathbb{R}\langle\varepsilon\rangle^{n+t}$. In particular the set solutions in $\mathbb{R}\langle\varepsilon\rangle^{n+t}$ to the system defining S is a semi-algebraic set which we denote by $\mathrm{ext}(S, \mathbb{R}\langle\varepsilon\rangle)$. We refer to [7] for properties of real Puiseux series fields and semi-algebraic sets defined over such field. We make use of the notions of bounded points of $\mathbb{R}\langle\varepsilon\rangle^n$ over $\mathbb{R}$ (those whose all coordinates have non-negative valuation) and their limits in $\mathbb{R}$ (when $\varepsilon \to 0$). We denote by $\lim_0$ the operator taking the limits of such points.
For $a = (a_1, \ldots, a_n)$, we consider the intersection of $\mathrm{ext}(S, \mathbb{R}\langle\varepsilon\rangle)$ with the semi-algebraic set defined by

$$\Phi^{(a)} = a_1 x_1^2 + \cdots + a_n x_n^2 - 1/\varepsilon \leq 0$$

where $a_i > 0$ in $\mathbb{R}$ for $1 \leq i \leq n$. We denote by S'_ϵ this intersection. Since $a_i > 0$ for all $1 \leq i \leq n$, S'_ϵ satisfies (P).

Lemma 6.4. *Assume that (F, G) satisfies* (A)*. There exists a non-empty Zariski open set $\mathscr{A} \subset \mathbb{C}^n$ such that for any choice of $a = (a_1, \ldots, a_n) \in \mathscr{A}$, $(F, G^{(a)})$ satisfies* (A) *with $G^{(a)} = G \cup \{\Phi^{(a)}\}$.*

Proof. Let $\boldsymbol{i} = \{i_1, \ldots, i_s\} \subset \{1, \ldots, q\}$. We prove below that there exists a non-empty Zariski open set $\mathscr{A}_{\boldsymbol{i}} \subset \mathbb{C}^n$ such that for $(a_1, \ldots, a_n) \in \mathscr{A}_{\boldsymbol{i}}$, the following property $(\mathrm{A})_{\boldsymbol{i}}$ holds. Denoting by $G^{(a),\boldsymbol{i}}$ the sequence $(g_{i_1}, \ldots, g_{i_s}, \Phi^{(a)})$, the Jacobian matrix of $(F, G^{(a),\boldsymbol{i}})$ has maximal rank at any point of $V(F, G^{(a),\boldsymbol{i}})$. Taking the intersection of the (finitely many) $\mathscr{A}_{\boldsymbol{i}}$'s is then enough to define $\mathscr{A}$. Consider new indeterminates $\alpha_1, \ldots, \alpha_n$ and the polynomial $\Phi^{(\alpha)} = \alpha_1 x_1^2 + \cdots + \alpha_n x_n^2 - 1/\varepsilon$. Let Ψ be the map

$$\Psi : (x, a) \to F(x), g_{i_1}(x), \ldots, g_{i_s}(x), \Phi^{(a)}(x)$$

Observe that $\mathbf{0}$ is a regular value for Ψ since (F, G) satisfies (A). Hence, Thom's weak transversality theorem (see e.g. [28, App. B]) implies that there exists $\mathscr{A}_{\boldsymbol{i}}$ such that $(\mathrm{A})_{\boldsymbol{i}}$ for any $a \in \mathscr{A}_{\boldsymbol{i}}$. □

Assume for the moment that (F, G') satisfies assumption (A). Observe that the coefficients of F and G' lie in $\mathbb{Q}(\varepsilon)$. Hence, applying the subroutine EliminateProper to (F, G') and the above inequality will output a polynomial $\Delta_\varepsilon \in \mathbb{Q}(\varepsilon)[\boldsymbol{y}]$ such that the restriction of π to $\overline{S'_\epsilon}$ realizes a trivialization over each connected component of $\mathbb{R}\langle\varepsilon\rangle^t - \{\Delta_\varepsilon = 0\}$. Without loss of generality, one can assume that $\Delta_\varepsilon \in \mathbb{Q}[\varepsilon][\boldsymbol{y}]$ and has content 1. In other words, one can write $\Delta_\varepsilon = \Delta_0 + \varepsilon\tilde{\Delta}$ with $\Delta_0 \in \mathbb{Q}[\boldsymbol{y}]$ and $\tilde{\Delta} \in \mathbb{Q}[\varepsilon][\boldsymbol{y}]$.

Lemma 6.5. *Let U be a connected component of $\mathbb{R}^t - \{\Delta_0 = 0\}$. Then, there exists a semi-algebraically connected component U_ε of $\mathbb{R}\langle\varepsilon\rangle^t - \{\Delta_\varepsilon = 0\}$ such that* $\mathrm{ext}(U, \mathbb{R}\langle\varepsilon\rangle) \subset U_\varepsilon$.

Proof. Let y and y' be two distinct points in U. Since U is a semi-algebraically connected component of $\mathbb{R}^t - \{\Delta_0 = 0\}$, there exists a semi-algebraic continuous function $\gamma : [0, 1] \to U$ with $\gamma(0) = y$ and $\gamma(1) = y'$ such that Δ_0 is sign invariant over $\gamma([0, 1])$ (assume, without loss of generality that it is positive). Note also for all $t \in [0, 1]$, $\Delta_0(\gamma(t)) \in \mathbb{R}$. We deduce that $\Delta_\varepsilon(\gamma(t)) > 0$ for all $t \in [0, 1]$. Now, take $\vartheta \in \mathrm{ext}([0, 1], \mathbb{R}\langle\varepsilon\rangle)$. Observe that ϑ is bounded over $\mathbb{R}$ and then $\lim_0 \vartheta$ exists and lies in $[0, 1]$. We deduce that $\Delta_\varepsilon(\lim_0 \vartheta) > 0$ and its limit when $\varepsilon \to 0$ is $\Delta_0(\lim_0 \vartheta) > 0$ in $\mathbb{R}$. We deduce that $\Delta_\varepsilon(\vartheta) > 0$. Hence, Δ_ε is sign invariant over $\mathrm{ext}(\gamma([0, 1]), \mathbb{R}\langle\varepsilon\rangle)$ and then y and y' both lie in the same semi-algebraically connected component of $\mathbb{R}\langle\varepsilon\rangle^t - \{\Delta_\varepsilon = 0\}$. □

We deduce that there exists $b' \in \mathbb{N}$ such that for all $y \in U$, the number of semi-algebraically connected components of $S'_\varepsilon \cap \pi^{-1}(y)$ is b. Using the transfer principle as in [9], we deduce that there exists $e' \in \mathbb{R}$ positive and small enough such that, the following holds. There exists $b \in \mathbb{N}$ such that for all $e \in]0, e'[$ the number of connected components of $S \cap \{a_1 x_1^2 + \cdots + a_n x_n^2 \leq \frac{1}{e}\} \cap \pi^{-1}(y)$. is b when y ranges over U. This proves the following lemma.

Lemma 6.6. *Let U be as above. Then the number of connected components of S_y is invariant when y ranges over U.*

Finally, we can describe the subroutine Eliminate whose correctness follows from the previous lemma.

Algorithm 3: Eliminate($F, G, \boldsymbol{x}, \boldsymbol{y}$)

Data: Finite sequences F and G in $\mathbb{Q}[\boldsymbol{x}, \boldsymbol{y}]$ with $\boldsymbol{x} = (x_1, \ldots, x_n)$ and $\boldsymbol{y} = (y_1, \ldots, y_t)$, defining a semi-algebraic set $S \subset \mathbb{R}^n \times \mathbb{R}^t$.

Assumes that (F, G) satisfies assumption (A).

Result: $\Delta \in \mathbb{Q}[\boldsymbol{y}]$ such that the number of connected components of S_y is invariant when y ranges over a connected component of $\mathbb{R}^t - \{\Delta = 0\}$

1 Choose $a_1 > 0, \ldots, a_n > 0$ in $\mathbb{Q}$ randomly and let $g \leftarrow a_1 x_1^2 + \cdots + a_n x_n^2 \leq \frac{1}{\varepsilon}$

2 $\Delta \leftarrow$ EliminateProper($F, G \cup g, \boldsymbol{x}, \boldsymbol{y}$)

3 $\Delta \leftarrow$ Normalize(Δ) **return** Δ_0.

7 COMPUTATIONS

We have implemented several variants of the roadmap algorithms sketched in Section 5 as well as variants of the algorithm ParametricSolve. To perform algebraic elimination, we use Gröbner bases implemented in the FGb library by J.-C. Faugère [17]. The roadmap algorithm and the routines for computing sample points in semi-algebraic sets are implemented in the RAGlib library [25]. We have not directly applied the most general version of ParametricSolve to the polynomial B. Indeed, since its variables v_2, v_3, v_4 lie in the Cartesian product $\mathbb{P}^1(\mathbb{R}) \times \mathbb{P}^1(\mathbb{R}) \times \mathbb{P}^1(\mathbb{R})$ (which is compact), the projection on the parameter's space is proper and it suffices to compute critical loci of that projection. There is one technical (but easy) difficulty to overcome: polynomial B actually admits a positive dimensional singular locus. But an easy computation shows that this singular locus has one purely complex component (which satisfies $v_4^2 + 1$) which can then be forgotten. The other component has a projection on the paramaters'space which Zariski closed (it is contained in the set satisfied by $a_2 a_3 = 0$). This way, we directly obtain the following polynomial for Δ by computing the critical locus and consider additionally the set defined by $a_2 a_3 = 0$.

$$a_2 a_3 d_5 \ (a_2 + a_3 + d_5) \ (a_2 + a_3 - d_5)$$

Computing Δ as above does not take more than 3 sec. on a standard laptop using FGb. Getting sample points in the set defined by $\Delta \neq 0$ is trivial. We obtain the following 10 sample points using RAGlib

$$\{a_2 = -1, a_3 = -3, d_5 = 3\}, \{a_2 = -1, a_3 = -1, d_5 = 3\}, \{a_2 = -1, a_3 = 2, d_5 = 3\}, \{a_2 = -1, a_3 = 5, d_5 = 3\}, \{a_2 = -1, a_3 = \frac{1}{2}, d_5 = 3\}, \{a_2 = 1, a_3 = -120, d_5 = 118\}, \{a_2 = 1, a_3 = -118, d_5 = 118\}, \{a_2 = 1, a_3 = 1, d_5 = 118\}, \{a_2 = 1, a_3 = 118, d_5 = 118\}, \{a_2 = 1, a_3 = -1/2, d_5 = 118\}$$

Our implementation allows us to compute a roadmap for one sample point within 20 minutes on a standard laptop. Analyzing the connectivity of these roadmaps is longer as it takes 40 min. All in all, approximately 10 hours are required to handle this positive dimensional parametric system. The data we computed are available at `http://ecarp.lip6.fr/papers/materials/issac20/`. These computations allow to retrieve the conclusions of our theoretical analysis of the UR family. They illustrate that prototype implementations of our algorithms are becoming efficient enough to tackle automated kinematic singularity analysis in robotics.

REFERENCES

[1] Angeles, J. *Fundamentals of Robotic Mechanical Systems, Theory, Methods, and Algorithms.* Springer, 2007.

[2] Aubry, P., Lazard, D., and Maza, M. M. On the theories of triangular sets. *Journal of Symbolic Computation 28*, 1-2 (1999), 105–124.

[3] Bank, B., Giusti, M., Heintz, J., and Mbakop, G. Polar varieties and efficient real equation eolving: the hypersurface case. *J. of Complexity 13*, 1 (1997), 5–27.

[4] Bank, B., Giusti, M., Heintz, J., Safey El Din, M., and Schost, E. On the geometry of polar varieties. *Applicable Algebra in Engineering, Communication and Computing 21*, 1 (2010), 33–83.

[5] Basu, S., Pollack, R., and Roy, M.-F. On computing a set of points meeting every cell defined by a family of polynomials on a variety. *Journal of Complexity 13*, 1 (1997), 28 – 37.

[6] Basu, S., Pollack, R., and Roy, M.-F. Computing roadmaps of semi-algebraic sets on a variety. *Journal of the American Mathematical Society 13*, 1 (2000), 55–82.

[7] Basu, S., Pollack, R., and Roy, M.-F. *Algorithms in real algebraic geometry.* Springer-Verlag, 2003.

[8] Basu, S., and Roy, M.-F. Divide and conquer roadmap for algebraic sets. *Discrete & Computational Geometry 52*, 2 (2014), 278–343.

[9] Basu, S., Roy, M.-F., Safey El Din, M., and Schost, É. A baby step–giant step roadmap algorithm for general algebraic sets. *Foundations of Computational Mathematics 14*, 6 (2014), 1117–1172.

[10] Bochnak, J., Coste, M., and Roy, M.-F. *Real Algebraic Geometry.* Springer-Verlag, 1998.

[11] Canny, J. *The complexity of robot motion planning.* MIT press, 1988.

[12] Canny, J. Computing roadmaps of general semi-algebraic sets. *The Computer Journal 36*, 5 (1993), 504–514.

[13] Coste, M., and Shiota, M. Thom's first isotopy lemma: a semialgebraic version, with uniform bound. In *Real Analytic and Algebraic Geometry: Proc. of the International Conference, Trento* (1995), Walter de Gruyter, p. 83.

[14] Eisenbud, D. *Commutative Algebra: with a view toward algebraic geometry*, vol. 150. Springer Science & Business Media, 2013.

[15] Faugère, J.-C. A new efficient algorithm for computing gröbner bases (f4). *Journal of pure and applied algebra 139*, 1-3 (1999), 61–88.

[16] Faugère, J. C. A new efficient algorithm for computing gröbner bases without reduction to zero (f 5). In *Proc. of the 2002 international symposium on Symbolic and algebraic computation* (2002), ACM, pp. 75–83.

[17] Faugère, J.-C. Fgb: A library for computing gröbner bases. In *Mathematical Software - ICMS 2010* (Berlin, Heidelberg, September 2010), K. Fukuda, J. Hoeven, M. Joswig, and N. Takayama, Eds., vol. 6327 of *Lecture Notes in Computer Science*, Springer, pp. 84–87.

[18] Giusti, M., Heintz, J., Morais, J.-E., Morgenstern, J., and Pardo, L.-M. Straight-line programs in geometric elimination theory. *Journal of Pure and Applied Algebra 124* (1998), 101–146.

[19] Giusti, M., Heintz, J., Morais, J.-E., and Pardo, L.-M. When polynomial equation systems can be solved fast? In *AAECC-11* (1995), vol. 948 of *LNCS*, Springer, pp. 205–231.

[20] Giusti, M., Lecerf, G., and Salvy, B. A gröbner free alternative for polynomial system solving. *Journal of Complexity 17*, 1 (2001), 154 – 211.

[21] Gournay, L., and Risler, J.-J. Construction of roadmaps in semi-algebraic sets. *Applicable Algebra in Engineering, Communication and Computing 4*, 4 (1993), 239–252.

[22] Mezzarobba, M., and Safey El Din, M. Computing roadmaps in smooth real algebraic sets. In *Proc. of Transgressive Computing* (2006), J.-G. Dumas, Ed., pp. 327–338.

[23] Murray, R., Li, Z., and Sastry, S. *A Mathematical Introduction to Robotic Manipulation.* CRC Press Taylor & Francis Group, 1994.

[24] Safey El Din, M. Finding sampling points on real hypersurfaces is easier in singular situations. *MEGA (Effective Methods in Algebraic Geometry) Electronic proceedings* (2005).

[25] Safey El Din, M. Real algebraic geometry library. available at `http://www-polsys.lip6.fr/~safey`, 2007.

[26] Safey El Din, M., and Schost, E. Polar varieties and computation of one point in each connected component of a smooth real algebraic set. In *Proc. of the 2003 Int. Symp. on Symb. and Alg. Comp.* (NY, USA, 2003), ISSAC'03, ACM, pp. 224–231.

[27] Safey El Din, M., and Schost, E. A baby steps/giant steps probabilistic algorithm for computing roadmaps in smooth bounded real hypersurface. *Disc. Comput. Geom. 45*, 1 (2011), 181–220.

[28] Safey El Din, M., and Schost, É. A nearly optimal algorithm for deciding connectivity queries in smooth and bounded real algebraic sets. *Journal of the ACM (JACM) 63*, 6 (2017), 48.

[29] Schwartz, J. T., and Sharir, M. Algorithmic motion planning in robotics. In *Algorithms and Complexity.* Elsevier, 1990, pp. 391–430.

[30] Selig, J. *Geometric Fundamentals of Robotics.* Monographs in Computer Science. Springer, 2005.

[31] Spong, M., Hutchinson, S., and Vidyasagar, M. *Robot Dynamics and Control*, 2nd ed. Monographs in Computer Science. John Wiley & Sons, 2005.

[32] Wang, D. *Elimination methods.* Springer Science & Business Media, 2001.

[33] Wenger, P. Cuspidal robots. In *Singular Configurations of Mechanisms and Manipulators*, Z. D. Müller A., Ed. Springer International Publishing, 2019, pp. 67–100.

[34] Weyrer, M., Brandstötter, M., and Husty, M. Singularity avoidance control of a non-holonomic mobile manipulator for intuitive hand guidance. *Robotics 8*, 1 (2019).

Signature-based Algorithms for Gröbner Bases over Tate Algebras

Xavier Caruso
Université de Bordeaux, CNRS, INRIA
Bordeaux, France
xavier.caruso@normalesup.org

Tristan Vaccon
Université de Limoges; CNRS, XLIM
UMR 7252
Limoges, France
tristan.vaccon@unilim.fr

Thibaut Verron
Johannes Kepler University
Institute for Algebra
Linz, Austria
thibaut.verron@jku.at

ABSTRACT

Introduced by Tate in [Ta71], Tate algebras play a major role in the context of analytic geometry over the p-adics, where they act as a counterpart to the use of polynomial algebras in classical algebraic geometry. In [CVV19] the formalism of Gröbner bases over Tate algebras has been introduced and effectively implemented. One of the bottlenecks in the algorithms was the time spent on reduction, which are significantly costlier than over polynomials. In the present article, we introduce two signature-based Gröbner bases algorithms for Tate algebras, in order to avoid many reductions. They have been implemented in SageMath. We discuss their superiority based on numerical evidence.

CCS CONCEPTS

• **Computing methodologies** → **Algebraic algorithms**.

KEYWORDS

Algorithms, Power series, Tate algebra, Gröbner bases, F5 algorithm, p-adic precision

ACM Reference Format:
Xavier Caruso, Tristan Vaccon, and Thibaut Verron. 2020. Signature-based Algorithms for Gröbner Bases over Tate Algebras. In *International Symposium on Symbolic and Algebraic Computation (ISSAC '20), July 20–23, 2020, Kalamata, Greece.* ACM, New York, NY, USA, 8 pages. https://doi.org/10.1145/3373207.3404035

This article is dedicated to the memory of John Tate.

1 INTRODUCTION

For several decades, many computational questions arising from geometry and arithmetics have received much attention, leading to the development of more and more efficient algorithms and software. A typical example is the development of the theory of Gröbner bases, which provides nowadays quite efficient tools for manipulating ideals in polynomial algebras and, eventually, algebraic varieties and schemes [Magma, Macaulay2, Sage, Singular]. At the intersection of geometry and number theory, one finds p-adic geometry and, more precisely, the notion of p-adic analytic varieties first defined by Tate in [Ta71] (see also [FP04]), which plays an important role in many modern theories and achievements (*e.g.* p-adic cohomologies [LS07], p-adic modular forms [Go88]).

The main algebraic objects upon which Tate's geometry is built are Tate algebras and their ideals. In an earlier paper [CVV19], the authors started to study computational aspects related to Tate algebras, introduced Gröbner bases in this context and designed two algorithms (adapted from Buchberger's algorithm and the F4 algorithm, respectively) for computing them.

In the classical setting, the main complexity bottleneck in Gröbner bases computations is the time spent reducing elements modulo the basis. The most costly reductions are typically reductions to 0, because they require successively eliminating all terms from the polynomial; yet their output has little value for the rest of the algorithm. Fortunately, it turns out that many such reductions can be predicted in advance (for example those coming from the obvious equality $fg - gf = 0$) by keeping track of some information on the module representation of elements of an ideal, called their *signature*. This idea was first presented in Algorithm F5 [Fa02] and led to the development of many algorithms showing different ways to define signatures, to use them or to compute them. The interested reader can look at [EF17] for an extensive survey.

The Tate setting is not an exception to the wisdom that reductions are expensive. The situation is actually even worse since reductions to 0 are theorically the result of an *infinite* sequence of reduction steps *converging to* 0. In practice, the process actually stops because we are working at finite precision; however, the higher the precision is, the more expensive the reductions to 0 are, for no benefit. This observation motivates investigating the possibility of adding signatures to Gröbner bases algorithms for Tate series.

Our contribution. In this paper, we present two signature-based algorithms for the computation of Gröbner bases over Tate algebras. They differ in that they use different orderings on the signatures.

Our first variant, called the PoTe (position over term) algorithm, is directly adapted from the G2V algorithm [GGV10]. It adopts an incremental point of view and uses the so-called cover criterion [GVW16] to detect reductions to 0. A key difficulty in the Tate setting is that the usual way to handle signatures assumes the constant term 1 to be the smallest one. However, this assumption fails in the Tate setting. We solve this issue by importing ideas from the paper [L+18], in which the case of local algebras is addressed.

The first author is supported by the ANR grant CLap–CLap, referenced ANR-18-CE40-0026-01. The third author is supported by the FWF grant P31571-N32.

ISSAC '20, July 20–23, 2020, Kalamata, Greece

ACM ISBN 978-1-4503-7100-1/20/07…$15.00
https://doi.org/10.1145/3373207.3404035

In the classical setting, incremental algorithms have the disadvantage of sometimes computing larger Gröbner bases for intermediate ideals, only to discard them later on. In order to mitigate this misfeature, the F5 algorithm uses a signature ordering taking into account the degree of the polynomials first, in order to process lower-degree elements first. In the Tate setting, the degree no longer makes sense and a better measure of progression of the algorithms is the valuation. Nonetheless, in analogy with the classical setting, an incremental algorithm could perform intermediate computations to high valuation and just discard them later on. The second algorithm we will present, called the VaPoTe (valuation over position over term) algorithm, uses an analogous idea to that of F5 to mitigate this problem.

Organization of the article. In Section 2, we recall the basic definitions and properties of Tate algebras and Gröbner bases over them, together with the principles of the G2V algorithm. Sections 3 and 4 are devoted to the PoTe and the VaPoTe algorithms respectively: they are presented and their correctness and termination are proved. Finally, implementation, benchmarks and possible future improvements are discussed in Section 5.

Notations. Throughout this article, we fix a positive integer n and use the short notation $\mathbf{X}$ for $(X_1, \dots, X_n)$. Given $\mathbf{i} = (i_1, \dots, i_n) \in \mathbb{N}^n$, we shall write $\mathbf{X}^{\mathbf{i}}$ for $X_1^{i_1} \cdots X_n^{i_n}$.

2 INGREDIENTS

In this section, we present the two main ingredients we are going to mix together later on. They are, first, the G2V [GGV10] and GVW [GVW16] signature-based algorithms, and, second, the Tate algebras and the theory of Gröbner bases over them as developed in [CVV19].

2.1 The G2V algorithm

In what follows, we present the G2V algorithm which was designed by Gao, Guan and Volny IV in [GGV10] as an incremental variant of the classical F5 algorithm. Our presentation includes the cover criterion which was formulated later on in [GVW16] by Gao, Volny IV and Wang. The incremental point of view is needed for the application we will discuss in Section 4. Moreover we believe that it has two extra advantages: first, it leads to simplified notations and, more importantly, it shows clearly where intermediate inter-reductions are possible.

Let k be a field and $k[\mathbf{X}]$ denote the ring of polynomials over k with indeterminates $\mathbf{X}$. We endow $k[\mathbf{X}]$ with a fixed monomial order $\leq_\omega$. Let I_0 be an ideal in $k[\mathbf{X}]$. Let G_0 be a Gröbner basis of I_0 with respect to $\leq_\omega$. Let $f \in k[\mathbf{X}]$. We aim at computing a GB of the ideal $I = I_0 + \langle f \rangle$. Let $M \subset k[\mathbf{X}] \times k[\mathbf{X}]$ be the $k[\mathbf{X}]$-sub-module defined by the (u, v) such that $uf - v \in I_0$. The leading monomial $LM(u)$ of u is the *signature* of (u, v).

Definition 2.1 (Regular reduction)**.** Let $p_1 = (u_1, v_1)$ and $p_2 = (u_2, v_2)$ be in M. We say that p_1 is *top-reducible* by p_2 if

(1) either $v_2 = 0$ and $LM(u_2)$ divides $LM(u_1)$,
(2) or $v_1 v_2 \neq 0$, $LM(v_2)$ divides $LM(v_1)$ and:

$$\frac{LM(v_1)}{LM(v_2)} \cdot LM(u_2) \leq LM(u_1).$$

The corresponding top-reduction is

$$p = p_1 - t p_2 = (u_1 - t u_2, v_1 - t v_2)$$

where $t = \frac{LM(u_1)}{LM(u_2)}$ is the first case and $t = \frac{LM(v_1)}{LM(v_2)}$ in the second case. This top-reduction is called *regular* when $LM(u_1) > tLM(u_2)$, that is when the signature of the reduced pair p agrees with that of p_1; it is called *super* otherwise.

Definition 2.2 (Strong Gröbner basis)**.** A finite subset G of M is called a *strong Gröbner basis* (SGB, for short) of M if any nonzero $(u, v) \in M$ is top-reducible by some element of G.

The G2V strategy derives the computation of a Gröbner basis through the computation of an SGB. They are related through the following proposition.

Proposition 2.3. *Suppose that $G = \{(u_1, v_1), \dots, (u_s, v_s)\}$ is an SGB of M. Then:*

(1) $\{u$ s.t. $(u, 0) \in G\}$ is a Gröbner basis of $(I_0{:}f)$.
(2) $\{v$ s.t. $(u, v) \in G$ for some $u\}$ is a Gröbner basis of I.

To compute an SGB, we rely on J-pairs instead of S-polynomials.

Definition 2.4 (J-pair)**.** Let $p_1 = (u_1, v_1)$ and $p_2 = (u_2, v_2)$ be two elements in M such that $v_1 v_2 \neq 0$. Let $t = \mathrm{lcm}(LM(v_1), LM(v_2))$ and set $t_i = t/LM(v_i)$ for $i \in \{1, 2\}$. Then:

- if $t_1 LM(u_1) < t_2 LM(u_2)$, the *J-pair* of (p_1, p_2) is $t_2 p_2$,
- if $t_1 LM(u_1) > t_2 LM(u_2)$, the *J-pair* of (p_1, p_2) is $t_1 p_1$,
- if $t_1 LM(u_1) = t_2 LM(u_2)$, the *J-pair* of (p_1, p_2) is not defined.

Definition 2.5 (Cover)**.** We say that $p = (u, v)$ is *covered* by $G \subset M$ if there is a pair $(u_i, v_i) \in G$ such that $LM(u_i)$ divides $LM(u)$ and:

$$\frac{LM(u)}{LM(u_i)} \cdot LM(v_i) < LM(v).$$

Theorem 2.6 (Cover Theorem). *Let G be a finite subset of M such that:*

- *G contains $(1, f)$;*
- *the set $\{g \in k[\mathbf{X}] : (0, g) \in G\}$ forms a Gröbner basis of I_0.*

Then G is an SGB of M iff every J-pair of G is covered by G.

This theorem leads naturally to the G2V algorithm (see [GGV10, Fig. 1]) which is rephrased hereafter in Algorithm 1 (page 4). We underline that, in Algorithm 1, the SGB does not entirely appear. Indeed, we remark that one can always work with pairs $(LM(u), v)$ in place of (u, v), reducing then drastically the memory occupation and the complexity. The algorithm maintains two lists G and S which are related to the SGB in construction as follows: $G \cup (S \times \{0\})$ is equal to the set of all $(LM(u), v)$ when (u, v) runs over the SGB. The criterion coming from the cover theorem is implemented on lines 10 and 11: the first (resp. the second) statement checks if (u, v) is covered by an element of G (resp. an element of $S \times \{0\}$).

Syzygies. The G2V algorithm does not give a direct access to the module of syzygies of the ideal. However, it does give access to a GB of $(I_0{:}f)$ (see Proposition 2.3), from which one can recover partial information about the syzygies, as shown below.

Definition 2.7. Given $f_1, \dots, f_m \in k[\mathbf{X}]$, we define

$$Syz(f_1, \dots, f_m) = \left\{ (a_1, \dots, a_m) \in k[\mathbf{X}]^m \text{ s.t. } \sum_{i=1}^{m} a_i f_i = 0 \right\}.$$

Lemma 2.8. *Let $f_1, \dots, f_m$ generate I_0 and let $u_1, \dots, u_s$ generate $(I_0{:}f)$. For $i \in \{1, \dots, s\}$, we write*

$$-u_i f = a_{i,1} f_1 + \cdots + a_{i,m} f_m \qquad (a_{i,j} \in k[\mathbf{X}])$$

and define $z_i = (a_{i,1}, \dots, a_{i,m}, u_i) \in Syz(f_1, \dots, f_m, f)$. Then

$$Syz(f_1, \dots, f_m, f) = (Syz(f_1, \dots, f_m) \times \{0\}) + \langle z_1, \dots, z_s \rangle .$$

Proof. Let $(a_1, \dots, a_m, u) \in Syz(f_1, \dots, f_m, f)$. Then $u \in (I_0{:}f)$ and we can write $u = \sum_{i=1}^{s} b_i u_i$. Then the syzygy $(a_1, \dots, a_m, u) - \sum_{i=1}^{s} b_i z_i$ has its last coordinate equal to 0 and thus belongs to $(Syz(f_1, \dots, f_m) \times \{0\})$, which is enough to conclude. □

2.2 Tate algebras

Definitions. We fix a field K equipped with a discrete valuation $\mathrm{val} : K \to \mathbb{Z} \sqcup \{+\infty\}$, normalized by $\mathrm{val}(K^\times) = \mathbb{Z}$. We assume that K is complete with respect to the distance defined by val. We let K° be the subring of K consisting of elements of nonnegative valuation and π be a uniformizer of K, that is an element of valuation 1. We set $k = K^\circ/\pi K^\circ$. The Tate algebra $K\{\mathbf{X}\}$ is defined by:

$$K\{\mathbf{X}\} := \Big\{ \sum_{\mathbf{i} \in \mathbb{N}^n} a_\mathbf{i} \mathbf{X}^\mathbf{i} \text{ s.t. } a_\mathbf{i} \in K \text{ and } \mathrm{val}(a_\mathbf{i}) \xrightarrow[|\mathbf{i}| \to +\infty]{} +\infty \Big\}$$

Series in $K\{\mathbf{X}\}$ have a natural analytic interpretation: they are analytic functions on the closed unit disc in K^n. We recall that $K\{\mathbf{X}\}$ is equipped with the so-called Gauss valuation defined by:

$$\mathrm{val}\Big(\sum_{\mathbf{i} \in \mathbb{N}^n} a_\mathbf{i} X^\mathbf{i} \Big) = \min_{\mathbf{i} \in \mathbb{N}^n} \mathrm{val}(a_\mathbf{i}).$$

Series with nonnegative valuation form a subring $K\{\mathbf{X}\}^\circ$ of $K\{\mathbf{X}\}$. The reduction modulo π defines a surjective homomorphism of rings $K\{\mathbf{X}\}^\circ \to k[\mathbf{X}]$.

Terms and monomials. By definition, an integral *Tate term* is an expression of the form $a\mathbf{X}^\mathbf{i}$ with $a \in K^\circ$, $a \neq 0$ and $\mathbf{i} \in \mathbb{N}^n$. Integral Tate terms form a monoid, denoted by $T\{\mathbf{X}\}^\circ$, which is abstractly isomorphic to $(K^\circ \backslash \{0\}) \times \mathbb{N}^n$. We say that two Tate terms $a\mathbf{X}^\mathbf{i}$ and $b\mathbf{X}^\mathbf{j}$ are equivalent when $\mathrm{val}(a) = \mathrm{val}(b)$ and $\mathbf{i} = \mathbf{j}$. Tate terms modulo equivalence define a quotient $\mathbb{T}\{\mathbf{X}\}^\circ$ of $T\{\mathbf{X}\}^\circ$, which is isomorphic to $\mathbb{N} \times \mathbb{N}^n$. The image in $\mathbb{T}\{\mathbf{X}\}^\circ$ of a term $t \in T\{\mathbf{X}\}^\circ$ is called the *monomial* of t and is denoted by $\mathrm{mon}(t)$.

We fix a monomial order $\leq_\omega$ on $\mathbb{N}^n$ and order $\mathbb{T}\{\mathbf{X}\}^\circ \simeq \mathbb{N} \times \mathbb{N}^n$ lexicographically by block with respect to the reverse natural ordering on the first factor $\mathbb{N}$ and the order $\leq_\omega$ on $\mathbb{N}^n$. Pulling back this order along the morphism mon, we obtain a preorder of $T\{\mathbf{X}\}^\circ$ that we shall continue to denote by $\leq$. The *leading term* of a Tate series $f = \sum a_\mathbf{i} \mathbf{X}^\mathbf{i} \in K\{\mathbf{X}\}^\circ$ is defined by:

$$LT(f) = \max_{\mathbf{i} \in \mathbb{N}^n} a_\mathbf{i} X^\mathbf{i} \in T\{\mathbf{X}\}^\circ.$$

We observe that the $a_\mathbf{i} X^\mathbf{i}$'s are pairwise nonequivalent in $T\{\mathbf{X}\}^\circ$, showing that there is no ambiguity in the definition of $LT(f)$. The *leading monomial* of f is by definition $LM(f) = \mathrm{mon}(LT(f))$.

Gröbner bases. The previous inputs allow us to define the notion of Gröbner bases for an ideal of $K\{\mathbf{X}\}^\circ$.

Definition 2.9. Let I be an ideal of $K\{\mathbf{X}\}^\circ$. A family $(g_1, \dots, g_s) \in I^s$ is a Gröbner basis (in short, GB) of I if, for all $f \in I$, there exists $i \in \{1, \dots, s\}$ such that $LM(g_i)$ divides $LM(f)$.

A classical argument shows that any GB of an ideal I generates I. The following theorem is proved in [CVV19, Theorem 2.19].

Theorem 2.10. *Every ideal of $K\{\mathbf{X}\}^\circ$ admits a GB.*

The explicit computation of such a GB is of course a central question. It was addressed in [CVV19], in which the authors describe a Buchberger algorithm and an F4 algorithm for this task. The aim of the present article is to improve on these results by introducing signatures in this framework and eventually design F5-like algorithms for the computation of GB over Tate algebras.

Important remarks. For the simplicity of exposition, we chose to restrict ourselves to the Tate algebra $K\{\mathbf{X}\}$ and not consider the variants $K\{\mathbf{X}; \mathbf{r}\}$ allowing for more general radii of convergence. However, using the techniques developed in [CVV19] (paragraph *General log-radii* of Section 3.2), all the results we will obtain in this article can be extended to $K\{\mathbf{X}; \mathbf{r}\}$.

In practice, the elements of K need to be truncated to fit in the memory of the computer; when doing so, we say that we are working at *finite precision*. We refer to [CVV19] (see in particular Theorem 3.8 and comments around it) for a thorough study of the behaviour of GB with respect to finite precision computations.

3 POSITION OVER TERM

The goal of this section is to adapt the G2V algorithm to the setting of Tate algebras. Although all definitions, statements and algorithms are *formally* absolutely parallel to the classical setting, proofs in the framework of Tate algebras are more subtle, due to the fact that the orderings on Tate terms are not well-founded but only topologically well-founded. In order to accomodate this weaker property, we import ideas from [L+18] where the case of local rings is considered.

3.1 The PoTe algorithm

We fix a monomial order $\leq_\omega$ of $\mathbb{N}^n$ and write $\leq$ for the term order on $T\{\mathbf{X}\}^\circ$ it induces. We consider an ideal I_0 in $K\{\mathbf{X}\}^\circ$ along with a GB G_0 of I_0. Let $f \in K\{\mathbf{X}\}^\circ$. We are interested in computing a GB of $I = I_0 + \langle f \rangle$. Mimicking what we have recalled in §2.1, we introduce the $K\{\mathbf{X}\}^\circ$-sub-module $M \subset K\{\mathbf{X}\}^\circ \times K\{\mathbf{X}\}^\circ$ consisiting of pairs (u, v) such that $uf - v \in I_0$. The definitions of regular reduction (Definition 2.1), strong Gröbner bases (Definition 2.2), J-pair (Definition 2.4) and cover (Definition 2.5) extend *verbatim* to the context of Tate algebras, with the precaution that the leading monomial is now computed with respect to the order $\leq$ as explained in Section 2.2.

Proposition 3.1. *Suppose that $G = \{(u_1, v_1), \dots, (u_s, v_s)\}$ is an SGB of M. Then:*

(1) $\{u$ s.t. $(u, 0) \in G\}$ is a Gröbner basis of $(I_0 : f)$.
(2) $\{v$ s.t. $(u, v) \in G$ for some $u\}$ is a Gröbner basis of I.

Proof. Let G be an SGB of M.

Let $h \in (I_0{:}f)$. Then $hf \in I_0$ and $(h, 0) \in M$. By definition, since G is an SGB of M, there exists $(u, 0) \in G$ such that $LM(u)$ divides $LM(h)$. This implies the first statement of the proposition.

Let now $h \in I$. If $LM(h) \in I_0$, there exists a pair $(0, h') \in M$ with $LM(h) = LM(h')$. This pair is divisible by some $(0, v) \in G$, proving that $LM(v)$ divides $LM(h') = LM(h)$ in this case. We now

Algorithm 1: G2V (resp. PoTe) algorithm

input : $f_1, \dots, f_m$ in $k[\mathbf{X}]$ (resp. $K\{\mathbf{X}\}^\circ$)
output: a GB of the ideal generated by the f_i's

$Q \leftarrow (f_1, \dots, f_m)$
$GBasis \leftarrow \emptyset$

for $f \in Q$ **do**
  $G \leftarrow \{(0, g) : g \in GBasis\} \cup \{(1, f)\}$
  $S \leftarrow \{LM(g) : g \in GBasis\}$
  $B \leftarrow \{\text{J-pair}((1, f), (0, g)) : g \in GBasis\}$
  while $B \neq \emptyset$ **do**
    pop (u, v) from B, with smallest u
    if (u, v) *is covered by* G **then continue**
    if u *is divisible by some* $s \in S$ **then continue**
    $v_0 \leftarrow \text{regular\_reduce}\ (u, v, G)$
    if $v_0 = 0$ **then**
      add u to S
    else
      for $(s, g) \in G$ **do**
        if J-pair$((u, v_0), (s, g))$ *is defined* **then**
          add J-pair$((u, v_0), (s, g))$ to B
      add (u, v_0) to G
  $GBasis \leftarrow \{v : (u, v) \in G\}$
return *GBasis*

Algorithm 2: VaPoTe algorithm

input : $f_1, \dots, f_m$ in $K\{\mathbf{X}\}^\circ$
output: a GB of the ideal generated by the f_i's

$Q \leftarrow (f_1, \dots, f_m)$
$GBasis \leftarrow \emptyset$
while $Q \neq \emptyset$ **do**
  pop f from Q, with smallest valuation
  $G \leftarrow \{(0, g) : g \in GBasis\} \cup \{(1, f)\}$
  $S \leftarrow \{LM(g) : g \in GBasis\}$
  $B \leftarrow \{\text{J-pair}((1, f), (0, g)) : g \in GBasis\}$
  while $B \neq \emptyset$ **do**
    pop (u, v) from B, with smallest u
    if (u, v) *is covered by* G **then continue**
    if u *is divisible by some* $s \in S$ **then continue**
    $v_0 \leftarrow \text{regular\_reduce}\ (u, v, G)$
    if $\text{val}(v_0) > \text{val}(f)$ **then**
      add u to S; **add** v_0 to Q
    else
      for $(s, g) \in G$ **do**
        if J-pair$((u, v_0), (s, g))$ *is defined* **then**
          add J-pair$((u, v_0), (s, g))$ to B
      add (u, v_0) to G
  $GBasis \leftarrow \{v : (u, v) \in G\}$
return *GBasis*

suppose that $LM(h) \notin LM(I_0)$. This assumption implies that any $a \in K\{\mathbf{X}\}^\circ$ with $(a, h) \in M$ (*i.e.* $af - h \in I_0$) must satisfy $LM(a) \geq LM(h)/LM(f)$. We can then choose a series $a \in K\{\mathbf{X}\}^\circ$ such that $(a, h) \in M$ and $LM(a)$ is minimal for this property. Moreover, since G is an SGB, the pair (a, h) has to be top-reducible by some $(u, v) \in G$. If $v \neq 0$, we deduce that $LM(v)$ divides $LM(h)$. Otherwise, letting $t = LT(a)/LT(u)$, we obtain $(a - tu, h) \in M$ with $LM(a - tu) < LM(a)$, contradicting the minimality of $LM(a)$. As a conclusion, we have shown that $LM(v)$ divides $LM(h)$ in all cases, which readily implies (2). □

Theorem 3.2 (Cover Theorem). *Let G be a finite subset of M such that:*

- *G contains $(1, f)$;*
- *the set $\{g \in K\{\mathbf{X}\}^\circ : (0, g) \in G\}$ forms a Gröbner basis of I_0.*

Then G is an SGB of M iff every J-pair of G is covered by G.

The proof of Theorem 3.2 is presented in Section 3.2 below. Before this, let us observe that Theorem 3.2 readily shows that the G2V algorithm (see Algorithm 1) extends *verbatim* to Tate algebras. The resulting algorithm is called the PoTe[1] algorithm. The correctness of the PoTe algorithm is clear thanks to Theorem 3.2. Its termination is not *a priori* guaranteed because the call to regular_reduce may enter an infinite loop (see [CVV19, Sec. 3.1]). However, if we assume that all regular reductions terminate (which is guaranteed in practice by working at finite precision), the PoTe algorithm terminates as well thanks to the Noetherianity of $K\{\mathbf{X}\}^\circ$.

[1] PoTe means "**Po**sition over **Te**rm".

3.2 Proof of the cover theorem

Throughout this subsection, we consider a finite set G satisfying the assumptions of Theorem 3.2.

We first assume that G is an SGB of M. Let $p_1, p_2 \in G$ and write $p_i = (u_i, v_i)$ for $i \in \{1, 2\}$. We set $t = \text{lcm}(LM(v_1), LM(v_2)) \in \mathbb{T}\{\mathbf{X}\}^\circ$ and $t_i = t/LM(v_i)$. If $LM(t_1u_1) = LM(t_2u_2)$, the J-pair of (p_1, p_2) is not defined and there is nothing to prove. Otherwise, if i (resp. j) is the index for which $LM(t_iu_i)$ is maximal (resp. $LM(t_ju_j)$ is minimal), the J-pair of (p_1, p_2) is t_ip_i, which is regularly top-reducible by p_j. Continuing to apply regular top-reductions by elements of G as long as possible, we reach a pair $(u_0, v_0) \in M$ which is no longer regularly top-reducible by any element of G and for which $LM(u_0) = LM(t_iu_i)$ and $LM(v_0) < LM(t_iv_i)$. Since G is an SGB of M, (u_0, v_0) must be super top-reducible by some pair $(u, v) \in G$. By definition of super top-reducibility, $LM(u)$ divides $LM(u_0) = LM(t_iu_i)$ and $LM(v) \cdot LM(u_0) = LM(v_0) \cdot LM(u)$. This shows that $LM(v) \cdot LM(u_i) < LM(v_i) \cdot LM(u)$ and then that (u, v) covers t_ip_i.

We now focus on the converse and assume that each J-pair of G is covered by G. We define:

$$W = \left\{ (u, v) \in M,\ \text{top-reducible by no pair of } G \right\}$$

and assume by contradiction that W is not empty.

Lemma 3.3. *The set W does not contain any pair of the form (u, v) with $u = 0$ or $LM(v) \in LM(I_0)$.*

Proof. By our assumptions, if $LM(v) \in LM(I_0)$, v is reducible by some g with $(0, g) \in G$. In particular, (u, v) is top-reducible by

$(0, g)$ and cannot be in W. If $u = 0$, then $v \in I_0$ and we are reduced to the previous case. □

Lemma 3.4. *Let $p_0 = (u_0, v_0) \in W$. Then there exists a pair $p_1 = (u_1, v_1) \in G$ such that $LT(u_1)$ divides $LT(u_0)$, say $LT(u_0) = t_1 LT(u_1)$, and $t_1 LT(v_1)$ is minimal for this property.*

Furthermore, $t_1 p_1$ is not regularly top-reducible by G.

Proof. We have already noticed that $u_0 \neq 0$. Since $(1, f) \in G$, there exists a pair in G satisfying the first condition. Since G is finite, there exists one that further satisfies the minimality condition.

We assume by contradiction that $t_1 p_1$ is regularly top-reducible by G. Consider $p_2 = (u_2, v_2) \in G$ be a regular reducer of $t_1 p_1$, in particular there exists a term t_2 such that $t_2 LT(v_2) = t_1 LT(v_1)$, and $t_2 LT(u_2) < t_1 LT(u_1)$. The J-pair of p_1 and p_2 is then defined and equals $\tau \cdot (u_1, v_1)$ with τ dividing t_1. Write $t_1 = \tau t_1'$ for some term t_1'. By hypothesis, this J-pair is covered, so there exists $P = (U, V) \in G$ and a term θ such that $\theta \cdot LT(U) = \tau \cdot LT(u_1)$ and $\theta \cdot LT(V) < \tau \cdot LT(v_1)$. As a consequence:

$$t_1' \theta \cdot LT(U) = t_1 \cdot LT(u_1) = LT(u_0)$$
$$t_1' \theta \cdot LT(V) < t \cdot LT(v_1).$$

So $t_1' P$ contradicts the minimality of p_1. □

Let ν be the minimal valuation of a series v for which $(u, v) \in W$. We make the following additional assumption: $\nu < +\infty$. In other words, we assume that W contains at least one element of the form (u, v) with $v \neq 0$. We set:

$$W_1 = \left\{ (u, v) \in W \text{ s.t. } \mathrm{val}(LM(v)) = \nu \right\}.$$

Lemma 3.5. *The set $L = \{LM(u) : (u, v) \in W_1\}$ admits a minimal element.*

Proof. We assume by contradiction that L does not have a minimal element. Thus, we can construct a sequence $(u_k, v_k)_{k \geq 1}$ with values in W_1 such that $LM(u_k)$ is strictly decreasing. As a consequence, in the Tate topology, $u_k f$ converges to 0. Hence, for k large enough, $\mathrm{val}(u_k f) > \nu = \mathrm{val}(v_k)$. From $W_1 \subset M$, we get $v_k - u_k f \in I_0$ and $LM(v_k) = LM(v_k - u_k f) \in LM(I_0)$. By Lemma 3.3, this is a contradiction. □

Let W_2 be the subset of W_1 consisting of pairs (u, v) for which $LM(u)$ is minimal. Note that by Lemma 3.3, this minimal value is nonzero.

Lemma 3.6. *For any $(u_1, v_1), (u_2, v_2) \in W_2$, $LM(v_1) = LM(v_2)$.*

Proof. Let (u_1, v_1) and (u_2, v_2) in W_2, and assume that the leading terms are not equivalent, that is $LM(v_1) \neq LM(v_2)$. Without loss of generality, we can assume that $LM(v_1) > LM(v_2)$. By construction of W_2, $LM(u_1) = LM(u_2)$, that is $LT(u_1) = aLT(u_2)$ for some $a \in K$, $\mathrm{val}(a) = 0$. Since u_1 and u_2 are nonzero, we can write $u_1 = LT(u_1) + r_1$ and $u_2 = LT(u_2) + r_2$. Eliminating the leading terms, we obtain a new element $(u', v') = (r_1 - ar_2, v_1 - av_2)$. By assumption, $LM(v') = LM(v_1)$, and $LM(u') < LM(u_1)$. Observe that (u', v') cannot be top-reduced by G as otherwise, (u_1, v_1) would also be top-reducible by G. Hence $(u', v') \in W_1$, contradicting the minimality of $LM(u_1)$. □

Let now $p_0 = (u_0, v_0) \in W_2$. From Lemma 3.4, there exists $p_1 = (u_1, v_1) \in G$ and a term t such that $LT(tu_1) = LT(u_0)$ and tp_1 is not regular top-reducible by G. We define

$$p_* = (u_*, v_*) = p_0 - tp_1 = (u_0, v_0) - t(u_1, v_1).$$

We remark that $LM(u_*) < LM(u_0)$. Moreover $LM(v_0) \neq LM(tv_1)$ since otherwise p_0 would be top-reducible by p_1, contradicting the fact that $p_0 \in W$.

We first examine the case where $LM(v_0) < LM(tv_1)$. It implies that $LM(v_*) = LM(tv_1) > LM(v_0)$. Let us prove first that $p_* \notin W$. We argue by contradiction. From $p_* \in W$, we would derive $\mathrm{val}(v_*) \geq \nu = \mathrm{val}(v_0)$ and then $\mathrm{val}(v_*) = \mathrm{val}(v_0)$ since the inequality in the other direction holds by assumption. We conclude by noticing that $LM(u_*) < LM(u_0)$ contradicts the minimality of $LM(u_0)$. So $p_* \notin W$, *i.e.* p_* is top-reducible by G. Let $p_2 = (u_2, v_2) \in G$ be top-reducing p_*. If $v_2 = 0$, then $LM(u_2)$ divides $LM(u_*)$. Besides, the pair $p_*' = (u_*', v_*) = \left(u_* - \frac{LT(u_*)}{LT(u_2)} u_2, v_*\right)$ satisfies $LM(u_*') < LM(u_*)$ and thus cannot be in W either. We iterate this process until we can only find a reductor $q = (U, V) \in G$ with $V \neq 0$. Let $t_2 = LM(v_*)/LM(V)$. Then $t_2 LM(V) = LM(v_*) = LM(tv_1)$ and $t_2 LM(U) \leq LM(u_*) < LM(tu_1)$ if $U \neq 0$. Therefore q regularly top-reduces tp_1, which contradicts Lemma 3.4.

Let us now move to the case where $LM(v_0) > LM(tv_1)$. Then $LM(v_*) = LM(v_0)$. Since $LM(u_*) < LM(u_0)$, it follows that $p_* \notin W$, *i.e.* p_* is top-reducible by G. As in the previous case, we construct $q = (U, V) \in G$ with $V \neq 0$, and a term t_2 with the properties that $t_2 LM(V) = LM(v_*) = LM(v_0)$ and $t_2 LM(U) \leq LM(u_*) < LM(u_0)$ if $U \neq 0$. Thus q regularly top-reduces p_0, which contradicts $p_0 \in W$.

As a conclusion, in both cases, we have reached a contradiction. This ensures that $\nu = +\infty$. In particulier, W contains an element p_0 of the form $(u_0, 0)$. Let $p_1 = (u_1, v_1) \in G$ be given by Lemma 3.4. If $v_1 = 0$, this pair would be a reducer of $(u_0, 0) \in W$, which is a contradiction. So $v_1 \neq 0$. Set $t = \frac{LT(u)}{LT(u_1)}$. Let:

$$p_* = (u_*, v_*) = (u_0, 0) - t(u_1, v_1) = (u_0 - tu_1, -v_1)$$

Then $LM(u_*) < LM(u_0)$ and $LM(v_*) = tLM(v_1)$. From $v_1 \neq 0$, we deduce $p_* \notin W$. So p_* is top-reducible by $p_2 = (u_2, v_2) \in G$, meaning that there exists a term t_1 such that $t_1 LM(v_2) = LM(v_*) = tLM(v_1)$ and $t_1 LM(u_2) \leq LM(u_*) < tLM(u_1)$. So p_2 is a regular top-reducer of tp_1, which contradicts Lemma 3.4.

Finally, we conclude that W is empty. By construction, G is an SGB of M.

4 VALUATION OVER POSITION OVER TERM

In this section, we design a variant of the PoTe algorithm in which, roughly speaking, signatures are first ordered by increasing valuations.

4.1 The VaPoTe algorithm

The VaPoTe[2] algorithm is Algorithm 2 (page 4). It is striking to observe that it looks formally very similar to the PoTe Algorithm (Algorithm 1) as they only differ on lines 3–4 and, more importantly, on lines 13–14. However, these slight changes may have significant consequences on the order in which the inputs are processed,

[2] VaPoTe means "**Va**luation over **Po**sition over **Te**rm"

implying possibly important differences in the behaviour of the algorithms.

The VaPoTe algorithm has a couple of interesting features. First, if we stop the execution of the algorithm at the moment when we first reach a series f of valuation greater than N on line 4, the value of *GBasis* is a GB of the image of $I = \langle f_1, \dots, f_m \rangle$ in $K\{\mathbf{X}\}^\circ/\pi^N K\{\mathbf{X}\}^\circ$. In other words, the VaPoTe algorithm can be used to compute GB of ideals of $K\{\mathbf{X}\}^\circ/(\pi^N) \simeq K^\circ[\mathbf{X}]/(\pi^N)$ (for our modified order) as well.

Secondly, Algorithm 2 remains correct if the reduction on line 12 is interrupted as soon as the valuation rises. The property allows for delaying some reductions, which might be expensive at one time but cheaper later (because more reductors are available). It also has a theoretical interest because the reduction process may *a priori* hang forever (if we are working at infinite precision); interrupting it prematurely removes this defect and leads to more satisfying termination results.

4.2 Proof of correctness and termination

We introduce some notation. For a series $f \in K\{\mathbf{X}\}^\circ$, we write $v(f) = \pi^{-\operatorname{val}(f)} f$ (which has valuation 0 by construction) and define $\rho(f)$ as the image of $v(f)$ in $K\{\mathbf{X}\}^\circ/\pi K\{\mathbf{X}\}^\circ \simeq k[\mathbf{X}]$. More generally if A is a subset of $K\{\mathbf{X}\}^\circ$, we define $v(A)$ and $\rho(A)$ accordingly.

We consider $f_1, \dots, f_m \in K\{\mathbf{X}\}^\circ$ and write I for the ideal of $K\{\mathbf{X}\}^\circ$ they generate. For an integer N, we set $I_N = I \cap (\pi^N K\{\mathbf{X}\}^\circ)$. Clearly $I_{N+1} \subset I_N$ for all N. Let $\bar{I}_N$ be the image of $\pi^{-N} I_N$ in $k[\mathbf{X}]$; we have a canonical isomorphism $\bar{I}_N \simeq I_N/I_{N+1}$. Besides, the morphism $I_N \to I_{N+1}$, $f \mapsto \pi f$ induces an inclusion $\bar{I}_N \hookrightarrow \bar{I}_{N+1}$. Hence, the $\bar{I}_N$'s form a nondecreasing sequence of ideals of $k[\mathbf{X}]$.

We define Q_{all} as the set of all series that are popped from Q on line 13 during the execution of Algorithm 2. Since the algorithm terminates when Q is empty, Q_{all} is also the set of all series that have been in Q at some moment. For an integer N, we define

$$Q_{>N} = \left\{ f \in Q_{\text{all}} \text{ s.t. } \operatorname{val}(f) > N \right\}.$$

and similarly Q_N and $Q_{\leq N}$. Let also τ_N be the first time we enter in the while loop on line 3 with $Q \subset \pi^N K\{\mathbf{X}\}^\circ$. If this event never occurs, τ_N is defined as the time the algorithm exits the main while loop. We finally let $GBasis_N$ be the value of the variable *GBasis* at the checkpoint τ_N.

Lemma 4.1. *Between the checkpoints τ_N and τ_{N+1}:*
(1) the elements popped from Q are exactly those of Q_N, and
(2) the "reduction modulo π^{N+1}" of the VaPoTe algorithm behaves like the G2V algorithm, with input polynomials $\rho(Q_N)$ and initial value of GBasis set to $\rho(GBasis_N)$.

Proof. We observe that, after the time τ_N, only elements with valuation at least N+1 are added to Q. The first statement then follows from the fact that the elements of Q have been popped by increasing valuation. The second statement is a consequence of (1) together with the fact that all f and v manipulated by Algorithm 2 between the times τ_N and τ_{N+1} have valuation N. □

Since the G2V algorithm terminates for polynomials over a field, Lemma 4.1 ensures that each checkpoint τ_N is reached in finite time if the call to regular_reduce does not hang forever. This latter property holds when we are working at finite precision and is also guaranteed if we interrupt the reduction as soon as the valuation raises.

We are now going to relate the ideals $\bar{I}_N$ with the sets Q_N, $Q_{\leq N}$ and $Q_{>N}$. For this, we introduce the syzygies between the elements of $\rho(Q_{\leq N})$. More precisely, we set:

$$S_N = \left\{ (a_f)_{f \in Q_{\leq N}} \text{ s.t. } \sum_{f \in Q_{\leq N}} a_f v(f) \equiv 0 \pmod{\pi} \right\}.$$

and let $\bar{S}_N$ be the image of S_N under the projection $K\{\mathbf{X}\}^\circ \to k[\mathbf{X}]$; in other words, $\bar{S}_N$ is the module of syzygies of the set $\rho(Q_{\leq N})$, *i.e.* $\bar{S}_n = Syz(\rho(Q_{\leq N}))$ with the notation of Definition 2.7. We also define a linear mapping $\varphi_N : (K\{\mathbf{X}\}^\circ)^{Q_{\leq N}} \to K\{\mathbf{X}\}^\circ$ by

$$\varphi_N : (a_f)_{f \in Q_{\leq N}} \mapsto \sum_{f \in Q_{\leq N}} a_f v(f).$$

By definition, φ_N takes its values in the ideal generated by $v(Q_{\leq N})$ and $\varphi_N(S_N) \subset \pi K\{\mathbf{X}\}^\circ$.

Proposition 4.2. *For any integer N, the following holds:*
(a) The family $\rho(GBasis_{N+1})$ is a GB of $\bar{I}_N$.
(b) $\varphi_N(S_N) \subset \left\langle \pi \cdot v(Q_{\leq N}),\ \pi^{-N} Q_{>N} \right\rangle$.
(c) $I_{N+1} = \left\langle \pi^{N+1} \cdot v(Q_{\leq N+1}),\ Q_{>N+1} \right\rangle$.
(d) $\bar{I}_{N+1} = \left\langle \rho(Q_{\leq N+1}) \right\rangle$.

Proof. When $N < 0$, we have $S_N = 0$ and $I_{N+1} = I$, so that the proposition is obvious. We now consider a nonnegative integer N and assume that the proposition holds for $N-1$. By the induction hypothesis, we know that $\rho(GBasis_N)$ is a GB of $\bar{I}_{N-1}$. It then follows from Lemma 4.1 that $\rho(GBasis_{N+1})$ is a GB of the ideal generated by $\bar{I}_{N-1}$ and $\rho(Q_N)$, which is equal to $\bar{I}_N$ by the induction hypothesis. The assertion (a) is then proved.

Between the checkpoints τ_N and τ_{N+1}, each signature u added to S on line 14 corresponds to a family $(a_f)_{f \in Q_{\leq N}}$ for which the sum $\sum_f a_f f$ equals the element v_0 added to Q on the same line. Rescaling the a_f's, we cook up an element $z \in S_N$ with the property that $\varphi_N(z) = \pi^{-N} v_0$. Let $Z \subset S_N$ be the set of those elements. From Proposition 2.3 and Lemma 2.8, we derive that $\bar{S}_N$ is generated by $\bar{S}_{N-1}$ (viewed as a submodule of $\bar{S}_N$ by filling new coordinates with zeroes) and Z. Thus:

$$\begin{aligned} \varphi_N(S_N) &= \varphi_{N-1}(S_{N-1}) + \left\langle \varphi_N(Z),\ \pi \cdot v(Q_{\leq N}) \right\rangle \\ &\subset \varphi_{N-1}(S_{N-1}) + \left\langle \pi^{-N} Q_{>N},\ \pi \cdot v(Q_{\leq N}) \right\rangle. \end{aligned}$$

The assertion (b) now follows from the induction hypothesis, once we have observed that $Q_{>N-1} = \pi^N v(Q_N) \cup Q_{>N}$.

Let us now prove (c). Let $h \in I_{N+1}$. Then $h \in I_N$ and we can use the induction hypothesis to write

$$h = \pi^N \sum_{f \in Q_{\leq N}} a_f v(f) \ + \sum_{g \in Q_{>N}} b_g g$$

for some $a_f, b_g \in K\{\mathbf{X}\}^\circ$. Reducing modulo π^{N+1}, we find that the family $(a_f)_{f \in Q_{\leq N}}$ belongs to S_N. From (b), we deduce that $\sum_{f \in Q_{\leq N}} a_f v(f) \in \left\langle \pi \cdot v(Q_{\leq N}),\ \pi^{-N} Q_{>N} \right\rangle$. We then conclude by noticing that $Q_{>N} = \pi^{N+1} v(Q_{N+1}) \cup Q_{>N+1}$.

Finally, (d) follows from (c) by dividing by π^{N+1} and reducing modulo π. □

Termination. Since $k[\mathbf{X}]$ is noetherian, the sequence of ideals $(\bar{I}_N)$ is eventually constant. This implies that *GBasis* cannot grow indefinitely; in other words, the final value of *GBasis* is reached in finite time. However, the reader should be careful that this does not mean that Algorithm 2 terminates. Indeed, once the final value of *GBasis* has been computed, one still has to check that the remaining series in Q reduce to zero; this is achieved by performing divisions and can hang forever if we are working at infinite precision. Nevertheless, this misfeature seems very difficult to avoid since, when working at infinite precision, the input series contain themselves an infinite number of coefficients and any modification on one of them could have a strong influence on the final result.

Correctness. Let G be the output of Algorithm 2, that is the limit of the ultimately constant sequence $(GBasis_N)$. For a positive integer N, we define $G_{\leq N}$ as the set of $f \in G$ with $\mathrm{val}(f) \leq N$. Since only elements of valuation at least N+1 are added to *GBasis* after the checkpoint τ_{N+1}, we deduce that $G_{\leq N} = GBasis_{N+1}$. Hence, by Proposition 4.2, $\rho(G_{\leq N})$ is a GB of $\bar{I}_N$ for all $N \geq 0$. We are going to show that this sole property implies that G is indeed a GB of I. For this, we consider $f \in I$. We write $N = \mathrm{val}(f)$, so that $\rho(f)$ is the image in $k[\mathbf{X}]$ of $\pi^{-N} f$. Moreover, we know that $LM(\rho(f))$ is divisible by $LM(\rho(g))$ for some $g \in G_{\leq N}$, *i.e.* there exists $\mathbf{i} \in \mathbb{N}^n$ such that $LM(\rho(f)) = \mathbf{X}^{\mathbf{i}} \cdot LM(\rho(g))$. This readily implies that $LM(f) = \pi^{N-\mathrm{val}(g)} \cdot \mathbf{X}^{\mathbf{i}} \cdot LM(g)$, showing that $LM(g)$ divides $LM(f)$ in $\mathbb{T}\{\mathbf{X}\}^\circ$ given that $\mathrm{val}(g) \leq N$. We have then proved that the leading monomial of any element of I is divisible by some $LM(g)$ with $g \in G$, *i.e.* that G is a GB of I.

5 IMPLEMENTATION

We have implemented both the PoTe and VaPoTe algorithms in SageMath[3]. Our implementation includes the following optimization: at the end of the loop (*i.e.* after line 20), we minimize and reduce the current GB in construction. This operation is allowed since all signatures are discarded after each iteration of the loop. Similarly, we reduce each new series f popped from Q on line 4 before proceeding it. These ideas were explored in the algorithm F5-C [EP10] and, as mentioned before, were one of the main motivations for adopting an incremental point of view.

Our implementation is also able to compute GB of ideals in $K\{\mathbf{X}\}$. For this, we simply use a reduction (for no extra cost) to the case of $K\{\mathbf{X}\}^\circ$ (see [CVV19, Proposition 2.23]). We also normalize the signatures in S to be monic after each iteration of the main loop; in the PoTe algorithm, this renormalization gives a stronger cover criterion and thus improves the performances.

As mentioned in Section 4.1, Algorithm 2 remains correct if the reductions are interrupted as soon as the valuation rises. This can be done in the reduction step before processing the next f, before adding elements to the SGB, as well as in the inter-reduction step. Delaying reductions could be interesting, for instance, if the input ideal is saturated: indeed, in this case, the algorithm never considers elements with positive valuation and delayed reductions do not need to be done afterwards. On the other hand, performing more reductions earlier leads to shorter reducers and potentially faster reductions later. In practice, in our current implementation, we have observed all possible scenarios: interrupting the reductions can make the computation faster, slower, or not make any significant difference.

[3] https://trac.sagemath.org/ticket/28777

Table 1: Timings for the computation of GBs related to the torsion points on the Tate curve (all times in seconds)

Parameters		Buchberger	PoTe	VaPoTe
$p = 5$,	$\ell = 5$, prec = 12	87.9	72.2	19.2
$p = 11$,	$\ell = 5$, prec = 12	321	30.5	28.9
$p = 57637$,	$\ell = 5$, prec = 12	83.2	13.3	13.3
$p = 7$,	$\ell = 7$, prec = 9	62.3	45.3	27.7
$p = 11$,	$\ell = 7$, prec = 9	168	36.0	28.5

5.1 Some timings

Numerous experimentations on various random inputs show that the VaPoTe algorithm performs slightly better than the PoTe algorithm on average. Besides, both PoTe and VaPoTe algorithms usually perform much better than Buchberger algorithm, although we observed important variations depending on the input system.

As mentioned in the introduction, Tate algebras are the building blocks of p-adic geometry. One can then cook up interesting systems associated to meaningful geometrical situations. As a basic example, let us look at torsion points on elliptic curves.

We recall briefly that (a certain class of) elliptic curves over $K = \mathbb{Q}_p$ are in one-to-one correspondence with a parameter q lying in the open unit disc [Ta95]. The parametric equation of these curves is $y^2 + xy = x^3 + a_4(q)\, x + a_6(q)$ with:

$$a_4(q) = 5 \sum_{n=0}^{\infty} n^3 \frac{q^n}{1-q^n}, \quad a_6(q) = \sum_{n=0}^{\infty} \frac{7n^5 + 5n^3}{12} \frac{q^n}{1-q^n}.$$

In order to fit with the framework of this article, we only consider parameters q in the closed unit disc of radius $|p|$ and perform the change of variables $q = pt$. Given an auxiliary prime number ℓ, we consider the ℓ-th division polynomial $\Phi_\ell(x,t) \in K\{t\}^\circ[x]$ associated to the Weierstrass form of the above equation. By definition, its roots are the abscissas of ℓ-torsion points of the Tate curve. We now fix p and ℓ and consider the system in 3 variables $\Phi_\ell(x,t_1) = \Phi_\ell(x,t_2) = 0$. Its solutions parametrize the pairs of elliptic curves sharing a common ℓ-torsion point. Computing a GB of it then provides information about torsion points on p-adic elliptic curves. Related (but more sophisticated) computations are likely to appear in the study of the arithmetics of p-adic modular forms [Go88] or the development of p-adic analogues and refinements of Tate's isogeny Theorem [Ta66].

Table 1 shows the timings obtained for computing a GB of the above systems for different values of p, ℓ and different precisions. We clearly see on these examples that both PoTe and VaPoTe outperform the Buchberger algorithm.

5.2 Towards further improvements

Faster reductions. Observing how our algorithms behave, one immediately notices that reductions are very slow. It is not that surprising since our reduction algorithm is currently very naive. For this reason, we believe that several structural improvements are quite possible. An idea in this direction would be to store a well-chosen representative sample of reductions and reuse them later on.

Typically, we could cache the reductions of all terms of the form $x_1^{2^{e_1}} \cdots x_n^{2^{e_n}}$ (with respect to the current GB in construction) and use them to emulate a fast exponentation algorithm in the quotient ring $K\{\mathbf{X}\}^\circ/\langle GB\rangle$.

Another attractive idea for accelerating reduction is to incorporate Mora's reduction algorithm [Mo82, MRW17] in our framework. Let us recall that Mora's algorithm is a special method for reducing terms with respect to local or mixed orders (*i.e.* orders for which there exist terms $t < 1$), avoiding infinite loops in the reduction process. In our framework, infinite loops of reductions cannot arise since the computations are truncated at a given precision. Nevertheless, we believe that Mora's algorithm can still be used to short-circuit some reductions.

The situation for Tate terms is actually significantly simpler than that of general local orders. Indeed, Mora's reduction algorithm roughly amounts to add πr to our list of reductors each time we encounter a remainder r (including f itself) in the reduction process. We believe that this optimization, if it is carefully implemented, could already have some impact on the performances. Besides, observing that the equality $f = r + \pi q f$ also reads $f = (1 - \pi q)^{-1} r$, we realize that Mora reductions of a Tate series are somehow related to its Weierstrass decomposition. Moreover, at least in the univariate case, it is well known that Weierstrass decompositions can be efficiently computed using a well-suited Newton iteration. It could be interesting to figure out whether this strategy extends to multivariate series and, more generally, to the computation of arbitrary Mora reductions.

Using overconvergence properties. In a different direction, we would like to underline that the orderings we are working with are by design block orders (comparing first the valuation). However, in the classical setting, we all know that graded orders often lead to much more efficient algorithms. Unfortunately, in the setting of this article, the very first definition of a Tate series already forces us to give the priority to the valuation in the comparison of terms; otherwise, the leading term would not be defined in general.

Nonetheless, we emphasize that even if graded orders do not exist over $K\{\mathbf{X}\}$, they do exist over some subrings. Precisely, recall that, given a tuple $\mathbf{r} = (r_1, \dots, r_n)$, we have defined[4]:

$$K\{\mathbf{X};\mathbf{r}\} := \left\{ \sum_{\mathbf{i}\in\mathbb{N}^n} a_\mathbf{i}\mathbf{X}^\mathbf{i} \text{ s.t. } a_\mathbf{i} \in K \text{ and } \operatorname{val}(a_\mathbf{i}) - \mathbf{r}\cdot\mathbf{i} \xrightarrow[|\mathbf{i}|\to+\infty]{} +\infty \right\}$$

where $\mathbf{r}\cdot\mathbf{i}$ denotes the scalar product of the vectors $\mathbf{r}$ and $\mathbf{i}$. When the r_i's are all nonnegative, $K\{\mathbf{X};\mathbf{r}\}$ embeds naturally into $K\{\mathbf{X}\}$; precisely, elements in $K\{\mathbf{X};\mathbf{r}\}$ are those series that overconverges over the polydisk of polyradius $(|\pi|^{-r_1}, \dots, |\pi|^{-r_n})$. Moreover, the algebra $K\{\mathbf{X};\mathbf{r}\}$ is equipped with the valuation $\operatorname{val}_\mathbf{r}$ defined by:

$$\operatorname{val}_\mathbf{r}\left(\sum_{\mathbf{i}\in\mathbb{N}^n} a_\mathbf{i}\mathbf{X}^\mathbf{i} \right) = \min_{\mathbf{i}\in\mathbb{N}^n} \operatorname{val}(a_\mathbf{i}) - \mathbf{r}\cdot\mathbf{i}.$$

This valuation defines a new term ordering $\leq_\mathbf{r}$. We observe that, from the point of view of $K\{\mathbf{X}\}$, it really looks like a graded order: the quantity $\operatorname{val}_\mathbf{r}(f)$ plays the role of (the opposite of) a "total degree" which mixes the contribution of the valuation and that of the classical degree.

[4]We refer to [CVV19] for more details

In light of the above remarks, we formulate the following question. Suppose that we are given an ideal $I \subset K\{\mathbf{X}\}^\circ$ (say, of dimension 0) generated by some series $f_1, \dots, f_m$. If we have the promise that the f_i's all overconverge, *i.e.* all lie in $K\{\mathbf{X};\mathbf{r}\}$ for a given $\mathbf{r}$, can we imagine an algorithm that computes a GB of I taking advantage of the term ordering $\leq_\mathbf{r}$? As an extreme case, if we have the promise that all the f_i's are polynomials (that is $r_i = +\infty$ for all i), can one use this assumption to accelerate the computation of a GB of I?

REFERENCES

[BGR84] Bosch Siegfried, Günzter Ulrich and Remmert Reinhold, Non-Archimedean analysis, Springer-Verlag (1984)

[Bu65] Buchberger Bruno, Ein Algorithmus zum Auffinden der Basiselemente des Restklassenringes nach einem nulldimensionalen Polynomideal (An Algorithm for Finding the Basis Elements in the Residue Class Ring Modulo a Zero Dimensional Polynomial Ideal), English translation in J. of Symbolic Computation, Special Issue on Logic, Mathematics, and Computer Science: Interactions. Vol. 41, Number 3-4, Pages 475–511, 2006

[CL14] Caruso Xavier and Lubicz David, Linear Algebra over $\mathbb{Z}_p[[u]]$ and related rings, LMS J. Comput. Math. 17 (2014), 302-344

[CVV19] Caruso Xavier, Vaccon Tristan and Verron Thibaut, Gröbner bases over Tate algebras, in Proceedings: ISSAC'19.

[EF17] Eder Christian and Faugère Jean-Charles, A survey on signature-based algorithms for computing Gröbner bases, J. of Symbolic Computation, 2017

[EP10] Eder Christian and Perry John, F5C: A variant of Faugère's F5 algorithm with reduced Gröbner bases, J. of Symbolic Computation, 2010

[Fa99] Faugère Jean-Charles, A new efficient algorithm for computing Gröbner bases (F4), Journal of Pure and Applied Algebra, 1999

[Fa02] Faugère, Jean-Charles, A new efficient algorithm for computing Gröbner bases without reduction to zero (F5), in Proceedings: ISSAC'02.

[FP04] Fresnel Jean and van der Put Marius, Rigid analytic geometry and its applications, Birkhäuser, 2004

[GGV10] Gao Shuhong, Guan Yinhua and Volny IV Frank, A new incremental algorithm for computing Groebner bases, In Proceedings: ISSAC'10.

[GVW16] Gao Shuhong, Volny IV Frank, and Wang Mingsheng, A new framework for computing Gröbner bases, Mathematics of computation, 2016, vol. 85, no 297, p. 449-465.

[Go88] Gouvea Fernando, Arithmetic of p-adic Modular Forms, Lecture Notes in Mathematics **1304**, Springer-Verlag, 1988

[GR95] Gräbe Hans-Gert, Algorithms in Local Algebra, J. of Symbolic Computation **19**, 1995, 545–557

[LS07] Le Stum Bernard, Rigid Cohomology, Cambridge tracts in mathematics **172**, Cambridge University Press, 2007

[L+18] Lu Dong, Wang Dingkang, Xiao Fanghiu, Zhou Jie, Extending the GVW Algorithm to Local Ring, Proceedings of 43th International Symposium on Symbolic and Algebraic Computation, ISSAC'18, New York, USA

[Macaulay2] Grayson Daniel R. and Stillman Michael, Macaulay2, a software system for research in algebraic geometry, available at https://faculty.math.illinois.edu/Macaulay2/

[Magma] Bosma Wieb, Cannon John, and Playoust Catherine, The Magma algebra system. I. The user language, J. Symbolic Comput. **24**, 1997, 235–265

[MRW17] Markwig Thomas, Ren Yue and Wienand Olivier, Standard bases in mixed power series and polynomial rings over rings, J. of Symbolic Computation **79**, 2017, 119–139

[Mo82] Mora Ferdinando, An algorithm to compute the equations of tangent cones, Proceedings of European Computer Algebra Conference in Marseille, 1982, 158–165

[NS01] Norton Graham H. and Sălăgean Ana, Strong Grobner bases and cyclic codes over a finite-chain ring, Electronic notes in discrete maths **6**, 2001, 240–250

[Sage] SageMath, the Sage Mathematics Software System (Version 8.6), The Sage Development Team, 2018, http://www.sagemath.org

[Singular] Decker Wolfram, Greuel Gert-Martin., Pfister Gerhard and Schönemann Hans, SINGULAR 4-1-2 — A computer algebra system for polynomial computations, http://www.singular.uni-kl.de, 2019

[Ta66] Tate John, Endomorphisms of abelian varieties over finite fields, Inventiones Mathematicae **2**, 1966, 134–144

[Ta71] Tate John, Rigid analytic spaces, Inventiones Mathematicae **12**, 1971, 257–289

[Ta95] Tate John, A review of non-Archimedean elliptic functions, in Elliptic curves, modular forms and Fermat's last theorem, Series in Number Theory, Int. Press, Cambridge, MA, 1995, 162–184

Syzygies of Ideals of Polynomial Rings over Principal Ideal Domains

Hara Charalambous
Dep. of Mathematics, Aristotle University of Thessaloniki
Thessaloniki, Greece
hara@math.auth.gr

Kostas Karagiannis
Dep. of Mathematics, Aristotle University of Thessaloniki
Thessaloniki, Greece
kkaragia@math.auth.gr

Sotiris Karanikolopoulos
Dep. of Mathematics, National and Kapodistrian
University of Athens
Athens, Greece
sotiriskaran@gmail.com

Aristides Kontogeorgis
Dep. of Mathematics, National and Kapodistrian
University of Athens
Athens, Greece
kontogar@math.uoa.gr

ABSTRACT

We study computational aspects of syzygies of graded modules over polynomial rings $R[w_1, \ldots, w_g]$ when the base R is a discrete valuation ring. In particular, we use the torsion of their syzygies over R to provide a formula which describes the behavior of the Betti numbers when changing the base to the residue field or the fraction field of R. Our work is motivated by the deformation theory of curves.

CCS CONCEPTS

• **Theory of computation → Computational geometry**; • **Mathematics of computing**;

KEYWORDS

Commutive algebra, syzygies, principal ideal domains, reduction, lifting MSC:13D02, 13P20

ACM Reference Format:
Hara Charalambous, Kostas Karagiannis, Sotiris Karanikolopoulos, and Aristides Kontogeorgis. 2020. Syzygies of Ideals of Polynomial Rings over Principal Ideal Domains. In *International Symposium on Symbolic and Algebraic Computation (ISSAC '20), July 20–23, 2020, Kalamata, Greece.* ACM, New York, NY, USA, 5 pages. https://doi.org/10.1145/3373207.3404046

1 INTRODUCTION

The study of syzygies of modules is one of the main topics of interest of combinatorial commutative algebra with numerous algorithmic applications. In the context of computational algebraic geometry, usually one studies syzygies of ideals of the polynomial ring $k[w_1, \ldots, w_g]$, where k is a field. However, deformation theory of curves deals with flat families of curves over discrete valuation rings R. In particular, non-hyperelliptic curves of genus g are better understood in terms of their canonical ideal, the ideal in $S = R[w_1, \ldots, w_g]$ that defines the canonical embedding of the family in $\mathbb{P}_R^{g-1}$.

The last two authors have studied the R-modules of relative polydifferentials for certain cyclic covers of the projective line [7], which lead to a description of the relative canonical ideal by the first, second and fourth author in [3]. Applications of syzygies and free resolutions to the study of curves with automorphisms are given by Terezakis, Tsouknidas and the fourth author in [8]. The difference in the behaviour of the Betti numbers in the special and generic fibre is expected to provide new obstructions to the theory of lifting of curves with automorphisms, see [9], [10], since liftings of indecomposable representations of the automorphism group should respect the free and torsion part. Moreover the relative point of view contributes to the understanding of the situation concerning Green's conjecture in positive characteristic, see [2] for a refined version.

Let R be a discrete valuation ring with maximal ideal $\mathfrak{m}_R = \langle x \rangle$, fraction field K and residue field k. Deformation theory and modular representation theory are related to the effect of taking the base to be any of the three rings R, K, k. Let $S = R[w_1, \ldots, w_g]$ and consider an S-module M such that the generator x of $\mathfrak{m}_R$ is not a zero divisor on M. This leads to the study a) of $\widehat{S} = K[w_1, \ldots, w_g]$ and the respective $\widehat{S}$-module $\widehat{M} = M \otimes \widehat{S}$ (corresponding to the *generic fibre*) and b) of $\overline{S} = k[w_1, \ldots, w_g]$ and the respective $\overline{S}$-module $\overline{M} = M \otimes \overline{S}$ (corresponding to the *special fibre*).

Grothendieck's relative point of view leads to the question of how the syzygies and the Betti numbers of the special and the generic fibre of a family are related when considered over k or K or even over R. The study of syzygies becomes automatically more challenging over R since the non-zero elements of the PID may not be invertible and modules might have torsion, see [1, chap. 4] for a more comprehensive account and also [12]. On the other hand simplicial homology over $\mathbb{Z}$ has been extensively studied and techniques have been developed to account for that case and the different behavior over $\mathbb{Q}$, [4]. It is well known that, even in the case of monomial ideals, the minimal free resolution depends on the characteristic of the ground field, the classical example being the triangulation of the projective plane.

Example 1. The Betti numbers of

$$B = \langle abc, abf, ace, ahe, ahf, bch, bhe, bef, chf, cef \rangle \lhd k[a, b, c, e, f, h]$$

ISSAC '20, July 20–23, 2020, Kalamata, Greece

ACM ISBN 978-1-4503-7100-1/20/07…$15.00
https://doi.org/10.1145/3373207.3404046

differ when $\mathrm{char}(k) = 0$ (table on the left) and $\mathrm{char}(k) = 2$ (table on the right).

	0	1	2	3
0	1	0	0	0
1	0	0	0	0
2	0	10	15	6
3	0	0	0	0

	0	1	2	3	4
0	1	0	0	0	0
1	0	0	0	0	0
2	0	10	15	6	1
3	0	0	0	1	0

Thus when $\mathrm{char}(k) = 2$, the ideal B has a third and a fourth graded syzygy of degree 6 which do not appear over characteristic zero (or any other characteristic $p \neq 2$ for that matter), see also [11, Ex. 12.4].

In the sequel we give explicit reasons for this behavior. We consider syzygies of finitely generated graded S-modules of the polynomial ring $S = R[w_1, \ldots, w_g]$. We will see that it makes sense to consider minimal free resolutions of the graded S-module M, and we will define the graded *Betti* numbers of M.

The structure of this paper is as follows: we first discuss minimal free resolutions of graded modules over $R[w_1, \ldots, w_g]$ and Nakayama's lemma (Lemma 2). Using the classification theorem for modules over PIDs, we see how the existence of R-torsion on the syzygies affects the resolution (Theorem 6).

We also explain how torsion can be read from the Smith normal form of the reduced matrix of the differentials (Corollary 7). In the last section we give a detailed computation of Example 1 and conclude with a method (Algorithm 1) which, given the generators of a graded ideal I of $\mathbb{Z}[w_1, \ldots, w_g]$, outputs all primes p for which the Betti numbers of I in $\mathbb{F}_p[w_1, \ldots, w_g]$ differ from the Betti numbers of I in $\mathbb{Q}[w_1, \ldots, w_g]$ and give information for possible obstructions.

2 SYZYGIES OVER GENERAL RINGS

Let $(R, \mathfrak{m}_R)$ be as in the introduction, and let $S = R[w_1, \ldots, w_g]$ be the polynomial ring in g variables, graded by assigning the degree 1 to each w_i, $i = 1, \ldots, g$. Thus $S = \sum_{i \geq 0} S_i$, with $S_0 = R$. We let $\mathfrak{m}$ and $\mathfrak{m}_S$ be respectively the prime and maximal ideals $\mathfrak{m} = \langle w_1, \ldots, w_g \rangle$, $\mathfrak{m}_S = \mathfrak{m} + \mathfrak{m}_R S$ of S. Observe that $k = S/\mathfrak{m}_S = R/\mathfrak{m}_R$.

Let M be a finitely generated graded S-module. Thus we have $M = \sum_{i \geq a} M_i$, where $S_j M_i \subset M_{i+j}$ and in particular M_i is an R-module, for $i \geq a$. Let $m_1, \ldots, m_n$ form a generating set of M. It is clear that $\{\overline{m}_i = m_i + \mathfrak{m}_S M,\ i = 1, \ldots, n\}$ is a generating set of $M/\mathfrak{m}_S M$. The converse, i.e. Nakayama's lemma, holds when the elements m_i are homogeneous. The proof follows the same lines as the standard proof for the graded case, [5, lemma 1.4]. We include it here for completeness of the exposition.

Lemma 2 (Nakayama). *Let M be a finitely generated positively graded S-module, $m_1, \ldots, m_n \in M$ homogeneous so that $\overline{m}_i$, $i = 1, \ldots, n$ generate $M/\mathfrak{m}_S M$. Then $m_1, \ldots, m_n$ generate M.*

Proof. Let $M' = \sum_{i=1}^{n} S m_i$ and consider the finitely generated graded S-module $N = M/M'$. By our assumption on the m_i, $M' + \mathfrak{m}_S M = M$, thus $N/\mathfrak{m}_S N = 0$ and $\mathfrak{m}_S N = N$. If $N \neq 0$, there is a nonzero graded element of least degree in N. Since $\mathfrak{m}_S N = N$, this element must have degree zero. It follows that $N_0 = \mathfrak{m}_R N_0$. Since R is a local PID, Nakayama's lemma in the local case gives that $N_0 = 0$. It follows that $N = 0$ as desired. □

It follows that the least number of homogeneous elements needed to generate M is the dimension of the $S/\mathfrak{m}_S$-vector space $M/\mathfrak{m}_S M$. We proceed to construct a **minimal graded** free resolution of M. Let $m_1, \ldots, m_n$ be a minimal set of homogeneous generators of M. We let F_0 be the free module $F_0 = \bigoplus_i S e_i$ on generators e_i, $\deg(e_i) = \deg(m_i)$ $(i = 1, \ldots n)$ and let $\pi_0 : F_0 \longrightarrow M$ be the epimorphism determined by $\pi_0(e_i) = m_i$. This gives the short exact sequence

$$0 \to \ker \pi_0 \xrightarrow{\iota_0} \mathrm{F}_0 \xrightarrow{\pi_0} \mathrm{M} \to 0,$$

where $\ker\pi_0 \subset \mathfrak{m}_S F_0$. Since $\ker\pi_0$ is a finitely generated graded S-module we repeat this procedure to obtain $\pi_1 : F_1 \longrightarrow \ker \pi_0$ and $\delta_1 : F_1 \longrightarrow F_0$ as the composition $F_1 \xrightarrow{\pi_1} \ker \pi_0 \xrightarrow{\iota_0} F_0$. Note that $\delta_1(F_1) = \ker \pi_0$ and that we have specified the basis of F_1 that maps to a minimal homogeneous generating set of $\ker \pi_0$. Iterating this procedure, we obtain a free graded resolution of M which is minimal since by construction $\ker\pi_i \subset \mathfrak{m}_S F_i$ for all $i \geq 0$:

$$(F_\bullet, \delta_\bullet) : \cdots \longrightarrow F_1 \xrightarrow{\delta_1} F_0 \xrightarrow{\pi_0} M \longrightarrow 0$$

For each $i \geq 1$, the resolution above breaks into a short exact sequence

$$0 \to \ker \pi_{\mathrm{i}} \xrightarrow{\iota_{\mathrm{i}}} \mathrm{F}_{\mathrm{i}} \xrightarrow{\pi_{\mathrm{i}}} \ker \pi_{\mathrm{i-1}} \to 0, \tag{1}$$

$\delta_i : F_i \to F_{i-1}$ being the composition $\iota_{i-1}\pi_i$ for $i \geq 1$. We note that the differentials δ_i are of degree zero, $\delta_i(F_i) \subset \mathfrak{m}_S F_{i-1}$ and that δ_i maps a basis of F_i to a minimal set of homogeneous generators of $\delta_i(F_i)$, as in [5, Corollary 1.5]. We write each F_i as a direct sum, indexed by $\mathbb{Z}$, of copies of S shifted by the degrees of the generators:

$$F_i = \bigoplus_{j \in \mathbb{Z}} S(-j)^{\beta_{i,j}},$$

where finitely many of the $\beta_{i,j}$ are nonzero. The exponent $\beta_{i,j} \in \mathbb{N}$ that counts the number of minimal generators of degree j in F_i is called the (i, j)-**graded Betti number** of M and equals the dimension $\dim_k \mathrm{Tor}_i^S(k, M)_j$ as in [5, Corollary 1.7]. We write $\beta_{i,j}(M)$ when needed to emphasize the module M. The S-modules

$$\Pi_i = \ker \pi_{i-1} = \ker \delta_{i-1}, \tag{2}$$

for $i \geq 1$, are known as the i-th **syzygies** of M, and we set $\Pi_0 = M$, so that Π_i is a graded S-module for all i. By successively taking homology of the short exact sequences in (1) we get that

$$\mathrm{Tor}_1^S(\Pi_{i-1}, R) = \mathrm{Tor}_i^S(M, R),\ i \geq 1. \tag{3}$$

We let $F_{i,j}$ be the direct summand of F_i at degree j and denote by $\delta_{i,j}$ the restriction

$$\delta_{i,j} = \delta_i|_{F_{i,j}} : F_{i,j} \longrightarrow F_{i-1,j}.$$

Remark 3. Let $F_\bullet$ be a minimal graded free resolution of the graded S-module M. By tensoring $F_\bullet$ with $R = S/\mathfrak{m}$ (over S) we obtain a graded complex of R-modules:

$$F_\bullet \otimes R : \cdots \longrightarrow F_1 \otimes R \xrightarrow{\delta_1 \otimes 1_R} F_0 \otimes R \longrightarrow M \otimes R \longrightarrow 0$$

whose homology at the i-th position is $\mathrm{Tor}_i^S(R, M)$, a graded S-module. Note that, for $\alpha > 0$, $S(-\alpha) \otimes R$ only lives in degree α. Thus, to compute $\mathrm{Tor}_i^S(R, M)_j = \mathrm{Tor}_1^S(R, \Pi_{i-1})_j$ one needs to consider the following s.e.s. derived from (1):

$$0 \to \mathrm{Tor}_1^S(\Pi_i, R)_j \to (\Pi_{i+1})_j \otimes R \to (S(-j)^{\beta_{i,j}} \otimes R)_j \to (\Pi_i)_j \otimes R \to 0. \tag{4}$$

3 SYZYGIES OVER GENERIC AND SPECIAL FIBRES

Let us start by recalling the notation set-up in the introduction: R is a discrete valuation ring with maximal ideal $\mathfrak{m}_R = \langle x \rangle$, K is the fraction field of R and $k = R/\mathfrak{m}_R$ is the residue field. We set $S = R[w_1, \ldots, w_g]$, $\mathfrak{m} = \langle w_1, \ldots, w_g \rangle \lhd S$ and $\mathfrak{m}_S = \langle x, w_1, \ldots, w_g \rangle$. Let M be a finitely generated graded S-module. Note that $\widehat{S} = K[w_1, \ldots, w_g]$ is the localization of S at the multiplicatively closed subset R^* and similarly for $\widehat{M} = M \otimes \widehat{S}$. Finally, $\overline{\mathfrak{m}} = \langle w_1, \ldots, w_g \rangle \lhd \overline{S}$ is the maximal graded ideal of $\overline{S} = k[w_1, \ldots, w_g]$ and $\overline{M} = M/xM$.

We note that if N is a finitely generated R-module, then since R is a local PID, N is a direct sum of the form

$$N = \bigoplus^{\mathrm{rk}(N)} R \oplus \mathrm{tor}(N),$$

where $\mathrm{rk}(N)$ is the rank of N as an R-module, while $\mathrm{tor}(N)$, the torsion part of N, is a direct sum of the form

$$\mathrm{tor}(N) = \bigoplus_{\nu=1}^{t(N)} R/Rx^{a(\nu,N)}, \text{ where } a(\nu, N) \in \mathbb{N}, \text{ for } \nu = 1, \ldots, t(N).$$

Observe that $\mathrm{tor}(N)$ is still visible when tensoring with k (special fibre), since $N \otimes_R k = k^{\mathrm{rk}(N)+t(N)}$, while it disappears when tensoring with K (generic fibre), since $N \otimes_R K = K^{\mathrm{rk}(N)}$.

Let M be a finitely generated graded S-module such that x is a not a zero divisor on M, i.e. multipication by x is injective and M is a flat R-module. Let $F_\bullet$ be a minimal free resolution of M. To study M we will tensor $F_\bullet$ with R and get a complex of R-modules.

It is known that under our assumptions, reduction to the special fibre preserves exactness, see for example [11, Thm 20.3]. The short proof is included here for completeness of the exposition.

Lemma 4. *If $F_\bullet$ is a free resolution of M as an S-module, then $F_\bullet \otimes S/xS$ is a free resolution of M/xM as an S/xS-module.*

Proof. The short exact sequences $0 \to S \to S \to S/xS \to 0$ and $0 \to M \to M \to M/xM \to 0$, imply that $\mathrm{Tor}_i^S(M, S/xS) = 0$, for $i \geq 1$ and thus $F_\bullet \otimes S/xS$ is exact. □

We also note that flatness of K over R implies flatness of the ring $K[w_1, \ldots, w_g]$ over S. Thus we have the following:

Lemma 5. *If $F_\bullet$ is a free resolution of M, seen as an S-module then $\widehat{F}_\bullet = F_\bullet \otimes K[w_1, \ldots, w_g]$ is a free resolution of $\widehat{M} = M \otimes K[w_1, \ldots, w_g]$ as an $K[w_1, \ldots, w_g]$-module.*

Let $(F_\bullet, \delta_\bullet)$ be a minimal graded free resolution of the graded S-module M. By Lemma 5, $\widehat{F}_\bullet$ is a graded free resolution of $\widehat{M}$, however it might not be minimal. We write $\Pi_{i,j}$ for the j-th graded piece of $\Pi_i = \ker(\delta_{i-1})$ and $\overline{\Pi}_{i,j}$ for the R-module $\Pi_{i,j} \otimes R$ which we decompose into its cyclic R-components. We will see that the quantities $f_{i,j} := \mathrm{rk}(\overline{\Pi}_{i,j})$, $t_{i,j} := t(\overline{\Pi}_{i,j})$ and $s_{i,j} = \mathrm{rk}(\mathrm{Tor}_1^S(R, \Pi_i)_j)$ are critical when we measure the difference between the graded Betti numbers of the generic and the special fibre.

Theorem 6. *Let S be $R[w_1, \ldots, w_g]$, M be a finitely generated graded S-module which is flat as an R-module, Π_i be the i-th syzygy of M and $t_{i,j}$ be the number of nonzero cyclic summands of $\overline{\Pi}_{i,j}$, for $i \geq 0$.*

(1) $\beta_{i,j}(M) = \beta_{i,j}(\overline{M})$, for $i \geq 0$.
(2) $\beta_{i,j}(M) = \beta_{i,j}(\widehat{M}) + t_{i,j} + t_{i-1,j}$ for $i \geq 1$.

Proof. Let $F_\bullet$ be a minimal graded free resolution of the graded S-module M. By Lemma 4, it follows that $\overline{F}_\bullet = F_\bullet \otimes S/xS$ is a free resolution of $\overline{M}$. Moreover, since $\delta_i(F_i) \subset \mathfrak{m}_S F_{i-1}$, it follows that $\overline{\delta_i}(\overline{F_i}) \subset \overline{\mathfrak{m}}\overline{F}_{i-1}$ and $\overline{F}_\bullet$ is a minimal free resolution of $\overline{M}$. Thus $\beta_{i,j}(M) = \beta_{i,j}(\overline{M})$.

For the generic fibre, by Lemma 5, $\widehat{F} = F_\bullet \otimes K[w_1, \ldots, w_g]$ is a free resolution of $\widehat{M}$ and we need to compute $\dim_K \mathrm{Tor}_i^{\widehat{S}}(\widehat{M}, K)_j$. By the Künneth formula [13, Th. 3.6.1], $\mathrm{Tor}_i^{\widehat{S}}(\widehat{M}, K)$ is the localization of $\mathrm{Tor}_i^S(M, R)$ at R^* and

$$\mathrm{Tor}_i^{\widehat{S}}(\widehat{M}, K)_j \cong \mathrm{Tor}_i^S(M, R)_j \otimes K.$$

Thus by (3) it suffices to examine the R-structure of $\mathrm{Tor}_1^S(\Pi_{i-1}, R)_j$. We consider the tensor product $F_\bullet \otimes R$. By (4) we have

$$0 \to \mathrm{Tor}_1^S(\Pi_{i-1}, R)_j \to \overline{\Pi}_{i,j} \longrightarrow R^{\beta_{i-1,j}} \longrightarrow \overline{\Pi}_{i-1,j} \to 0.$$

Since $\overline{\Pi}_{i,j}/\mathrm{Tor}_1^S(\Pi_{i-1}, R)_j \hookrightarrow R^{\beta_{i-1,j}}$, it follows that the quotient $\overline{\Pi}_{i,j}/\mathrm{Tor}_1^S(\Pi_{i-1}, R)_j$ is free and

$$\mathrm{tor}(\mathrm{Tor}_1^S(\Pi_{i-1}, R)_j) = \mathrm{tor}(\overline{\Pi}_{i,j}).$$

Thus

$$\overline{\Pi}_{i,j}/\mathrm{Tor}_1^S(\Pi_{i-1}, R)_j = R^{f_{i,j}-s_{i-1,j}}.$$

By the short exact sequence

$$\begin{array}{ccccccc} 0 & \to & \overline{\Pi}_{i,j}/\mathrm{Tor}_1^S(\Pi_{i-1}, R)_j & \to & R^{\beta_{i-1,j}} & \longrightarrow & \overline{\Pi}_{i-1,j} \longrightarrow 0 \\ & & \| & & & & \| \\ & & R^{f_{i,j}-s_{i-1,j}} & & & & R^{f_{i-1,j}} \oplus \mathrm{tor}(\overline{\Pi}_{i-1,j}) \end{array}$$

we have that

- $\beta_{i-1,j} = f_{i-1,j} + t_{i-1,j}$ (from the epimorphism),
- $\beta_{i-1,j} = (f_{i,j} - s_{i-1,j}) + f_{i-1,j}$ (from the additivity of ranks).

It follows that the rank $s_{i-1,j}$ of $\mathrm{Tor}_1^S(\Pi_{i-1}, R)_j$ is equal to

$$s_{i-1,j} = (f_{i,j} + f_{i-1,j}) - \beta_{i-1,j} = f_{i,j} + (f_{i-1,j} - \beta_{i-1,j}) = (\beta_{i,j} - t_{i,j}) - t_{i-1,j}.$$

We tensor $\mathrm{Tor}_i^S(R, M)_j$ with K to obtain that

$$\beta_{i,j}(\widehat{M}) = s_{i-1,j} = \beta_{i,j}(M) - t_{i,j} - t_{i-1,j}.$$

□

How does one compute $t_{i,j}$? This can be done by computing the Smith normal form of the matrix of differentials $\overline{\delta}_{i,j} = \delta_{i,j} \otimes R$. We proceed as in [4]. Note that $R^{\beta_{i,j}} = F_{i,j} \otimes R$, while $R^{\beta_{i-1,j}} = F_{i-1,j} \otimes R$ and $\overline{\delta}_{i,j} : R^{\beta_{i,j}} \longrightarrow R^{\beta_{i-1,j}}$. Let $B_{i,j}$ be the matrix of $\overline{\delta}_{i,j}$ with respect to the canonical bases of $R^{\beta_{i,j}}$ and $R^{\beta_{i-1,j}}$. There is a change of basis for $R^{\beta_{i,j}}$ and $R^{\beta_{i-1,j}}$ so that the matrix of $\overline{\delta}_{i,j}$ with respect to these new bases is the *Smith normal form* of $B_{i,j}$, say $A_{i,j}$. The Smith normal form $A_{i,j}$ contains an upper left diagonal block

$$\mathrm{diag}\left(b_1, b_2, \ldots, b_{t(i-1,j)}\right),$$

with $b_1 | b_2 | \cdots | b_{t(i-1,j)} \neq 0$, while the rest of the blocks of $A_{i,j}$ are zero. We note that since $F_\bullet$ is a minimal resolution, all $b_a \in \mathfrak{m}_R$, for $a = 1, \ldots, t(i-1, j)$ and thus $b_a = x^{e(a)}$, for some positive integer

$e(a)$. It is clear that $t(i-1,j)$ is the rank of $\operatorname{Im}\overline{\delta}_{i,j}$ and thus the rank of $\ker\overline{\delta}_{i,j}$ equals $\beta_{i,j}-t(i-1,j)$. Let us now consider the Smith normal form for $\overline{\delta}_{i+1,j}$. Suppose that its nonzero block is

$$\operatorname{diag}\left(c_1,c_2,\ldots,c_{t(i,j)}\right)$$

and let $\epsilon_1,\ldots,\epsilon_{\beta_{i,j}}$ be the basis of $R^{\beta_{i,j}}$ relative to this normal form. Thus $c_a\epsilon_a\in\operatorname{Im}\overline{\delta}_{i+1,j}$. Since $\overline{\delta}_{i,j}\overline{\delta}_{i+1,j}=0$, we have that $\overline{\delta}_{i,j}(c_a\epsilon_a)=c_a\overline{\delta}_{i,j}(\epsilon_a)=0$, and we conclude that $\epsilon_1,\ldots,\epsilon_{t(i,j)}$ are in $\ker\overline{\delta}_{i,j}$, for $a=1,\ldots,t(i,j)$. Thus,

$$\operatorname{Tor}_i^S(M,R)_j\cong R^{\beta_{i,j}-t(i-1,j)-t(i,j)}\oplus R/c_1R\oplus\cdots\oplus R/c_{t(i,j)}R.$$

By the uniqueness of the decomposition of $\operatorname{Tor}_i^S(M,R)_j$ and induction on i, it follows that $t(i,j)=t_{i,j}$ for all i. We have shown the following:

Corollary 7. *If $(F_\bullet,\delta_\bullet)$ is a minimal graded free resolution of M over S and the Smith normal form of the matrix of $\overline{\delta}_{a,j}$ has rank $t(a-1,j)$, $a\geq 1$, then $t_{i,j}=t(i,j)$ for $i\geq 0$ and $\beta_{i,j}(\widehat{M})=\beta_{i,j}(M)-t_{i,j}-t_{i-1,j}$.*

4 EXAMPLE

Let us now return to Example 1. Let $\mathbb{Z}_2$ be the ring of 2-adic integers with fraction field $\mathbb{Q}_2$ and residue field $\mathbb{F}_2$. Let $S=\mathbb{Z}_2[a,\ldots,h]$, $\mathfrak{m}=\langle a,\ldots,h\rangle$, $B=\langle abc,abf,ace,ahe,ahf,bch,bhe,bef,chf,cef\rangle$ and $M=S/B$. We will show that

$$\beta_{0,0}(M)=1,\beta_{1,3}(M)=10,$$

$$\beta_{2,4}(M)=15,\beta_{3,5}(M)=6,\beta_{3,6}(M)=1,\beta_{4,6}(M)=1,$$

and that M has a minimal graded free resolution over S of the form

$$\begin{array}{ccccccccc} & & F_4 & & F_3 & & F_2 & & F_1 \\ & & \| & & \| & & \| & & \| \\ 0 & \longrightarrow & S(-6) & \xrightarrow{\delta_4} & S(-5)^6\oplus S(-6) & \xrightarrow{\delta_3} & S(-4)^{15} & \xrightarrow{\delta_2} & S(-3)^{10} \end{array}$$

$$\xrightarrow{\delta_1} S=F_0\longrightarrow M\longrightarrow 0 \tag{5}$$

We will show that Π_4, the kernel of $\delta_3:F_3\to F_2$, has a minimal generating set of two elements, with the generator of degree 6 becoming torsion in $F_\bullet\otimes_S R\cong F_\bullet\otimes S/\mathfrak{m}$, impling that $t_{4,6}=1$. This means that for $\widehat{S}=\mathbb{Q}_2[a,\ldots,h]$, we get the following exact diagram:

$$\begin{array}{ccccccccc} 0 & \longrightarrow & \widehat{S}(-6) & \xrightarrow{\cong} & \widehat{S}(-6) & & \longrightarrow & & 0 \\ & & \downarrow & & \downarrow & & & & \\ 0 & \longrightarrow & \widehat{S}(-6) & \xrightarrow{\delta_4} & \widehat{S}(-5)^6\oplus\widehat{S}(-6) & \xrightarrow{\delta_3} & \widehat{S}(-4)^{15} & & \\ & & & & & & \xrightarrow{\delta_2} & & \\ & & \widehat{S}(-3)^{10} & \xrightarrow{\delta_1} & \widehat{S} & & \longrightarrow & \widehat{M} & \longrightarrow 0 \end{array}$$

The degree 6 elements in both F_4,F_3 have to be removed in order to obtain a minimal free resolution in the generic fibre.

We used Macaulay2 [6] in order to compute the above resolution. The following code

```
T = ZZ[a,b,c,e,f,h]
J = ideal(a*b*c,a*b*f,a*c*e,a*h*e,a*h*f,
     b*c*h,b*h*e,b*e*f,c*h*f,c*e*f)
rs = res J
rs.dd
```

produces the free resolution $G_\bullet$ of T/J over T

$$0\longrightarrow T^2\xrightarrow{\theta_4}T^{10}\xrightarrow{\theta_3}T^{17}\xrightarrow{\theta_2}T^{10}\xrightarrow{\theta_1}T, \tag{6}$$

where the differentials θ_3,θ_4 correspond to the matrices (also denoted for simplicity by θ_3,θ_4)

$$\theta_4=\begin{pmatrix} 0 & f\\ e & 0\\ -b & 0\\ -h & 0\\ 0 & -c\\ -c & 0\\ 0 & a\\ a & 0\\ \boxed{-1} & \boxed{1}\\ \boxed{-1} & \boxed{-1}\end{pmatrix}$$

$$\theta_3=\begin{pmatrix} 0 & -h & 0 & e & 0 & 0 & 0 & 0 & -eh & -eh\\ -h & 0 & 0 & -f & 0 & 0 & 0 & 0 & fh & 0\\ -b & 0 & -f & 0 & 0 & 0 & 0 & 0 & bf & 0\\ 0 & 0 & -c & 0 & 0 & b & 0 & 0 & 0 & 0\\ 0 & -c & 0 & 0 & 0 & -e & 0 & 0 & 0 & 0\\ e & f & 0 & 0 & 0 & 0 & 0 & 0 & 0 & ef\\ a & 0 & 0 & 0 & 0 & 0 & -f & 0 & 0 & 0\\ -c & 0 & 0 & 0 & -f & 0 & 0 & 0 & 0 & 0\\ 0 & 0 & 0 & 0 & a & 0 & c & 0 & 0 & 0\\ 0 & 0 & 0 & c & h & 0 & 0 & 0 & 0 & -ch\\ 0 & a & 0 & 0 & 0 & 0 & 0 & -e & 0 & 0\\ 0 & -b & -e & 0 & 0 & 0 & 0 & 0 & 0 & 0\\ 0 & 0 & a & 0 & 0 & 0 & 0 & b & 0 & 0\\ 0 & 0 & 0 & a & 0 & 0 & 0 & h & 0 & 0\\ 0 & 0 & h & -b & 0 & 0 & 0 & 0 & 0 & 0\\ 0 & 0 & 0 & 0 & \boxed{-1} & \boxed{1} & 0 & 0 & -c & 0\\ 0 & 0 & 0 & 0 & 0 & 0 & \boxed{-1} & \boxed{-1} & 0 & -a\end{pmatrix}$$

The matrix of θ_4 is reduced modulo $\langle a,\ldots,h\rangle$ to a 10×2 matrix, which is zero in all entries except for the lower 2×2 submatrix. We see that

$$\begin{pmatrix}-1 & 1\\ -1 & -1\end{pmatrix}=\begin{pmatrix}1 & 1\\ 0 & 1\end{pmatrix}\begin{pmatrix}2 & 0\\ 0 & 1\end{pmatrix}\begin{pmatrix}0 & 1\\ -1 & -1\end{pmatrix}.$$

A similar computation shows that the reduction of the matrix of θ_3 modulo $\langle a,\ldots,h\rangle$ has a Smith normal form whose nonzero diagonal block is the two by two identity matrix. Thus, through a series of base changes $G_\bullet\otimes\mathbb{Z}_2[a,\ldots,h]$ breaks into

$$\begin{array}{ccccccccccc} 0 & \longrightarrow & S & \xrightarrow{\cong} & S & \longrightarrow & 0 & & & & \\ & & \oplus & & \oplus & & & & & & \\ 0 & \longrightarrow & S & \xrightarrow{\delta_4} & S^7 & \xrightarrow{\delta_3} & S^{15} & \xrightarrow{\delta_2} & S^{10} & \xrightarrow{\delta_1} & S \\ & & & & \oplus & & \oplus & & & & \\ & & 0 & \longrightarrow & S^2 & \xrightarrow{\cong} & S^2 & \longrightarrow & 0 & & \end{array}$$

The middle row above gives the minimal graded free resolution $(F_\bullet, \delta_\bullet)$ of (5). In particular, with respect to the appropriate basis of S^7, the differential δ_4 is

$$\theta_4 = \begin{pmatrix} -f & e & -b & h & c & a & 2 \end{pmatrix}^T$$

and we can see that the kernel of $\delta_3 \otimes_S S/\mathfrak{m}$ is isomorphic to $\mathbb{Z}_2^6 \oplus \mathbb{Z}_2/2\mathbb{Z}_2$.

Let us now consider B in $S = \mathbb{Z}_p[a, \ldots, h]$, where p is a prime, $p \neq 2$. We note that 2 is now a unit and through a series of base changes $G_\bullet \otimes S$ breaks into

$$\begin{array}{ccccccccccc} 0 & \longrightarrow & S^2 & \xrightarrow{\cong} & S^2 & \longrightarrow & 0 & & & & \\ & & & & \oplus & & & & & & \\ & & 0 & \longrightarrow & S^7 & \xrightarrow{\delta_3} & S^{15} & \xrightarrow{\delta_2} & S^{10} & \xrightarrow{\delta_1} & S \\ & & & & \oplus & & \oplus & & & & \\ & & 0 & \longrightarrow & S^2 & \xrightarrow{\cong} & S^2 & \longrightarrow & 0 & & \end{array}$$

where he middle row above gives the minimal graded free resolution S/B. In this case the Betti numbers of $M = S/B$ in the special and generic fibre coincide. The uniqueness of the Smith normal form leads us to the following algorithm, to decide whether the Betti numbers differ in the special and generic fibre.

Algorithm 1: Testing whether the minimal free resolution depends on the characteristic of the base field.

Input: Homogeneous elements $f_1, \ldots, f_s \in T = \mathbb{Z}[w_1, \ldots, w_g]$.

Output: The set of primes p for which the Betti numbers of $I = \langle f_1, \ldots, f_s \rangle$ in $k[w_1, \ldots, w_g]$ depend on $\mathrm{char}(k)$.

Method:

(1) Compute a free resolution $(G_\bullet, \theta_\bullet)$ of T/I.
(2) Let A_i be the corresponding matrices of the differentials, for $i \geq 1$. Set $w_1, \ldots, w_g = 0$ for all entries of A_i to obtain the matrices B_i, for $i \geq 1$.
(3) Compute the Smith normal form of B_i, for $i \geq 1$.
(4) Collect all primes p that divide some nonzero entry of the Smith normal form of B_i, for $i \geq 1$.

We note that given a graded ideal I of $\mathbb{Z}[w_1, \ldots, w_g]$, the above algorithm indicates the primes for which the Betti numbers of $I\mathbb{Q}_p[w_1, \ldots, w_g]$ differ from the Betti numbers of $I\mathbb{F}_p[w_1, \ldots, w_g]$ and provide possible obstruction to the lifting problem.

ACKNOWLEDGMENTS

Received financial support by program: "Supporting researchers with emphasis to young researchers, cycle B", MIS 5047968.

REFERENCES

[1] William W. Adams and Philippe Loustaunau. *An introduction to Gröbner bases*, volume 3 of *Graduate Studies in Mathematics*. American Mathematical Society, Providence, RI, 1994. doi:10.1090/gsm/003.

[2] Christian Bopp and Frank-Olaf Schreyer. A version of green's conjecture in positive characteristic. *Experimental Mathematics*, 0(0):1–6, 2019. arXiv:https://doi.org/10.1080/10586458.2019.1576082, doi:10.1080/10586458.2019.1576082.

[3] Hara Charalambous, Kostas Karagiannis, and Aristides Kontogeorgis. The relative canonical ideal of the Artin-Schreier-Kummer-Witt family of curves, 2019. arXiv:arXiv:1905.05545.

[4] Jean-Guillaume Dumas, Frank Heckenbach, David Saunders, and Volkmar Welker. Computing simplicial homology based on efficient Smith normal form algorithms. In *Algebra, geometry, and software systems*, pages 177–206. Springer, Berlin, 2003.

[5] David Eisenbud. *The geometry of syzygies*, volume 229 of *Graduate Texts in Mathematics*. Springer-Verlag, New York, 2005. A second course in commutative algebra and algebraic geometry.

[6] Daniel R. Grayson and Michael E. Stillman. Macaulay2, a software system for research in algebraic geometry. Available at http://www.math.uiuc.edu/Macaulay2/.

[7] Sotiris Karanikolopoulos and Aristides Kontogeorgis. Integral representations of cyclic groups acting on relative holomorphic differentials of deformations of curves with automorphisms. *Proc. Amer. Math. Soc.*, 142(7):2369–2383, 2014. URL: https://doi.org/10.1090/S0002-9939-2014-12010-7.

[8] Aristides Kontogeorgis, Alexios Terezakis, and Ioannis Tsouknidas. Automorphisms and the canonical ideal, 2019. arXiv:arXiv:1909.10282.

[9] Andrew Obus. The (local) lifting problem for curves. In *Galois-Teichmüller theory and arithmetic geometry*, volume 63 of *Adv. Stud. Pure Math.*, pages 359–412. Math. Soc. Japan, Tokyo, 2012.

[10] Andrew Obus. Lifting of curves with automorphisms. In *Open problems in arithmetic algebraic geometry*, volume 46 of *Adv. Lect. Math. (ALM)*, pages 9–59. Int. Press, Somerville, MA, [2019] ©2019.

[11] Irena Peeva. *Graded syzygies*, volume 14 of *Algebra and Applications*. Springer-Verlag London, Ltd., London, 2011. doi:10.1007/978-0-85729-177-6.

[12] Mahrud Sayrafi. Computations over local rings in macaulay2, 2017. arXiv:arXiv:1710.09830.

[13] Charles A. Weibel. *An introduction to homological algebra*. Cambridge University Press, Cambridge, 1994.

Compatible Rewriting of Noncommutative Polynomials for Proving Operator Identities

Cyrille Chenavier*
Clemens Hofstadler*
Clemens G. Raab†
Georg Regensburger
{cyrille.chenavier,clemens.hofstadler,clemens.raab,georg.regensburger}@jku.at
Institute for Algebra, Johannes Kepler University
Linz, Austria

ABSTRACT

The goal of this paper is to prove operator identities using equalities between noncommutative polynomials. In general, a polynomial expression is not valid in terms of operators, since it may not be compatible with domains and codomains of the corresponding operators. Recently, some of the authors introduced a framework based on labelled quivers to rigorously translate polynomial identities to operator identities. In the present paper, we extend and adapt the framework to the context of rewriting and polynomial reduction. We give a sufficient condition on the polynomials used for rewriting to ensure that standard polynomial reduction automatically respects domains and codomains of operators. Finally, we adapt the noncommutative Buchberger procedure to compute additional compatible polynomials for rewriting. In the package `OperatorGB`, we also provide an implementation of the concepts developed.

CCS CONCEPTS

• **Theory of computation** → **Equational logic and rewriting**; *Automated reasoning*; • **Computing methodologies** → *Symbolic calculus algorithms.*

KEYWORDS

Rewriting, noncommutative polynomials, quiver representations, automated proofs, completion

ACM Reference Format:
Cyrille Chenavier, Clemens Hofstadler, Clemens G. Raab, and Georg Regensburger. 2020. Compatible Rewriting of Noncommutative Polynomials for Proving Operator Identities. In *International Symposium on Symbolic and Algebraic Computation (ISSAC '20), July 20–23, 2020, Kalamata, Greece.* ACM, New York, NY, USA, 8 pages. https://doi.org/10.1145/3373207.3404047

*The author was supported by the Austrian Science Fund (FWF): P 27229 and P 32301.
†The author was supported by the Austrian Science Fund (FWF): P 31952.

ISSAC '20, July 20–23, 2020, Kalamata, Greece

ACM ISBN 978-1-4503-7100-1/20/07…$15.00
https://doi.org/10.1145/3373207.3404047

1 INTRODUCTION

Properties of linear operators can often be expressed in terms of identities they satisfy. Algebraically, these identities can be represented in terms of noncommutative polynomials in some set X. The elements of X correspond to basic operators and polynomial multiplication models composition of operators. In contrast to addition and multiplication of polynomials, addition and composition of operators are restricted by their domains and codomains. Similarly, also in path algebras all elements can be added together without restriction. Hence, not every computation with polynomials (or in path algebras) can be translated into a computation with operators. Proving that a claimed operator identity follows from assumed identities corresponds to the polynomial associated to the claimed identity being expressible in terms of polynomials associated to the assumptions. However, having such a representation is not enough for proving an operator identity in general, since computations with noncommutative polynomials ignore compatibility conditions between domains and codomains of the operators. Our aim is to work out criteria so that the computations with polynomials are automatically valid in terms of operators.

In order to represent domains and codomains of operators, we use the framework introduced recently in [19]. So, we consider a quiver (i.e., a directed multigraph) Q, where vertices correspond to functional spaces and edges correspond to basic operators between those spaces and are labelled with symbols from X. Then, paths in Q correspond to composition of basic operators and induce monomials over X that are compatible with Q. We denote the set of polynomials associated to the assumptions by F and the polynomial associated to the claimed identity by f. Note that we can allow the same label for different edges if the corresponding operators satisfy the same identities. For instance, differential and integral operators can act on different functional spaces, as illustrated in our running example below. Even though it is always possible to label edges uniquely, using the same label for different edges allows to reduce the number of indeterminates and polynomials in the computation. In particular, certain infinite quivers can also be treated with finitely many polynomials. Informally, a polynomial is compatible with the quiver if it makes sense in terms of operators and f is called Q-consequence of F if it can be obtained from F by doing computations using compatible polynomials only. This means that these computations also make sense in terms of operators.

Obviously, the claim f and the assumptions F have to be compatible with Q. In [19], it was shown that f is a Q-consequence of F if

f lies in the two-sided ideal (F) and each element of F is uniformly compatible. Uniform compatibility of a polynomial means that all its monomials can be assigned the same combinations of domains and codomains. This is in particular the case when each edge has a unique label and polynomials do not have a constant term. Note that ideal membership can be checked independently of Q and is undecidable in general. In practice, however, it can often be checked by computing a (partial) noncommutative Gröbner basis G from F and reducing f to zero by G, see [17]. The package `OperatorGB` [13] can check compatibility of polynomials with quivers and, using partial Gröbner bases, can compute explicit representations of polynomials in terms of generators of the ideal. For more details on algorithmic aspects and the package see [12]. Versions for Mathematica and SageMath along with documentation can be obtained at:

http://gregensburger.com/softw/OperatorGB

In this paper, we first generalize the algebraic framework of [19] so that the quiver can be chosen in a more flexible way and more operator statements can be proven, see the running example below. Doing so requires to work out a proper way to arrange polynomial rewriting and partial noncommutative Gröbner basis computations algorithmically. In particular, in Section 3, we generalize the formal definition of Q-consequences to the case when elements of F are compatible but not necessarily uniformly compatible. Then, we show in Section 4 that being a Q-consequence implies that the corresponding operator identity can indeed be proven by computations with operators, see Theorem 15. Since elements of F do not have to be uniformly compatible, we impose in Section 5 restrictions on the polynomial rewriting, so that it respects the quiver. For the same reason, we also impose restrictions on the computation of partial Gröbner bases in Section 6. Based on such a partial Gröbner basis, one often can prove algorithmically that f is a Q-consequence of F just by standard polynomial reduction, see Corollary 24 and Theorem 28. To this end, we also extend the package `OperatorGB`.

Gröbner bases for noncommutative polynomials have been applied to operator identities in the pioneering work [10, 11], where Gröbner bases are used to simplify matrix identities in linear systems theory. In [9, 15], the main strategy for solving matrix equations, coming from factorization of engineering systems and matrix completion problems, is to apply Gröbner bases with respect to an ordering appropriate for elimination. The same approach was used in [20] to compute Green's operators for linear two-point boundary problems with constant coefficients.

If edges of the quiver have unique labels, it has been observed in the literature that the operations used in the noncommutative analog of Buchberger's algorithm respect compatibility of polynomials with domains and codomains of operators, cf. [10, Thm. 25]. See also Remark 31 and Theorem 32 for a formal statement using the framework of the present paper. For an analogous observation in the context of path algebras, see [18, Sec. 47.10], for which a Gröbner basis theory has been established, see also e.g. [7]. We were informed in personal communication that questions related to proving operator identities via computations of Gröbner bases are also addressed in [16].

Alternatively, computations with operators can also be modelled by partial algebras arising from diagrams, for which an analogous notion of Gröbner bases was sketched in [1, Sec. 9] and developed in [3]. Moreover, generalizations of Gröbner bases and syzygies are considered in [8], where higher-dimensional linear rewriting systems are introduced for rewriting of operators with domains and codomains.

We conclude this section with a small running example that we use throughout the paper to illustrate the notions that we introduce from practical point of view. This example, which was treated in [19] with infinite smoothness of coefficients and an infinite quiver, can be proven now for finite smoothness of coefficients using a finite quiver and non-uniformly compatible polynomials. A Mathematica notebook that illustrates the use of the new functionality of the package using this running example can be obtained at the webpage mentioned above.

Example 1. *Consider the inhomogeneous linear differential equation*

$$y''(x) + A_1(x)y'(x) + A_0(x)y(x) = r(x)$$

and assume that it can be factored into the two first-order equations

$$y'(x) - B_2(x)y(x) = z(x) \quad \text{and} \quad z'(x) - B_1(x)z(x) = r(x).$$

It is well-known that a particular solution is given by the nested integral

$$y(x) = H_2(x) \int_{x_2}^{x} H_2(t)^{-1} H_1(t) \int_{x_1}^{t} H_1(u)^{-1} r(u)\, du\, dt, \quad (1)$$

where $H_i(x)$ is a solution of $y'(x) - B_i(x)y(x) = 0$ such that $H_i(x)^{-1}$ exists. In order to translate this claim into an operator identity, let us consider the differentiation $\partial : y(x) \mapsto y'(x)$ and the two integrations

$$\int_1 : y(x) \mapsto \int_{x_1}^{x} y(t)\, dt \quad \text{and} \quad \int_2 : y(x) \mapsto \int_{x_2}^{x} y(t)\, dt.$$

Moreover, any function $F(x)$ induces a multiplication operator $F : y(x) \mapsto y(x)F(x)$ and $\cdot$ denotes the composition of operators. Thus, the factored differential equation and the solution correspond to the following operators

$$L := (\partial - B_1) \cdot (\partial - B_2), \quad S := H_2 \cdot \textstyle\int_2 \cdot H_2^{-1} \cdot H_1 \cdot \int_1 \cdot H_1^{-1}$$

and the claim corresponds to the identity $L \cdot S = \mathrm{id}$. In terms of functions, this means that $y(x) = (Sr)(x)$ is a solution of

$$(Ly)(x) = r(x). \quad (2)$$

We express properties used in the proof by identities. By the Leibniz rule, H_i being a solution of the factor differential equation means

$$\partial \cdot H_i = H_i \cdot \partial + B_i \cdot H_i$$

and the invertibility corresponds to $H_i \cdot H_i^{-1} = \mathrm{id}$. The fundamental theorem of calculus corresponds to

$$\partial \cdot \textstyle\int_1 = \mathrm{id}, \qquad \partial \cdot \int_2 = \mathrm{id}\,.$$

In Example 4, we will show how these operator identities can be translated into noncommutative polynomials that are compatible with a quiver. Then, we complete the proof of $L \cdot S = \mathrm{id}$ using our algebraic framework in Examples 8 and 16.

Throughout the paper, we fix a commutative ring R with unit as well as a set X.

2 PRELIMINARIES

In this section, we recall the main definitions and basic facts from [19] that formalize compatibility of polynomials with a labelled quiver.

We consider the free noncommutative algebra $R\langle X\rangle$ generated by the alphabet X: it can be regarded as the ring of noncommutative polynomials in the set of indeterminates X with coefficients in R, where indeterminates commute with coefficients but not with each other. The monomials are words $x_1 \dots x_n \in \langle X\rangle$, $x_i \in X$, including the empty word 1. Every polynomial $f \in R\langle X\rangle$ has a unique representation as a sum

$$f = \sum_{m\in\langle X\rangle} c_m m$$

with coefficients $c_m \in R$, such that only finitely many coefficients are nonzero, and its support is $\mathrm{supp}(f) := \{m \in \langle X\rangle \mid c_m \neq 0\}$.

Recall that a quiver is a tuple (V,E,s,t), where V is a set of vertices, E is a set of edges, and $s,t : E \to V$ are source and target maps, that are extended to all paths $p = e_n \cdots e_1$ by letting $s(p) = s(e_1)$ and $t(p) = t(e_n)$. For every vertex $v \in V$, there is a distinct empty path ϵ_v that starts and ends in v without passing through any edge and satisfies $\epsilon_{t(p)}p = p = p\epsilon_{s(p)}$ for all paths p. A labelled quiver, $Q = (V,E,X,s,t,l)$ is a quiver equipped with a label function $l : E \to X$ of edges into the alphabet X. We extend l into a function from paths to monomials by letting $l(p) = l(e_n)\cdots l(e_1) \in \langle X\rangle$, and $l(\epsilon_v) = 1$ is the empty word for every vertex v. For the remainder of this section, we fix a labelled quiver $Q = (V,E,X,s,t,l)$.

Definition 2. *Given a monomial $m \in \langle X\rangle$, we define the* set of signatures *of m as*

$$\sigma(m) := \{(s(p),t(p)) \mid p \text{ a path in } Q \text{ with } l(p) = m\} \subseteq V \times V.$$

A polynomial $f \in R\langle X\rangle$ is said to be compatible *with Q if its set of signatures $\sigma(f)$ is nonempty, where:*

$$\sigma(f) := \bigcap_{m\in\mathrm{supp}(f)} \sigma(m) \subseteq V \times V.$$

Finally, we denote by $s(f)$ and $t(f)$ the images of $\sigma(f)$ under the natural projections of $V \times V$ on V.

Note that we have $\sigma(0) = V \times V$ and $\sigma(1) = \{(v,v) \mid v \in V\}$.

Computing with compatible polynomials does not always result in compatible polynomials. However, under some conditions, the sum and product of compatible polynomials are compatible as well. The following properties of sets of signatures are straightforward to prove; see also Lemmas 10 and 11 in [19].

Lemma 3. *Let $f,g \in R\langle X\rangle$ be compatible with Q. Then,*

1. *If $\sigma(f) \cap \sigma(g) \neq \emptyset$, then $f+g$ is compatible and $\sigma(f+g)$ contains $\sigma(f) \cap \sigma(g)$.*
2. *If $s(f) \cap t(g) \neq \emptyset$, then fg is compatible and $\sigma(fg)$ contains*

$$\{(u,w) \mid \exists v \in s(f) \cap t(g) : (u,v) \in \sigma(g) \wedge (v,w) \in \sigma(f)\}.$$

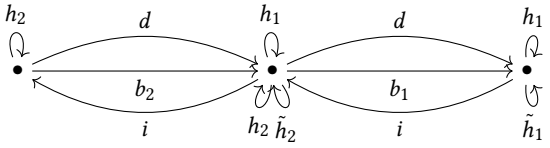
Example 4. *Let us continue the running example. The Leibniz rule and invertibility for H_1 and H_2 and the fundamental theorem of calculus correspond to the following noncommutative polynomials in $\mathbb{Z}\langle X\rangle$, where $X = \{h_1,h_2,b_1,b_2,\tilde{h}_1,\tilde{h}_2,i,d\}$.*

$$f_1 = dh_1 - h_1d - b_1h_1, \qquad f_2 = dh_2 - h_2d - b_2h_2,$$
$$f_3 = h_1\tilde{h}_1 - 1, \qquad f_4 = h_2\tilde{h}_2 - 1, \qquad f_5 = di - 1$$

We collect these polynomials in the set $F := \{f_1,\dots,f_5\}$. Notice that we represent the two integrals by a single indeterminate, so we only need one polynomial for the fundamental theorem of calculus. The claimed identity corresponds to

$$f := (d-b_1)(d-b_2)h_2i\tilde{h}_2h_1i\tilde{h}_1 - 1.$$

Since integration and differentiation decrease and increase the regularity of functions, it is natural to consider the following labelled quiver with 3 vertices (more details are given Section 4) with labels in the alphabet X. Instead of giving names to vertices and edges, we draw them as bullets and arrows oriented from source to target, respectively, and the label of an edge is shown next to the arrow representing that edge.



Either directly or by the package, we check that f and each element of F are compatible with the quiver. Denoting the vertices from left to right by v_1,v_2,v_3, we obtain the following signatures.

$$\sigma(f_1) = \{(v_2,v_3)\}, \qquad \sigma(f_2) = \{(v_1,v_2)\},$$
$$\sigma(f_3) = \{(v_3,v_3)\}, \qquad \sigma(f_4) = \{(v_2,v_2)\},$$
$$\sigma(f_5) = \{(v_2,v_2),(v_3,v_3)\}, \qquad \sigma(f) = \{(v_3,v_3)\}$$

To determine $\sigma(f)$, for example, notice that $\sigma(h_2i\tilde{h}_2h_1i\tilde{h}_1) = \{(v_3,v_1)\}$ and that $\sigma(dd) = \sigma(b_1d) = \sigma(db_2) = \sigma(b_1b_2) = \{(v_1,v_3)\}$ and recall that $\sigma(1)$ contains all pairs of the form (v_i,v_i).

3 Q-CONSEQUENCES

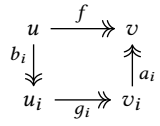
The following definition characterizes the situations when a representation of the claim f in terms of the assumptions F is also valid in terms of operators. This generalizes the notion of Q-consequence given in [19] and we prove new properties of this notion. Throughout the section, we fix a labelled quiver Q with labels in X.

Definition 5. *A* Q-consequence *of some $F \subseteq R\langle X\rangle$ is a polynomial $f \in R\langle X\rangle$, compatible with Q, such that there exist $n \in \mathbb{N}$, $g_i \in F$, $a_i,b_i \in R\langle X\rangle$, $1 \le i \le n$, such that*

$$f = \sum_{i=1}^{n} a_ig_ib_i \tag{3}$$

and, for every $(u,v) \in \sigma(f)$ and every i, there exist vertices u_i,v_i such that $(u,u_i) \in \sigma(b_i)$, $(u_i,v_i) \in \sigma(g_i)$, and $(v_i,v) \in \sigma(a_i)$.

The conditions on the signatures mean that, for each i, there exist three paths in the quiver as illustrated in the following diagram.



Representations of the form (3) are not unique and finding them is a hard problem in general, their existence is even undecidable. However, proving that a given representation (3) satisfies the required conditions of the above definition is straightforward, but in

general these conditions are not satisfied. In Proposition 7, we give an alternative criterion for Q-consequences. This criterion will play an important role later in Section 5 on rewriting. Before, we need the following lemma.

Lemma 6. *Let $m \in \langle X\rangle$ and $g \in R\langle X\rangle$ such that $\sigma(m) \subseteq \sigma(g)$. Then, for all monomials $a, b \in \langle X\rangle$, we have $\sigma(amb) \subseteq \sigma(agb)$. Moreover, for every $(u,v) \in \sigma(amb)$, there exist two vertices $\tilde{u}, \tilde{v}$ such that $(u,\tilde{u}) \in \sigma(b)$, $(\tilde{u},\tilde{v}) \in \sigma(g)$, and $(\tilde{v},v) \in \sigma(a)$.*

Proof. For every $(u,v) \in \sigma(amb)$, there exists a path from u to v with label amb. We split this path in 3 parts: the first part β has label b, the third part α has label a, and the second part has label m. Since $\sigma(m) \subseteq \sigma(g)$, for every $\tilde{m} \in \operatorname{supp}(g)$, there exists a path γ from $\tilde{u} := t(\beta)$ to $\tilde{v} := s(\alpha)$ with label $\tilde{m}$. Hence, $a\tilde{m}b$ is the label of $\alpha\gamma\beta$. Consequently, $(\tilde{u},\tilde{v}) \in \sigma(\tilde{m})$ and $\sigma(amb) \subseteq \sigma(a\tilde{m}b)$ for every $\tilde{m} \in \operatorname{supp}(g)$, i.e. $(\tilde{u},\tilde{v}) \in \sigma(g)$ and $\sigma(amb) \subseteq \sigma(agb)$. □

Proposition 7. *Let $F \subseteq R\langle X\rangle$ be a set of polynomials such that for every $g \in F$, there exists $m_g \in \operatorname{supp}(g)$ such that $\sigma(m_g) \subseteq \sigma(g)$. Let $f \in R\langle X\rangle$ be a compatible polynomial such that there exist $n \in \mathbb{N}$, $\lambda_i \in R$, $g_i \in F$, $a_i, b_i \in \langle X\rangle$, $1 \le i \le n$, such that*

$$f = \sum_{i=1}^{n} \lambda_i a_i g_i b_i, \tag{4}$$

and $\sigma(f) \subseteq \sigma(a_i m_{g_i} b_i)$ for all i. Then, f is a Q-consequence of F.

Proof. By hypotheses, f is compatible and for every $(u,v) \in \sigma(f)$ and for every $1 \le i \le n$, we have $(u,v) \in \sigma(a_i m_{g_i} b_i)$. Hence, using the hypothesis $\sigma(m_{g_i}) \subseteq \sigma(g_i)$, from Lemma 6, there exist vertices u_i and v_i such that $(u,u_i) \in \sigma(b_i)$, $(u_i,v_i) \in \sigma(g_i)$ and $(v_i,v) \in \sigma(a_i)$. As a consequence, f is a Q-consequence of F. □

Note that if for $m_g \in \operatorname{supp}(g)$, we have $\sigma(m_g) \subseteq \sigma(g)$, then $\sigma(m_g) = \sigma(g)$ holds by definition.

Example 8. *Let us continue Example 4. We show that f is a Q-consequence of F by considering the following representation:*

$$f = f_1 i\tilde{h}_1 + (d-b_1) f_2 i\tilde{h}_2 h_1 i\tilde{h}_1 + f_3 + (d-b_1) f_4 h_1 i\tilde{h}_1 + (d-b_1) h_2 f_5 \tilde{h}_2 h_1 i\tilde{h}_1 + h_1 f_5 \tilde{h}_1. \tag{5}$$

Such a representation can be obtained with the package by tracking cofactors in polynomial reduction w.r.t. a monomial order. Here, we consider any degree-lexicographic order (i.e. monomials are sorted first by their length then from left to right by their letters) such that d is greater than h_i's and b_i's. Then, f can be reduced to zero using F, which gives (5)*. While the package only computes over the field $\mathbb{Q}$, one can check that this representation also holds over the ring $\mathbb{Z}$. Representations computed by the package are not necessarily optimal regarding their degree or number of terms. Now, we have to check assumptions on signatures by checking either Definition 5 or the assumptions of Proposition 7, both options are implemented in the package. For applying Proposition 7 by hand, we can choose $m_{f_1} = dh_1, m_{f_2} = h_2 d, m_{f_3} = h_1\tilde{h}_1, m_{f_4} = h_2\tilde{h}_2$, and $m_{f_5} = di$, which satisfy $m_{f_i} \in \operatorname{supp}(f_i)$ and $\sigma(m_{f_i}) = \sigma(f_i)$. Expanding* (5) *in the form* (4)*, we may check that $\sigma(a_i m_{g_i} b_i) = \{(v_3, v_3)\} = \sigma(f)$ for every summand in the representation* (4)*, which proves that f is a Q-consequence of F.*

In Section 6, we will exploit that being a Q-consequence is transitive, which requires a new proof compared to [19, Corollary 18].

Proposition 9. *Let $F, G \subseteq R\langle X\rangle$ be sets of polynomials such that each element of G is a Q-consequence of F. Then, any Q-consequence of G is also a Q-consequence of F.*

Proof. Let h be a Q-consequence of G. So, it is compatible with Q. Moreover, $h = \sum_i a_i g_i b_i$, with $g_i \in G$ and $a_i, b_i \in R\langle X\rangle$ such that for every $(u,v) \in \sigma(h)$ and every i, there exist vertices u_i, v_i such that $(u,u_i) \in \sigma(b_i)$, $(u_i,v_i) \in \sigma(g_i)$, and $(v_i,v) \in \sigma(a_i)$. For every i, since g_i is a Q-consequence of F, there exist $a_{i,j}, b_{i,j} \in R\langle X\rangle$ and $f_{i,j} \in F$ such that $g_i = \sum_j a_{i,j} f_{i,j} b_{i,j}$ and, for every $(u_i,v_i) \in \sigma(g_i)$ and every j, there exist $(u_{i,j},v_{i,j}) \in \sigma(f_{i,j})$ such that $(u_i,u_{i,j}) \in \sigma(b_{i,j})$ and $(v_{i,j},v_i) \in \sigma(a_{i,j})$. Altogether, we have

$$h = \sum_i \sum_j a_i a_{i,j} f_{i,j} b_{i,j} b_i.$$

For all i,j, the vertices u_i and v_i belong to $s(b_{i,j}) \cap t(b_i)$ and $s(a_i) \cap t(a_{i,j})$, respectively, so that, by Item 2 of Lemma 3, $(u,u_{i,j}) \in \sigma(b_{i,j} b_i)$ and $(v_{i,j},v) \in \sigma(a_i a_{i,j})$, respectively. Hence, h is a Q-consequence of F. □

4 REALIZATIONS

In this section, we formalize the translation of polynomials to operators by substituting indeterminates by basic operators. In particular, we show in Theorem 15 that being a Q-consequence is enough to ensure that the corresponding operator identity can be inferred from the assumed operator identities. To this end, first, we summarize the relevant notions and basic facts from [19, Section 5].

For a quiver (V,E,s,t) and a ring R, $(\mathcal{M},\varphi)$ is called a *representation* of the quiver (V,E,s,t), if $\mathcal{M} = (\mathcal{M}_v)_{v\in V}$ is a family of R-modules and φ is a map that assigns to each $e \in E$ an R-linear map $\varphi(e) : \mathcal{M}_{s(e)} \to \mathcal{M}_{t(e)}$, see e.g. [5, 6]. Note that any nonempty path $e_n \ldots e_1$ in the quiver induces an R-linear map $\varphi(e_n)\cdot\ldots\cdot\varphi(e_1)$, since the maps $\varphi(e_{i+1})$ and $\varphi(e_i)$ can be composed for every $i \in \{1,\ldots,n-1\}$ by definition of φ. Similarly, for every $v \in V$, the empty path ϵ_v induces the identity map on $\mathcal{M}_v$.

Remark 10. *All notions and results of this section naturally generalize to R-linear categories by considering objects and morphisms in such a category instead of R-modules and R-linear maps, respectively. For more details, see Section 5.2 in [19].*

Definition 11. *Let Q be a labelled quiver with labelling l. We call a representation $(\mathcal{M},\varphi)$ of Q* consistent *with the labelling l if for any two nonempty paths $p = e_n \ldots e_1$ and $q = d_n \ldots d_1$ in Q with the same source and target, equality of labels $l(p) = l(q)$ implies $\varphi(e_n)\cdot\ldots\cdot\varphi(e_1) = \varphi(d_n)\cdot\ldots\cdot\varphi(d_1)$ as R-linear maps.*

Remark 12. *If all paths with the same source and target have distinct labels, then every representation of that labelled quiver is consistent with its labelling. In particular, this holds if for every vertex all outgoing edges have distinct labels or analogously for incoming edges. These sufficient conditions can be verified without the need for considering all possible paths.*

For Definition 13 and Lemma 14, we fix a labelled quiver $Q = (V,E,X,s,t,l)$ and a consistent representation $\mathcal{R} = (\mathcal{M},\varphi)$ of Q. In order to define realizations of a polynomial, we first need to

introduce some notation. Given two vertices v, w, we write $R\langle X\rangle_{v,w}$ for the set of polynomials $f \in R\langle X\rangle$ such that $(v,w) \in \sigma(f)$. By Item 1 of Lemma 3, $R\langle X\rangle_{v,w}$ is a module, which is free with the monomials m such that $(v,w) \in \sigma(m)$ as its basis. Also, we denote by $\mathrm{Hom}_R(\mathcal{M}_v, \mathcal{M}_w)$ the set of R-linear maps from $\mathcal{M}_v$ to $\mathcal{M}_w$.

Definition 13. *For vertices $v, w \in V$, we define the R-linear map $\varphi_{v,w} : R\langle X\rangle_{v,w} \to \mathrm{Hom}_R(\mathcal{M}_v, \mathcal{M}_w)$ by*

$$\varphi_{v,w}(l(e_n \ldots e_1)) := \varphi(e_n)\cdot \ldots \cdot \varphi(e_1)$$

for all nonempty paths $e_n \ldots e_1$ in Q from v to w and, if $v = w$, also by $\varphi_{v,v}(1) := \mathrm{id}_{\mathcal{M}_v}$. For all $f \in R\langle X\rangle_{v,w}$, we call the R-linear map $\varphi_{v,w}(f)$ a realization *of f w.r.t. the representation $\mathcal{R}$ of Q.*

Notice that the map $\varphi_{v,w}$ is well-defined since, by consistency of $\mathcal{R}$, for every monomial $m \in R\langle X\rangle_{v,w}$, its realization $\varphi_{v,w}(m)$ does not depend on the path from v to w with label m.

In the proof of Theorem 15, we use an intermediate result given in [19, Lemma 31], whose statement is the following.

Lemma 14. *Let $u, v, w \in V$. Then, for all $f \in R\langle X\rangle_{v,w}$ and $g \in R\langle X\rangle_{u,v}$, we have that $fg \in R\langle X\rangle_{u,w}$ and*

$$\varphi_{u,w}(fg) = \varphi_{v,w}(f)\cdot\varphi_{u,v}(g).$$

Theorem 15. *Let $F \subseteq R\langle X\rangle$ be a set of polynomials and let Q be a labelled quiver with labels in X. If a polynomial $f \in R\langle X\rangle$ is a Q-consequence of F, then for all consistent representations of the quiver Q such that all realizations of all elements of F are zero, all realizations of f are zero.*

Proof. Assume that f is a Q-consequence, so that it is compatible with Q and it can be written in the form $\sum_i a_i g_i b_i$, such that for each $(u,v) \in \sigma(f)$ and each i, there exist vertices u_i, v_i such that $(u,u_i) \in \sigma(b_i)$, $(u_i,v_i) \in \sigma(g_i)$ and $(v_i,v) \in \sigma(a_i)$. Let us fix a consistent representation $\mathcal{R} = (\mathcal{M}, \varphi)$ of Q. By linearity of $\varphi_{u,v}$ and from Lemma 14, we have

$$\varphi_{u,v}(f) = \sum_i \varphi_{u,v}(a_i g_i b_i) = \sum_i \varphi_{v_i,v}(a_i)\cdot\varphi_{u_i,v_i}(g_i)\cdot\varphi_{u,u_i}(b_i).$$

Hence, if all realizations of all elements of F are zero, then $\varphi_{u,v}(f) = 0$, which means that all realizations of f w.r.t. $\mathcal{R}$ are zero. □

Example 16. *We finish our proof of* (2) *by considering certain representations of the quiver of Example 4. For a nonnegative integer k and an open interval $I \subseteq \mathbb{R}$, we assign the spaces $C^k(I)$, $C^{k+1}(I)$, and $C^{k+2}(I)$ to the vertices from right to left. Hence, differentiation and integration induce operators $\partial : C^{k+1}(I) \to C^k(I)$, $\partial : C^{k+2}(I) \to C^{k+1}(I)$, $\int_1 : C^k(I) \to C^{k+1}(I)$, and $\int_2 : C^{k+1}(I) \to C^{k+2}(I)$. We also assume the following regularity of functions: B_1 is C^k, H_1 and B_2 are C^{k+1}, and H_2 is C^{k+2} on I. Then, the natural representation associated with these operators is consistent. Moreover, we have seen in Example 8 that f is a Q-consequence of F. Since all realizations of f_i's are zero, by Theorem 15, all realizations of f are zero. In particular, for every nonnegative integer k and every $r(x) \in C^k(I)$, the function $y(x)$ defined by* (1) *is a solution of the differential equation* (2).

Instead of considering scalar differential equations we could consider differential systems of the form (2) *for vector-valued functions $y(x)$ of arbitrary dimension n. More explicitly, we can also consider coefficients $B_1(x), B_2(x)$ as $n \times n$ matrices, $r(x)$ as a vector of dimension n, and $H_1(x)$ and $H_2(x)$ as fundamental matrix solutions of the homogeneous systems $y'(x) - B_i(x)y(x) = 0$. We still obtain consistent representations of the quiver where the vertices are mapped to $C^k(I)^n$, $C^{k+1}(I)^n$, and $C^{k+2}(I)^n$, respectively. Then, Theorem 15 immediately proves that the function $y(x)$ defined by* (1) *is a solution of the inhomogeneous differential equation* (2). *Similarly, analogous statements for other suitable functional spaces can be proven just by choosing different representations of the quiver.*

5 COMPATIBLE REWRITING

In this section, we give conditions on the polynomials used for rewriting such that rewriting any compatible polynomial to zero by them proves that it is a Q-consequence. First, we recall from [19, Definition 2] a general notion of rewriting one polynomial by another in terms of an arbitrary monomial division. Notice that the standard polynomial reduction is a particular case, where m is the leading monomial of g w.r.t. a monomial order and λ is such that amb is cancelled in (6).

Definition 17. *Let $f, g \in R\langle X\rangle$ and let $m \in \mathrm{supp}(g)$ such that m divides some monomial $m_f \in \mathrm{supp}(f)$, i.e., $m_f = amb$ for monomials $a, b \in \langle X\rangle$. For every $\lambda \in R$, we say that f* can be rewritten *to*

$$h := f + \lambda agb, \tag{6}$$

using (g, m).

We fix a labelled quiver Q with labels in X. It turns out that to obtain Q-consequences using rewriting (Theorem 21), we need to choose suitable divisor monomials such that sets of signatures only increase. In particular, this is the case when divisor monomials have minimal set of signatures, as stated in the following lemma.

Lemma 18. *Let $f, g \in R\langle X\rangle$ and let $m \in \mathrm{supp}(g)$ be such that $\sigma(m) = \sigma(g)$. If f can be rewritten to $h = f + \lambda agb$ using (g, m), then*

$$\sigma(f) \subseteq \sigma(h) \quad \text{and} \quad \sigma(f) \subseteq \sigma(amb).$$

Proof. By definition of signatures, $\sigma(f) \subseteq \sigma(amb)$. By Lemma 6 and from $\sigma(m) = \sigma(g)$, we have $\sigma(amb) \subseteq \sigma(agb)$. Altogether, $\sigma(f)$ is included in $\sigma(agb)$, which itself is contained in $\sigma(\lambda agb)$. By Item 1 of Lemma 3, we deduce $\sigma(f) \subseteq \sigma(h)$. □

Now, we define the rewriting relation induced by a fixed choice of divisor monomials and its compatibility with a quiver. For any rewriting relation we denote single rewriting steps by $\to$ and the reflexive transitive closure by $\overset{*}{\to}$.

Definition 19. *Let $G \subseteq R\langle X\rangle$ be a set of polynomials and let $\mathrm{DM} : G \to \mathcal{P}(\langle X\rangle)$ be a function from G to the power set of $\langle X\rangle$, such that $\mathrm{DM}(g) \subseteq \mathrm{supp}(g)$, for every $g \in G$.*

1. *For $g \in G$, we say that $m \in \mathrm{DM}(g)$ is a* divisor monomial *of g w.r.t.* DM.
2. *We say that f* rewrites *to h by (G, DM), denoted as $f \to_{G,\mathrm{DM}} h$, if there exists $g \in G$ and a divisor monomial $m \in \mathrm{DM}(g)$ such that f can be rewritten to h using (g, m).*
3. *We say that* DM *is* compatible *with a labelled quiver Q if, for every $g \in G$ and every $m \in \mathrm{DM}(g)$, we have $\sigma(m) = \sigma(g)$.*

From now on, we fix a set of polynomials $G \subseteq R\langle X\rangle$ as well as a map DM selecting divisor monomials.

Remark 20. *Notice that there exist two extreme cases for* DM*:*

(1) DM *selects exactly one monomial for each $g \in G$, for instance, the leading monomial* $\mathrm{LM}(g)$ *w.r.t. a monomial order, see the example in Section 6.*
(2) *All monomials in* $\mathrm{supp}(g)$ *are divisor monomials. Then,* $\to_{G,\mathrm{DM}}$ *coincides with the rewriting relation introduced in [19, Definition 2], for which ideal membership is equivalent to reduction to zero, see Lemma 4 in [19]. Moreover, if such a* DM *is compatible with Q, then all polynomials in G are uniformly compatible, i.e., all monomials of a polynomial have the same signature set. The following theorem generalizes Corollary 17 in [19].*

Theorem 21. *Let $G \subseteq R\langle X\rangle$ and let* DM *be a function selecting divisor monomials as in Definition 19. Let $f \in R\langle X\rangle$ such that $f \overset{*}{\to}_{G,\mathrm{DM}} 0$. Then, for every labelled quiver Q with labels in X such that* DM *is compatible with Q, we have that*

$$f \text{ is compatible with } Q \iff f \text{ is a } Q\text{-consequence of } G.$$

Proof. Since f rewrites to zero, for some $n \in \mathbb{N}$, there exists a sequence $f = h_0 \to h_1 \to \cdots \to h_n = 0$. Hence, there exist $\lambda_i \in R$, $a_i, b_i \in \langle X\rangle$, $g_i \in G$, and $m_i \in \mathrm{DM}(g_i)$ such that $h_i = h_{i-1} + \lambda_i a_i g_i b_i$ and $a_i m_i b_i \in \mathrm{supp}(h_{i-1})$. Hence, f can be written as $f = \sum_{i=1}^{n} -\lambda_i a_i g_i b_i$. From Lemma 18, we conclude inductively that $\sigma(f) \subseteq \sigma(h_{i-1}) \subseteq \sigma(a_i m_i b_i)$. Hence, if f is compatible with Q, then f is a Q-consequence of G by Proposition 7. Conversely, if f is a Q-consequence of G, then it is compatible by definition. □

Example 22. *Let us translate Example 8 in the language introduced in this section. The leading monomials w.r.t. the degree-lexicographic order used in that example can be understood as the divisor monomials selected by the function* DM *defined on F such that* $\mathrm{DM}(f_i) = \{\mathrm{LM}(f_i)\}$ *holds for all i. In particular,*

$$\mathrm{DM}(f_1) = \{dh_1\}, \quad \mathrm{DM}(f_2) = \{dh_2\},$$
$$\mathrm{DM}(f_3) = \{h_1\tilde{h}_1\}, \quad \mathrm{DM}(f_4) = \{h_2\tilde{h}_2\}, \quad \mathrm{DM}(f_5) = \{di\}.$$

Then, DM *is not compatible with Q, since $\sigma(f_2) = \{(v_1, v_2)\}$ is not equal to $\sigma(dh_2) = \{(v_1, v_2), (v_2, v_3)\}$. Hence, we cannot apply Theorem 21 to show that f is a Q-consequence of F even though $f \overset{*}{\to}_{F,\mathrm{DM}} 0$. So, we need to look at the explicit representation of f induced by this reduction, see Example 8. To apply Theorem 21, we need to redefine* DM *so that it is compatible with Q. In particular, we need to impose* $\mathrm{DM}(f_2) \subseteq \{h_2 d, b_2 h_2\}$. *If $b_2 h_2 \in \mathrm{DM}(f_2)$, then $f \overset{*}{\to}_{F,\mathrm{DM}} 0$, which gives another proof that f is a Q-consequence of F based on Theorem 21. Otherwise, if* $\mathrm{DM}(f_2) = \{h_2 d\}$, *then f is irreducible w.r.t. $\overset{*}{\to}_{F,\mathrm{DM}}$. Therefore, we need to complete F with Q-consequences of it such that* DM *remains compatible with Q and f reduces to zero, which is the topic of the next section.*

6 COMPATIBLE REDUCTIONS AND PARTIAL GRÖBNER BASES

In this section, we discuss standard noncommutative polynomial reduction as a special case of the rewriting approach from the previous section. Since in the noncommutative case, Gröbner bases are not necessarily finite, see [17], we also have to work with partial Gröbner bases which are obtained by finitely many iterations of the Buchberger procedure. We adapt the noncommutative Buchberger procedure for computing (partial) Gröbner bases that can be used for compatible rewriting.

In what follows, R is assumed to be a field $\mathbb{K}$ and we fix a monomial order $\leq$ on $\langle X\rangle$, that is, a well-founded total order compatible with multiplication on $\langle X\rangle$. We also fix a labelled quiver Q with labels in X and a set of polynomials $F \subseteq \mathbb{K}\langle X\rangle$. Given a set of polynomials $G \subseteq \mathbb{K}\langle X\rangle$, one step of the standard polynomial reduction w.r.t. G is denoted by $f \to_G h$.

As explained in Remark 20, the monomial order induces the DM function that selects leading monomials of a set $G \subseteq \mathbb{K}\langle X\rangle$. This DM function is compatible with Q if and only if all elements of G are Q-order compatible in the following sense.

Definition 23. *A compatible polynomial f is said to be* Q-order compatible *if $\sigma(\mathrm{LM}(f)) = \sigma(f)$.*

By transitivity of Q-consequences, see Proposition 9, and Theorem 21, we obtain the following statement.

Corollary 24. *Let $F \subseteq \mathbb{K}\langle X\rangle$, $G \subseteq (F)$, and $f \in \mathbb{K}\langle X\rangle$ such that $f \overset{*}{\to}_G 0$. Then, for all labelled quivers Q such that all elements of G are both Q-consequences of F and Q-order compatible, we have*

$$f \text{ is compatible with } Q \iff f \text{ is a } Q\text{-consequence of } F.$$

Remark 25. *For polynomials, being Q-order compatible can also be interpreted in terms of a partial monomial order. Given $m, m' \in \langle X\rangle$, we define $m \leq_Q m'$ if $m \leq m'$ and $\sigma(m') \subseteq \sigma(m)$. The partial order $\leq_Q$ respects multiplication since, by Lemma 6, $\sigma(m') \subseteq \sigma(m)$ implies $\sigma(am'b) \subseteq \sigma(amb)$ for all $a, b \in \langle X\rangle$. Then, f is Q-order compatible if and only if* $\mathrm{supp}(f)$ *admits a greatest element.*

Candidates for G as in Corollary 24 are partial Gröbner bases that are computed by the noncommutative Buchberger procedure [4, 17]. However, in view of the assumptions, we only add reduced S-polynomials that are both Q-consequences of F and Q-order compatible in each iteration. Checking Q-order compatibility is easy. Selecting Q-consequences is harder since we do not want to use explicit representations as in Definition 5. Instead, we propose a simpler criterion based on the following lemma and discussion.

First, we recall some terminology and fix notation for S-polynomials. Let $G \subseteq \mathbb{K}\langle X\rangle$. *Ambiguities* of G defined in [1], also called compositions in [2], are given by minimal overlaps or inclusions of the two leading monomials $\mathrm{LM}(g)$ and $\mathrm{LM}(g')$, where g and g' belong to G. Formally, each ambiguity can be described by a 6-tuple $\mathfrak{a} = (g, g', a, b, a', b')$, where a, b, a', b' are monomials such that, among other conditions, we have

$$a\,\mathrm{LM}(g)b = a'\,\mathrm{LM}(g')b'.$$

This monomial is called the *source* of $\mathfrak{a}$ and, if w.l.o.g. g and g' are monic, the S-polynomial of $\mathfrak{a}$ is $\mathrm{SP}(\mathfrak{a}) := agb - a'g'b'$, cf. [17].

Lemma 26. *Let $G \subseteq \mathbb{K}\langle X\rangle$ be a set of Q-order compatible polynomials and let s be a S-polynomial of G with compatible source $m \in \langle X\rangle$. Then $\sigma(m) \subseteq \sigma(s)$. If, moreover, $s \overset{*}{\to}_G \hat{s}$ with $\sigma(\hat{s}) \subseteq \sigma(m)$, then $\sigma(s) = \sigma(\hat{s}) = \sigma(m)$ and $\hat{s}$ is a Q-consequence of G.*

Proof. W.l.o.g. we assume that elements in G are monic. Since s is a S-polynomial of G of source m, there exist $g, g' \in G$ and monomials $a, a', b, b' \in \langle X\rangle$ such that $s = (m - agb) - (m - a'g'b')$ with $a\,\mathrm{LM}(g)b = a'\,\mathrm{LM}(g')b' = m$.

Let us prove the first assertion. The polynomials g and g' being Q-order compatible, we have $\sigma(\mathrm{LM}(g)) = \sigma(g)$ and $\sigma(\mathrm{LM}(g')) = \sigma(g')$. Hence, from Lemma 6, we have

$$\begin{aligned}\sigma(m) &= \sigma(a\,\mathrm{LM}(g)b) \subseteq \sigma(agb),\\ \sigma(m) &= \sigma(a'\,\mathrm{LM}(g')b') \subseteq \sigma(a'g'b').\end{aligned} \tag{7}$$

From this and $s = a'g'b' - agb$, we get that $\sigma(m) \subseteq \sigma(s)$.

Now, we assume that $s \overset{*}{\to}_G \hat{s}$, that is there is a rewriting sequence

$$s_0 = s \to_G s_1 \to_G \cdots \to_G s_n = \hat{s},$$

for some $n \in \mathbb{N}$, so that there exist $g_i \in G$, $a_i, b_i \in \langle X\rangle$, and $\lambda_i \in \mathbb{K}$, $1 \le i \le n$, such that $a_i\,\mathrm{LM}(g_i)b_i \in \mathrm{supp}(s_i)$ and $s_{i+1} = s_i + \lambda_i a_i g_i b_i$, so that we have $\hat{s} - s = \sum_i \lambda_i a_i g_i b_i$ and

$$\hat{s} = \sum_{i=1}^{n} \lambda_i a_i g_i b_i + a'g'b' - agb.$$

Using inductively $s_i \to_G s_{i+1}$ and Lemma 18, we get

$$\sigma(s) \subseteq \sigma(\hat{s}) \quad \text{and} \quad \sigma(s) \subseteq \sigma(a_i\,\mathrm{LM}(g_i)b_i), \tag{8}$$

so that we have

$$\sigma(m) \subseteq \sigma(s) \subseteq \sigma(\hat{s}).$$

If, moreover $\sigma(\hat{s}) \subseteq \sigma(m)$, then we get the following inclusions:

$$\sigma(\hat{s}) \subseteq \sigma(m) \subseteq \sigma(s) \subseteq \sigma(\hat{s}).$$

Hence, the equality $\sigma(\hat{s}) = \sigma(s) = \sigma(m)$ holds. Now, we show that $\hat{s}$ is a Q-consequence using Proposition 7. Since the elements of G are Q-order compatible, we have $\sigma(\mathrm{LM}(\tilde{g})) = \sigma(\tilde{g})$, for all $\tilde{g} \in G$. Moreover, since m is compatible and $\sigma(\hat{s}) = \sigma(m)$, $\hat{s}$ is compatible. Finally, from (7) and (8), we have the following inclusions: $\sigma(\hat{s}) \subseteq \sigma(a\,\mathrm{LM}(g)b)$, $\sigma(\hat{s}) \subseteq \sigma(a'\,\mathrm{LM}(g')b')$, and $\sigma(\hat{s}) \subseteq \sigma(a_i\,\mathrm{LM}(g_i)b_i)$. As a conclusion, $\hat{s}$ is a Q-consequence of G. □

Starting with a set of Q-order compatible polynomials F, we apply this lemma in the case where G is the current partial Gröbner basis computed during the completion procedure. In particular, if a reduced S-polynomial $\hat{s}$ satisfies $\sigma(\hat{s}) \subseteq \sigma(m)$ as in the lemma, then it is a Q-consequence of G. By transitivity, it is also a Q-consequence of F, which follows from the following observation.

Remark 27. *Consider a set $F \subseteq \mathbb{K}\langle X\rangle$ of compatible polynomials and a family of sets G_i inductively defined by $G_0 = \emptyset$ and $G_{i+1} = G_i \cup \{g_{i+1}\}$, where g_{i+1} is a Q-consequence of $F \cup G_i$. Using inductively transitivity of Q-consequences proven in Proposition 9, we obtain that, for each i, all elements of $F \cup G_i$ are Q-consequences of F.*

In summary, we obtain the following adaptation of the noncommutative version of Buchberger's procedure for computing a partial Gröbner basis composed of elements that are both Q-consequences of F and Q-order compatible. At each step, we select an S-polynomial s whose source m is a compatible monomial, and we keep a reduced form $\hat{s}$ only if it is Q-order compatible and $\sigma(\hat{s}) \subseteq \sigma(m)$. This procedure is implemented in the Mathematica package `OperatorGB`. Note that since the Buchberger procedure does not terminate in general for noncommutative polynomials, also our adaptation of it is not guaranteed to terminate.

Notice that the completion procedure just described can be slightly generalized by not necessarily computing reduced forms of S-polynomials. Instead, we only reduce an S-polynomial as long as it remains a Q-consequence, see the discussion above, and it remains Q-order compatible. This is stated formally in Procedure 1.

Procedure 1 Q-order compatible completion

Input: $F \subseteq \mathbb{K}\langle X\rangle$, a labelled quiver Q with labels in X, and a monomial order $\le$ such that every $f \in F$ is Q-order compatible
Output: $G \supseteq F$ a set of Q-consequences of F that are Q-order compatible

```
1:  P := ambiguities of F; G := F
2:  while P ≠ ∅ do
3:    choose 𝔞 ∈ P
4:    P := P \ {𝔞}; s := SP(𝔞); m := the source of 𝔞
5:    if m is compatible and σ(s) ⊆ σ(m) and s is Q-order compatible then
6:      while ∃s' : s →_G s' do
7:        if s' = 0 then
8:          go to 2 (i.e., break the outer if statement)
9:        else if σ(s') ⊆ σ(m) and s' is Q-order compatible then
10:         s := s'
11:       else
12:         go to 15 (i.e., break the inner while loop)
13:       end if
14:     end while
15:     G := G ∪ {s}
16:     P := P ∪ {ambiguities created by s}
17:   end if
18: end while
19: return G
```

Due to the checks in lines 5 and 9, each element of the output G of the procedure is both a Q-consequence of F and Q-order compatible. In summary, we have shown that our procedure is correct.

Theorem 28. *Let $F \subseteq \mathbb{K}\langle X\rangle$, let Q be a labelled quiver with labels in X and let $\le$ be a monomial order such that each element of F is Q-order compatible. Then, each element of the output G of Procedure 1 is both a Q-consequence of F and Q-order compatible.*

Remark 29. *In practice, additional criteria are needed to ensure termination of Procedure 1. In our package, we implement a stopping criterion by ignoring all ambiguities whose sources exceed a degree bound. In fact, the implementation proceeds in generations by deferring new ambiguities until all previous ones have been treated (instead of adding them in line 16). This ensures a fair selection strategy [17] and provides an additional stopping criterion on the number of generations. Moreover, in the package, cofactors are tracked through the computation to provide cofactor representations of elements of G in terms of F. For further details on the implementation see [12].*

Example 30. *Let us continue Example 22 in the case $\mathrm{DM}(f_2) = \{h_2 d\}$. For that, we consider the field $\mathbb{K} = \mathbb{Q}$ and a degree-lexicographic order such that $d < b_2 < h_2$ and d is greater than b_1 and h_1. Then, choosing the ambiguity $(f_2, f_5, 1, i, h_2, 1)$, the first iteration of the outer loop in Procedure 1 yields $G := F \cup \{b_2 h_2 i - d h_2 i + h_2\}$. With this G, we have $f \overset{*}{\to}_G 0$. From this reduction to 0, and since f is compatible with Q, f is a Q-consequence of F by Corollary 24. These*

computations can also be done by the package, see the example file accompanying this paper at the webpage of the package.

Remark 31. *We consider the special case when all edges of Q have unique labels. Then, all non-constant monomials have at most one signature. Therefore, every compatible polynomial is Q-order compatible for any monomial order, since the monomial 1 is the smallest. Moreover, one can show easily that the source of an ambiguity of two polynomials is compatible whenever these two polynomials are compatible with Q. In addition, from Lemma 18, it follows that polynomial reduction of compatible polynomials by compatible ones does not change the set of signatures unless the result of the reduction lies in $\mathbb{K}$. Altogether, Procedure 1 reduces to the standard Buchberger procedure (i.e., without checking signatures and compatibility during computation) as long as no S-polynomial is (or is reduced to) a nonzero constant. In other words, we have the following theorem, which, together with Theorem 15, generalizes Theorem 1 in [19].*

Theorem 32. *Let $F \subseteq \mathbb{K}\langle X\rangle$, let G be a (partial) Gröbner basis computed from F by the standard Buchberger procedure, and let $f \in \mathbb{K}\langle X\rangle$ such that $f \xrightarrow{*}_G 0$. If G does not contain a constant polynomial, then for every labelled quiver Q such that edges have unique labels in X and elements of F are compatible with Q, we have*

$$f \text{ is compatible with } Q \iff f \text{ is a } Q\text{-consequence of } F.$$

Moreover, if $1 \notin (F)$, then this equivalence holds for every $f \in (F)$.

7 SUMMARY AND DISCUSSION

By Theorem 15, for proving new operator identities from known ones, it suffices to show that the corresponding polynomials are Q-consequences. In practice, there are several options to prove that a compatible polynomial f is a Q-consequence of some set F of compatible polynomials. Each of these options can be turned into a certificate that f is a Q-consequence of F. Given an explicit representation (3) of f in terms of F, one can either check Definition 5 directly, or expand cofactors into monomials and apply Proposition 7. Alternatively, using compatible rewriting, if $f \xrightarrow{*}_{F,\mathrm{DM}} 0$ and the selection of divisor monomials by DM is compatible with Q, then f is a Q-consequence by Theorem 21. Altogether, from Theorems 15 and 21, we immediately obtain the following.

Corollary 33. *Let F be a set of polynomials,* DM *a function selecting divisor monomials, and f a polynomial such that $f \xrightarrow{*}_{F,\mathrm{DM}} 0$. Then, for all labelled quivers Q such that f, F, and* DM *are compatible with Q and for all consistent representations of Q such that all realizations of all elements of F are zero, all realizations of f are zero.*

Note that rewriting to zero w.r.t. F and DM is independent of the quiver Q. In particular, Theorem 32 in [19], which is the main result there, is obtained as a special case by Remark 20, if the above corollary is interpreted in terms of R-linear categories.

More generally, if one cannot verify that f can be rewritten to zero by F, there still might exist a set G of Q-consequences of F with divisor monomials selected by some DM such that $f \xrightarrow{*}_{G,\mathrm{DM}} 0$ and Proposition 9 can be applied. Algorithmically, based on suitable monomial orderings, Procedure 1 produces candidates G such that Corollary 24 can be used to prove that f is a Q-consequence of F by standard polynomial reduction.

Notice that Procedure 1 can be extended in various directions. For example, in order to systematically generate more Q-consequences, reduced S-polynomials that are not Q-order compatible could be collected in a separate set, which should not be used for constructing and reducing new S-polynomials. Instead of fixing a monomial ordering from the beginning, one might start with a partial ordering that is then extended during the completion procedure in order to make obtained S-polynomials Q-order compatible. More generally, without any partial ordering on monomials, one might even consider compatible functions DM which not necessarily select only one divisor monomial per polynomial and aim at completing the induced rewriting relation. However, termination of such rewriting relations is an issue. Finally, another topic for future research is to generalize the results of this paper to tensor reduction systems used for modelling linear operators as described in [14].

REFERENCES

[1] George M. Bergman. 1978. The diamond lemma for ring theory. *Adv. in Math.* 29 (1978), 178–218.

[2] Leonid A. Bokut'. 1976. Imbeddings into simple associative algebras. *Algebra i Logika* 15 (1976), 117–142, 245.

[3] Leonid A. Bokut, Yuqun Chen, and Yu Li. 2012. Gröbner-Shirshov bases for categories. In *Operads and universal algebra*. World Sci. Publ., Hackensack, NJ, 1–23.

[4] Bruno Buchberger. 1965. Ein Algorithmus zum Auffinden der Basiselemente des Restklassenrings nach einem nulldimensionalen Polynomideal. *Universität Innsbruck, Austria, Ph. D. Thesis* (1965). English translation in J. Symbolic Comput. 41 (2006), 475–511.

[5] Harm Derksen and Jerzy Weyman. 2005. Quiver representations. *Notices Amer. Math. Soc.* 52 (2005), 200–206.

[6] Peter B. Gothen and Alastair D. King. 2005. Homological algebra of twisted quiver bundles. *J. London Math. Soc. (2)* 71 (2005), 85–99.

[7] Edward L. Green. 1999. Noncommutative Gröbner bases, and projective resolutions. In *Computational methods for representations of groups and algebras (Essen, 1997)*. Birkhäuser, Basel, 29–60.

[8] Yves Guiraud, Eric Hoffbeck, and Philippe Malbos. 2019. Convergent presentations and polygraphic resolutions of associative algebras. *Math. Z.* 293 (2019), 113–179.

[9] J. William Helton and Mark Stankus. 1999. Computer assistance for "discovering" formulas in system engineering and operator theory. *J. Funct. Anal.* 161 (1999), 289–363.

[10] J. William Helton, Mark Stankus, and John J. Wavrik. 1998. Computer simplification of formulas in linear systems theory. *IEEE Trans. Automat. Control* 43 (1998), 302–314.

[11] J. William Helton and John J. Wavrik. 1994. Rules for computer simplification of the formulas in operator model theory and linear systems. In *Nonselfadjoint operators and related topics (Beer Sheva, 1992)*. Birkhäuser, Basel, 325–354.

[12] Clemens Hofstadler. 2020. *Certifying operator identities and ideal membership of noncommutative polynomials.* Master's thesis. Johannes Kepler University Linz, Austria.

[13] Clemens Hofstadler, Clemens G Raab, and Georg Regensburger. 2019. Certifying operator identities via noncommutative Gröbner bases. *ACM Commun. Comput. Algebra* 53 (2019), 49–52.

[14] Jamal Hossein Poor, Clemens G. Raab, and Georg Regensburger. 2018. Algorithmic operator algebras via normal forms in tensor rings. *J. Symbolic Comput.* 85 (2018), 247–274.

[15] F. Dell Kronewitter. 2001. Using noncommutative Gröbner bases in solving partially prescribed matrix inverse completion problems. *Linear Algebra Appl.* 338 (2001), 171–199.

[16] Viktor Levandovskyy and Leonard Schmitz. 2020. Algorithmic algebraic proofs of identities between not only matrices. In preparation.

[17] Teo Mora. 1994. An introduction to commutative and noncommutative Gröbner bases. *Theoret. Comput. Sci.* 134 (1994), 131–173.

[18] Teo Mora. 2016. *Solving polynomial equation systems. Vol. IV. Buchberger theory and beyond.* Cambridge University Press, Cambridge.

[19] Clemens G Raab, Georg Regensburger, and Jamal Hossein Poor. 2019. Formal proofs of operator identities by a single formal computation. *arXiv:1910.06165v1 [math.RA]* (2019).

[20] Markus Rosenkranz, Bruno Buchberger, and Heinz W. Engl. 2003. Solving linear boundary value problems via non-commutative Gröbner bases. *Appl. Anal.* 82 (2003), 655–675.

Integral Bases for P-Recursive Sequences*

Shaoshi Chen[a,b], Lixin Du[a,b,c], Manuel Kauers[c], and Thibaut Verron[c]
[a]KLMM, Academy of Mathematics and Systems Science, Chinese Academy of Sciences, Beijing 100190, China
[b]School of Mathematical Sciences, University of Chinese Academy of Sciences, Beijing 100049, China
[c]Institute for Algebra, Johannes Kepler University, Linz, A4040, Austria
schen@amss.ac.cn,dulixin17@mails.ucas.ac.cn
manuel.kauers@jku.at,thibaut.verron@jku.at

ABSTRACT

In an earlier paper, the notion of integrality known for algebraic number fields and fields of algebraic functions has been extended to D-finite functions. The aim of the present paper is to extend the notion to the case of P-recursive sequences. In order to do so, we formulate a general algorithm for finding all integral elements for valued vector spaces and then show that this algorithm includes not only the algebraic and the D-finite cases but also covers the case of P-recursive sequences.

CCS CONCEPTS

• **Computing methodologies → Algebraic algorithms**.

ACM Reference Format:
Shaoshi Chen[a,b], Lixin Du[a,b,c], Manuel Kauers[c], and Thibaut Verron[c]. 2020. Integral Bases for P-Recursive Sequences. In *International Symposium on Symbolic and Algebraic Computation (ISSAC '20), July 20–23, 2020, Kalamata, Greece.* ACM, New York, NY, USA, 8 pages. https://doi.org/10.1145/3373207.3404004

1 INTRODUCTION

Singularities play an essential role in algorithms for analyzing recurrence or differential equations, and for symbolic summation and integration. The "local" behaviour at a singularity typically gives rise to severe restrictions of the possible "global" shape of a solution, and such restrictions are exploited in the design of algorithms for finding such solutions. It is therefore important to have access to information about what is going on at the singularities. Integral bases provide such access.

For algebraic number fields and algebraic function fields, this is a classical notion. Let $k = C(x)$ be the field of rational functions in x over a field C and $K = k(\alpha)$ be an algebraic extension of k. Every element of K has a minimal polynomial $m \in C[x][y]$. An element of K is called *integral* if all its series expansions only involve terms with nonnegative exponents. The integral elements of K form a $C[x]$-submodule of K, which somehow plays the role in K that $\mathbb{Z}$ plays in $\mathbb{Q}$. An integral basis of K is a k-vector space basis of K which at the same time is a $C[x]$-module basis of the module of integral elements.

Trager [2–4, 17] used integral bases in his integration algorithm for algebraic functions. This was one of the motivations for introducing the notion of integral D-finite functions [14], which were then used not only for integration [5] but also for solving differential equations in terms of hypergeometric series [11, 12]. Also for D-finite functions, integrality is defined in terms of the exponents appearing in the series expansions. The goal of the present paper is to introduce a notion of integrality for the recurrence case. Our hope is that this work will subsequently be useful for the development of new summation algorithms.

A major difference between the differential case and the shift case is the fact that singularities are no longer isolated points $\alpha \in C$. Instead, as pointed out for instance in [19], singularities should be viewed as orbits $\alpha + \mathbb{Z} \in C/\mathbb{Z}$ consisting of some $\alpha \in C$ together with all elements of C that have integer distance to α. Instead of certain kinds of series solutions at α of differential operators or algebraic equations, we have to consider certain kinds of sequence solutions $\alpha+\mathbb{Z} \to C$ of a recurrence operator. This makes the matter considerably more technical.

We proceed in two stages. In the first stage (Sections 2 and 3), we give a general formulation of the algorithm proposed by van Hoeij for algebraic function fields [18] and adapted to D-finite functions by Kauers and Koutschan [14]. The general formulation applies to arbitrary valued vector spaces, and we identify the computational assumptions on which the correctness and termination arguments of the algorithms are based. In Section 4, we show how it indeed generalizes the previous algorithms. In the second stage (Section 5), we show how the general setting developed in Sections 2 and 3 can be applied to the shift case.

2 VALUE FUNCTIONS AND INTEGRAL ELEMENTS

In this section, we recall basic terminologies about valuations on fields and vector spaces from [10, 16, 20]. Let k be a field of characteristic zero and Γ be a totally ordered abelian group, written additively, and let $\Gamma_\infty = \Gamma \cup \{\infty\}$ in which $\alpha + \infty = \infty + \alpha = \infty$ for all $\alpha \in \Gamma_\infty$ and $\beta < \infty$ for all $\beta \in \Gamma$. A mapping $v : k \to \Gamma_\infty$ is called a *valuation* on k if for all $a, b \in k$,

(*i*) $v(a) = \infty$ if and only if $a = 0$;

*S. Chen was supported by the NSFC grants 11871067, 11688101 and the Fund of the Youth Innovation Promotion Association, CAS. L. Du was supported by the NSFC grant 11871067 and the Austrian FWF grant P31571-N32. M. Kauers was supported by the Austrian FWF grants F5004 and P31571-N32. T. Verron was supported by the Austrian FWF grant P31571-N32.

ISSAC '20, July 20–23, 2020, Kalamata, Greece

ACM ISBN 978-1-4503-7100-1/20/07…$15.00
https://doi.org/10.1145/3373207.3404004

(*ii*) $v(ab) = v(a) + v(b)$;
(*iii*) $v(a+b) \geq \min\{v(a), v(b)\}$.

The pair (k, v) is called a *valued field* and $v(k \setminus \{0\}) \subseteq \Gamma$ is called the *value group* of v. The set $O_{(k,v)} := \{a \in k \mid v(a) \geq 0\}$ forms a subring of k that is called the *valuation ring* of v.

Example 1. *A typical example of a valued field is the field of rational functions. Let C be a field of characteristic 0 and $\Gamma = \mathbb{Z}$. For any irreducible $p \in C[x]$ and $f \in C(x) \setminus \{0\}$, we can always write $f = p^m a/b$ for some $m \in \mathbb{Z}$ and $a, b \in C[x]$ with $\gcd(a, b) = 1$ and $p \nmid ab$. The valuation $v_p(f)$ of f at p is defined as the integer m. Set $v_p(0) = \infty$. Then $(C(x), v_p)$ is a valued field with $O_{(C(x), v_p)} = \{f \in C(x) \mid v_p(f) \geq 0\}$ being a local ring with its maximal ideal generated by p. The valuation v_∞ defined by $v_\infty(f) = \deg_x(b) - \deg_x(a)$ for any $f = a/b \in C(x)$ is called the valuation at ∞. Any valuation v on the field $C(x)$ is either v_∞ or v_p for some irreducible $p \in C[x]$ (see [6, Chapter 1, § 3] in the language of places). When $p = x - z$ with $z \in C$, we will write v_z instead of v_p. For $z \in C$, the field of formal Laurent series $C((x-z))$ admits a valuation $v_{(z)}$, defined as $v_{(z)}\left(\sum_{i \geq n} c_i (x-z)^i\right) = n$, where $c_n \neq 0$. Any $r \in C(x)$ admits a representation r_L in $C((x-z))$ with $v_z(r) = v_{(z)}(r_L)$.*

Definition 2. *Let V be a vector space over a valued field (k, v). A map* val : $V \to \Gamma_\infty$ *is called a* value function *on V if for all $x, y \in V$ and $a \in k$,*

(*i*) $\mathrm{val}(x) = \infty$ *if and only if* $x = 0$;
(*ii*) $\mathrm{val}(ax) = v(a) + \mathrm{val}(x)$;
(*iii*) $\mathrm{val}(x + y) \geq \min\{\mathrm{val}(x), \mathrm{val}(y)\}$.

The pair (V, val) *is called a* valued vector space *over k. An element $x \in V$ is said to be* integral *if* $\mathrm{val}(x) \geq 0$.

Remark 3. *Let U be any subspace of a valued vector space* (V, val). *Then the restriction of* val *on U is also a value function on U, which makes* (U, val) *a valued vector space.*

Proposition 4. *Let (k, v) be a valued field and* (V, val) *be a valued vector space over k. The set $O_{(V,\mathrm{val})} \subseteq V$ of all integral elements in V forms an $O_{(k,v)}$-module.*

Proof. For any $a, b \in O_{(k,v)}$ and $x, y \in O_{(V,\mathrm{val})}$, we have

$$\begin{aligned}\mathrm{val}(ax + by) &\geq \min\{\mathrm{val}(ax), \mathrm{val}(by)\} \\ &= \min\{v(a) + \mathrm{val}(x), v(b) + \mathrm{val}(y)\}.\end{aligned}$$

Since $v(a), v(b) \geq 0$ and $\mathrm{val}(x), \mathrm{val}(y) \geq 0$, we have $\mathrm{val}(ax + by) \geq 0$. So $ax + by \in O_{(V,\mathrm{val})}$. ∎

A k-vector space basis of a valued vector space (V, val) which is at the same time an $O_{(k,v)}$-module basis of $O_{(V,\mathrm{val})}$ is called a *(local) integral basis* with respect to val. Assume that the module $O_{(V,\mathrm{val})}$ has a local integral basis $\{x_1, \ldots, x_r\}$ and $x = a_1 x_1 + \cdots + a_r x_r \in V$. Then $\mathrm{val}(x) \geq 0$ if and only if $v(a_i) \geq 0$ for all $i = 1, \ldots, r$. When does a local integral basis exist and how to construct such a basis are the main problems we study in this paper. Value functions and integral bases for algebraic function fields have been extensively studied both theoretically [6, 9, 16] and algorithmically [17–19] and have also been extended to the D-finite case [14].

Example 5. *(See [16, Example 3.3]) Any finite dimensional k-vector space can be equipped with a valuation. More precisely, let V be a vector space over a valued field (k, v) of dimension r. Let $\{B_1, \ldots, B_r\}$ be a basis of V. Take values $\gamma_1, \ldots, \gamma_r$ in Γ and define* val : $V \to \Gamma \cup \{\infty\}$ *by for all $a_1, \ldots, a_r \in k$,*

$$\mathrm{val}\left(\sum_{i=1}^{r} a_i B_i\right) = \min\{\gamma_1 + v(a_1), \ldots, \gamma_r + v(a_r)\}.$$

It is easy to check that val *is a value function on V.*

Example 6. *Let C be an algebraically closed field of characteristic 0, $k = C(x)$ and v_z be the valuation of k at $z \in C$ as in Example 1. Then (k, v_z) is a valued field. Let $K = k(\beta)$ with β being algebraic over $C(x)$. Let $\beta_1, \ldots, \beta_r$ be all conjugates of β over k and each conjugate β_ℓ can be expanded as a Puiseux series around z. We extend the valuation v_z on k to a nonzero Puiseux series*

$$P = \sum_{i \geq 0} c_i (x - z)^{r_i},$$

defined as $v_z(P) = r_0$, where $c_i \in C$ with $c_0 \neq 0$ and $r_i \in \mathbb{Q}$ with $r_0 < r_1 < \cdots$. Any element $B \in K$ can be uniquely written as $B = f(\beta)$ with $f = f_0 + f_1 y + \ldots + f_{r-1} y^{r-1} \in k[y]$. The value function $\mathrm{val}_z : K \to \mathbb{Q} \cup \{\infty\}$ *is then defined by* $\mathrm{val}_z(B) = \min_{\ell=1}^{r} \{v_z(f(\beta_\ell))\}$. *In this setting, $O_{(K,\mathrm{val}_z)}$ is a free $C[x]$-module.*

Example 7. *Let C be an algebraically closed field of characteristic 0 and consider a linear differential operator $L = \ell_0 + \cdots + \ell_r D^r \in C(x)[D]$ with $\ell_r \neq 0$. The quotient module $V = C(x)[D]/\langle L \rangle$ is a $C(x)$-vector space with $1, D, \ldots, D^{r-1}$ as a basis. Its element 1 is a solution of L. If $z \in C$ is a so-called regular singular point of L [13], then there are r linearly independent solutions in the C-vector space generated by*

$$C[[[x - z]]] := \bigcup_{\nu \in C} (x - z)^\nu C[[x - z]][\log(x - z)].$$

Following [14], we construct a value function val_z on V as follows. First choose a function $\iota : C/\mathbb{Z} \times \mathbb{N} \to C$ with $\iota(\nu + \mathbb{Z}, j) \in \nu + \mathbb{Z}$ for every $\nu \in C$ and $j \in \mathbb{N}$, with

$$\iota(\nu_1 + \mathbb{Z}, j_1) + \iota(\nu_2 + \mathbb{Z}, j_2) - \iota(\nu_1 + \nu_2 + \mathbb{Z}, j_1 + j_2) \geq 0$$

for every $\nu_1, \nu_2 \in C$ and $j_1, j_2 \in \mathbb{N}$, and with $\iota(\mathbb{Z}, 0) = 0$. This function picks from each $\mathbb{Z}$-equivalence class in C a canonical representative.

Using this auxiliary function, the valuation $\mathrm{val}_z(t)$ of a term $t := (x-z)^{\nu+i} \log(x-z)^j$ is the integer $\nu + i - \iota(\nu + i, j)$, and the valuation $\mathrm{val}_z(f)$ of a series $f \in C[[[x-z]]]$ is the minimum of the valuations of all the terms appearing in it (with nonzero coefficients). The valuation of 0 is defined as ∞.

The value function $\mathrm{val}_z(\cdot) : V \to \mathbb{Z} \cup \{\infty\}$ is then defined as the smallest valuation of a series $B \cdot f$, when f runs through all solutions of L. We now check that the function val_z is indeed a value function.

(*i*) *Let $B \in V$. Clearly if $B = 0$, $\mathrm{val}_\alpha(B) = \infty$ for all $\alpha \in C$. Conversely, assume that $\mathrm{val}_\alpha(B) = \infty$, then by definition $\mathrm{val}_\alpha(B \cdot f) = \infty$ and so $B \cdot f = 0$ for all $f \in \mathrm{Sol}_\alpha(L)$, which implies that the dimension of the solution space of B is at least r. But the order of B is less than r, and the dimension of the solution space of a nonzero operator cannot exceed its order, so it follows that $B = 0$.*
(*ii*) *For any $a \in C(x) \subseteq C[[[x-\alpha]]]$ and $f \in C[[[x-\alpha]]]$, the valuation of af is the sum of the valuations of a and f by definition. Then for any $B \in V$, $\mathrm{val}_\alpha(aB) = \min_{f \in \mathrm{Sol}_\alpha(L)} \{\mathrm{val}_\alpha(aB \cdot f)\}$, which is then equal to $v_\alpha(a) + \mathrm{val}_\alpha(B)$.*

(*iii*) *By* $\mathrm{val}_\alpha((B_1+B_2)\cdot f)) \geq \min\{\mathrm{val}_\alpha(B_1\cdot f), \mathrm{val}_\alpha(B_1\cdot f)\}$ *for all* $f \in \mathrm{Sol}_\alpha(L)$*, we have for any* $B_1, B_2 \in V$*,*

$$\mathrm{val}_\alpha(B_1+B_2) \geq \min\{\mathrm{val}_\alpha(B_1), \mathrm{val}_\alpha(B_2)\}.$$

When $\Gamma = \mathbb{Z}$, the valued field (k, v) can be endowed with a topology. We summarize here the relevant constructions, more details can be found in [15, Chapter 2]. For $a \in k$, let $|a| = \mathrm{e}^{-v(a)}$. The properties of the valuation ensure that $|\cdot|$ is an absolute value, called the v-adic absolute value. This absolute value defines a topology on k, in which elements are "small" if their valuation is "large".

Recall that a sequence of elements $(c_n) \in k^{\mathbb{N}}$ is said to be Cauchy if for each $\epsilon > 0$, there exists $N \in \mathbb{N}$ such that for every $m, n > N$, $|c_m - c_n| < \epsilon$, or, equivalently, if for each $M \in \mathbb{Z}$, there exists $N \in \mathbb{N}$ such that for every $m, n > N$, $v(c_m - c_n) > M$. The field k is said to be complete if every Cauchy sequence is convergent.

The completion of k is a minimal field extension k_v which is complete. It can be constructed as follows. As a set, let k_v be the set of all Cauchy sequences in k, modulo the equivalence relation $(c_n) \equiv (d_n) \Leftrightarrow (c_n - d_n)$ converges to 0 at infinity. The field k is contained in k_v via the constant sequences. Ring operations on k extend to k_v component-wise, and make k_v a field. The valuation on k extends to k_v by taking the limit of the valuations of the terms of the sequences, we use the same letter v for that valuation.

An important feature of the topology on k and k_v is that the v-adic absolute value is ultrametric: it satisfies the stronger triangular condition $|a+b| \leq \max(|a|, |b|)$. In particular, any series $\sum_{n=0}^{\infty} a_n$ with $a_n \in k_v$ and $|a_n| \to 0$ is convergent in k_v.

Example 8. *The completion of* $C(x)$ *w.r.t. the valuation* v_z *is* $C((x-z))$*, and its completion w.r.t.* v_∞ *is* $C((1/x))$.

These definitions extend naturally to a valued k-vector space. Just like in the case of fields, the hypotheses *(i)* and *(iii)* of Definition 2 ensure that we can define a norm on V by setting $||v|| = \mathrm{e}^{-\mathrm{val}(v)}$. This turns V into a topological vector space: addition and scalar multiplication are continuous.

Part *(ii)* of Definition 2 further ensures that $||cv|| = |c| \cdot ||v||$ for $c \in k$, $v \in V$. In particular, if a sequence $(a_n)_{n\in\mathbb{N}}$ in k converges to 0, then $(a_n v)_{n\in\mathbb{N}}$ converges to 0 in V.

More generally, if $B_1, \ldots, B_r \in V$ and $(a_n^{(1)}), \ldots, (a_n^{(r)})$ are sequences in k converging to $a_\infty^{(1)}, \ldots, a_\infty^{(r)}$, respectively, then the sequence $(a_n^{(1)}B_1 + \cdots + a_n^{(r)}B_r)$ in V converges to $a_\infty^{(1)}B_1 + \cdots + a_\infty^{(r)}B_r$.

Let V_v be the k_v-vector space obtained from scalar extension of V. If V is finite dimensional and $B_1, \ldots, B_r$ is a basis, V_v can be seen as the k_v-vector space generated by $B_1, \ldots, B_r$, identifying its elements with elements of V whenever possible, and it is the completion of V with respect to the above topology.

Remark 9. *The inequality* $\dim_{k_v} V_v \leq \dim_k V$ *always holds, but it may happen that the inequality is strict. For example, consider* $C((x))$ *as a* $C(x)$*-vector space, with valuation* $v = v_0$*, and let* V *be a* r*-dimensional sub-vector space of* $C((x))$*. Then* $V_v = C((x))$ *has dimension* 1 *over* $C((x))$.

3 COMPUTING INTEGRAL BASES

In this section, we present a general algorithm for computing local and global integral bases of valued vector spaces and conditions on the termination of this algorithm.

3.1 The local case

Given a valued field (k, v), a basis of a k-vector space V of dimension r, and a value function val on V, our goal is to compute a local integral basis of V if it exists. The algorithm described below is based on the algorithm given by van Hoeij [18] for computing integral bases of algebraic function fields. It also covers the adaption by Kauers and Koutschan to D-finite functions [14]. For simplicity, we restrict to the case $\Gamma = \mathbb{Z}$.

For the algorithm to apply in the general setting, we need to make the following assumptions.

(A) Arithmetic in k and V is constructive, and v and val are computable.
(B) We know an element $x \in k$ with $v(x) = 1$.
(C) For any given $B_1, \ldots, B_d \in V$, we can find $\alpha_1, \ldots, \alpha_{d-1}$ in k such that
$$\mathrm{val}(\alpha_1 B_1 + \cdots + \alpha_{d-1}B_{d-1} + B_d) > 0$$
or prove that no such α_i's exist.
(D) The completion V_v of V has dimension r.

Algorithm 10. *INPUT: a* k*-vector space basis* $B_1, \ldots, B_r$ *of* V
OUTPUT: a local integral basis of V *w.r.t.* val

1 *for* $d = 1, \ldots, r$*, do:*
2 *replace* B_d *by* $x^{-\mathrm{val}(B_d)}B_d$.
3 *while there exist* $\alpha_1, \ldots, \alpha_{d-1} \in k$ *such that*
$$\mathrm{val}(\alpha_1 B_1 + \cdots + \alpha_{d-1}B_{d-1} + B_d) > 0,$$
4 *choose such* $\alpha_1, \ldots, \alpha_{d-1}$.
5 *replace* B_d *by* $x^{-1}(\alpha_1 B_1 + \cdots + \alpha_{d-1}B_{d-1} + B_d)$.
6 *return* $B_1, \ldots, B_r$.

Theorem 11. *Alg. 10 is correct.*

Proof. We show by induction on d that for every $d = 1, \ldots, r$, the output elements $B_1, \ldots, B_d$ form a local integral basis for the subspace of V generated by the input elements $B_1, \ldots, B_d$. From the updates in lines 2 and 5, it is clear that the output elements generate the same subspace, so the only claim to be proven is that they are also module generators for the module of integral elements.

For $d = 1$, line 2 ensures that $\mathrm{val}(B_1) = 0$, and no further change is going to happen in the while loop. When $\mathrm{val}(B_1) = 0$, then the integral elements of the subspace generated by B_1 are precisely the elements uB_1 for $u \in k$ with $v(u) \geq 0$, so B_1 is an integral basis.

Now assume that d is such that $B_1, \ldots, B_{d-1}$ is an integral basis, and let $B_d \in V$. After executing line 2, we may assume $\mathrm{val}(B_d) \geq 0$. After termination of the while loop, we know that there are no $\alpha_1, \ldots, \alpha_{d-1} \in k$ such that $\mathrm{val}(\alpha_1 B_1 + \cdots + \alpha_{d-1}B_{d-1} + B_d) > 0$. Let $\alpha_1, \ldots, \alpha_d \in k$ be such that $A = \alpha_1 B_1 + \cdots + \alpha_d B_d$ is an integral element. We have to show that $v(\alpha_i) \geq 0$ for $i = 1, \ldots, d$.

We cannot have $v(\alpha_d) < 0$, otherwise, $\mathrm{val}(\alpha_d^{-1}A) > 0$, which would contradict the termination condition of the while loop. Thus $v(\alpha_d) \geq 0$. But then, $\mathrm{val}(\alpha_d B_d) \geq 0$, so $A - \alpha_d B_d$ is also integral. Since $A - \alpha_d B_d$ is in the k-subspace generated by $B_1, \ldots, B_{d-1}$ and the latter is an integral basis by induction hypothesis, it follows that $v(\alpha_i) \geq 0$ for $i = 1, \ldots, d-1$. ∎

We prove that under the assumptions (A), (B), (C), the termination of Alg. 10 is equivalent to assumption (D). It is moreover

equivalent to the the existence of a discriminant function, which is defined as follows and generalizes the corresponding notion for fields of algebraic numbers or functions. With such a function at hand, we can also bound the number of iterations of the main loop.

Definition 12. *Let* (V, val) *be a valued vector space of finite dimension* r *over a valued field* (k, v) *with the value group* $\mathbb{Z}$. *Let* $x \in k$ *be such that* $v(x) = 1$ *and* $\mathbb{B}_V$ *denote the set of all bases of* V. *A map* $\mathrm{Disc} : \mathbb{B}_V \to \mathbb{Z}$ *is called a* discriminant function *on* V *if for every basis* $B_1, \dots, B_r$ *of* V, *we have*

(i) $\gamma := \mathrm{Disc}(\{B_1, \dots, B_r\}) \geq 0$ *if all the* B_i*'s are integral in* V
(ii) for all $\alpha_1, \dots, \alpha_{d-1} \in k$ *with* $d \leq r$,

$$\mathrm{Disc}(B_1, \dots, B_{d-1}, \alpha_1 B_1 + \cdots + \alpha_{d-1} B_{d-1} + B_d, B_{d+1}, \dots, B_r) = \gamma$$

(iii) $\mathrm{Disc}(B_1, \dots, B_{d-1}, x^{-1} B_d, B_{d+1}, \dots, B_r) < \gamma$.

Theorem 13. *Let* (V, val) *be a valued vector space of finite dimension* r *over a valued field* (k, v) *with the value group* $\mathbb{Z}$. *Then the following four statements are equivalent under the hypotheses* (A), (B), (C)*:*

(*a*) *There is a local integral basis of* V *w.r.t.* val.
(*b*) *There is a discriminant function* $\mathrm{Disc} : \mathbb{B}_V \to \mathbb{Z}$.
(*c*) *Alg. 10 terminates.*
(*d*) *The topological assumption* (D) *on* V *is satisfied.*

Proof. (*c*) $\Rightarrow$ (*a*) follows from Theorem 11.

(*a*) $\Rightarrow$ (*b*): Given a local integral basis $\{C_1, \dots, C_r\}$ and a basis $B = \{B_1, \dots, B_r\}$ of V with $B_i = \sum_{j=1}^{r} b_{i,j} C_j$ for some $b_{i,j} \in k$, the discriminant function can be defined as

$$\mathrm{Disc}(B) := v(\det((b_{i,j})_{i,j=1}^{r})).$$

(*b*) $\Rightarrow$ (*c*): By assumption (B), there exists $x \in k$ such that $v(x) = 1$. Let $\{B_1, \dots, B_r\}$ be any basis of V over k. We may always assume that $\mathrm{val}(B_i) = 0$ by replacing B_i by $x^{-\mathrm{val}(B_i)} B_i$ for all i. It suffices to show that Alg. 10 terminates on $\{B_1, \dots, B_r\}$. Let $\gamma = \mathrm{Disc}(\{B_1, \dots, B_r\}) \in \mathbb{N}$. At any intermediate step of Alg. 10, $B_1, \dots, B_r$ are always integral and form a basis of V. If α_i's exist in the while loop, γ decreases strictly. So there can be at most γ basis updates, which implies that Alg. 10 terminates.

(*d*) $\Rightarrow$ (*c*): Assume that for some $d \in \{1, \dots, r\}$, the inner loop does not terminate. Let $B_{d,i}$ be the value of B_d before entering the ith iteration, and let $\tilde{B}_{d,i} = x^i B_{d,i}$. The operation for computing $B_{d,i}$ from $B_{d,i-1}$ (step 5) ensures that for all i, $\mathrm{val}(B_{d,i}) \geq 0$ and $\mathrm{val}(\tilde{B}_{d,i}) \geq i$. For all $i \in \mathbb{N}$, there exists $a_{j,i} \in k$ for $j \in \{0, \dots, d-1\}$ such that

$$\tilde{B}_{d,i} = x^i \cdot \frac{1}{x} \left(B_{d,i-1} + \sum_{j=0}^{d-1} a_{j,i} B_j \right) = \tilde{B}_{d,i-1} + x^{i-1} \sum_{j=0}^{d-1} a_{j,i} B_j$$

and $B_{d,i}$ has valuation 0. We can unroll the sum as

$$\tilde{B}_{d,i} = B_{d,0} + \sum_{j=0}^{d-1} \left(\sum_{\ell=0}^{i-1} x^\ell a_{j,\ell} \right) B_j.$$

Viewing this equality in V_v and taking the limit as $i \to \infty$ yields

$$\tilde{B}_{d,\infty} := \lim_{i \to \infty} \tilde{B}_{d,i} = B_{d,0} + \sum_{j=0}^{d-1} \left(\sum_{\ell=0}^{\infty} x^\ell a_{j,\ell} \right) B_j.$$

Furthermore, $\tilde{B}_{d,\infty}$ has valuation ∞, so it is zero and

$$B_{d,0} = -\sum_{j=0}^{d-1} \left(\sum_{\ell=0}^{\infty} x^\ell a_{j,\ell} \right) B_j \quad \text{in } V_v.$$

But by hypothesis (D), V_v has dimension r, so $B_1, \dots, B_r$ must be linearly independent over k_v too, a contradiction. So the loop terminates.

(*c*) $\Rightarrow$ (*d*): Let $B_1, \dots, B_r$ be the output of Alg. 10. If the dimension falls, then there exist some $a_i \in k_v$ and $d \leq r$ such that $B_d = \sum_{i=1}^{d-1} a_i B_i$. For each i, let $a_{i,j}$ be a sequence in k converging to a_i. Let $\tilde{B}_{d,j} = B_d - \sum_{i=1}^{d-1} a_{i,j} B_i$. By assumption, $\tilde{B}_{d,j}$ goes to 0 when j goes to infinity, so its valuation goes to infinity. We can assume $\mathrm{val}(\tilde{B}_{d,j}) \geq j$. Then $B_{d,j} := x^{-j} \tilde{B}_{d,j}$ is an infinite sequence such that Alg. 10 does not terminate, a contradiction. ∎

3.2 The global case

In a next step, we seek integral bases with respect to several valuations simultaneously. Instead of a single valuation $\mathrm{val} : V \to \mathbb{Z} \cup \{\infty\}$, we have a set of valuations $v_z : k \to \mathbb{Z} \cup \{\infty\}$ $(z \in Z)$ and a set of value functions $\mathrm{val}_z : V \to \mathbb{Z} \cup \{\infty\}$ $(z \in Z)$ and want to find a vector space basis $B_1, \dots, B_r$ of V that is also an $O_{(k, v_z)}$-module basis of $O_{(V, \mathrm{val}_z)}$ for every $z \in Z$. The idea is to apply Alg. 10 repeatedly. In order to make this work, we impose the following additional assumptions:

(B′) For every $z \in Z$ we know an element $x_z \in k$ with $v_z(x_z) = 1$ and $v_\zeta(x_z) = 0$ for all $\zeta \in Z \setminus \{z\}$.
(C′) For every $z \in Z$ and any given $B_1, \dots, B_d \in V$, we can compute $\alpha_1, \dots, \alpha_{d-1} \in k$ with $v_\zeta(\alpha_i) \geq 0$ for all i and all $\zeta \in Z \setminus \{z\}$ such that

$$\mathrm{val}_z(\alpha_1 B_1 + \cdots + \alpha_{d-1} B_{d-1} + B_d) > 0,$$

or prove that no such α_i's exist.
(D′) For every $z \in Z$, the completion V_{v_z} of V has dimension r.
(E) We know a finite set $Z_0 \subseteq Z$ and a basis $B_1, \dots, B_r$ of V that is an integral basis for all $z \in Z \setminus Z_0$.

Under these circumstances, we can proceed as follows.

Algorithm 14. *INPUT: a* k*-vector space basis* $B_1, \dots, B_r$ *of* V *which is an integral basis for all* $z \in Z \setminus Z_0$
OUTPUT: an integral basis for all $z \in Z$

1 *for all* $z \in Z_0$, *do:*
2 *apply Alg. 10 to* $B_1, \dots, B_r$, *using* v_z, val_z *and* x_z *in place of* v, val, *and* x, *and ensuring in step 3 that* $v_\zeta(\alpha_i) \geq 0$ *for all* i *and all* $\zeta \in Z$.
3 *replace* $B_1, \dots, B_r$ *by the output of Alg. 10.*
4 *return* $B_1, \dots, B_r$.

Theorem 15. *Alg. 14 is correct.*

Proof. We only have to show that one application of Alg. 10 does not destroy the integrality properties arranged in earlier calls. To see that this is the case, consider the effects of steps 2 and 5 with respect to a value function other than val_z. If val_ζ is such a function, then by (B′), we have $v_\zeta(x_z) = 0$, so $B_1, \dots, B_{d-1}, B_d$ and $B_1, \dots, B_{d-1}, x_z^e B_d$ generate the same $O_{(k, v_\zeta)}$-module, for any $e \in \mathbb{Z}$. Hence this step is safe. Likewise, by the choice of the α_i

in step 5, $\{B_1, \dots, B_{d-1}, B_d\}$ and $\{B_1, \dots, B_{d-1}, B_d + \sum_{i=1}^{d-1} \alpha_i B_i\}$ generate the same $O_{(k,\nu_\zeta)}$-module. So this step is safe too. ∎

3.3 Avoiding constant field extensions

We shall discuss one more refinement, which also appears already in earlier versions of the algorithm [11, 14, 18]. In applications, we typically have $k = \bar{C}(x)$ where C is a field and $\bar{C}$ is an algebraic closure of C, with the usual valuation ν_z for $z \in \bar{C}$ (see Example 1). For this valuation, $x_z = x - z$ is a canonical choice.

For theoretical purposes it is advantageous to work with vector spaces over k, but computationally it would be preferable to work with coefficients in $C(x)$ rather than $\bar{C}(x)$. It is therefore desirable to ensure that the basis elements returned by Alg. 14 have coefficients in $C(x)$ with respect to the input basis.

Note that in this setting, we have the following properties:

Lemma 16. (1) *For every automorphism $\sigma\colon \bar{C} \to \bar{C}$ leaving C fixed, for every $z \in Z$, and for every $u \in \bar{C}(x)$, we have $\nu_z(u) = \nu_{\sigma(z)}(\sigma(u))$, where $\sigma(u)$ is the element of $\bar{C}(x)$ obtained by applying σ to the coefficients of u.*

(2) *For every $u \in \bar{C}(x) \setminus \{0\}$, and for every $z \in Z$, u admits a unique Laurent series expansion*

$$u = c_z(x-z)^{\nu_z(u)} + (x-z)^{\nu_z(u)+1} r$$

with $c_z \in \bar{C} \setminus \{0\}$ and $\nu_z(r) \geq 0$.

The constant c_z in item 2 is called the *leading coefficient* of u.

The second property of the lemma ensures that the coefficients $\alpha_1, \dots, \alpha_{d-1} \in \bar{C}(x)$ from (C) and (C′) can be chosen in $\bar{C}$. Indeed, we can replace α_i by its leading coefficient if $\nu_z(\alpha_i) = 0$ and by zero otherwise, because whenever $\alpha_1, \dots, \alpha_{d-1} \in \bar{C}(x)$ is a solution and $\beta_1, \dots, \beta_{d-1} \in \bar{C}(x)$ are arbitrary with $\nu_z(\beta_i) \geq 1$ for all i, then also $\alpha_1 + \beta_1, \dots, \alpha_{d-1} + \beta_{d-1}$ is a solution.

If we restrict $\alpha_1, \dots, \alpha_{d-1}$ to $\bar{C}$, then there can be at most one solution whenever we seek a solution in step 3 of Alg. 10, because the difference of any two distinct solutions would be a nontrivial $\bar{C}$-linear combination of $B_1, \dots, B_{d-1}$, and by the invariant of the outer loop, $B_1, \dots, B_{d-1}$ already form an integral basis of the k-subspace they generate.

We shall adopt the following last assumption, stating that we can apply σ on V:

(F) We know a basis $B_1, \dots, B_r$ as in (E) such that for every automorphism $\sigma\colon \bar{C} \to \bar{C}$ fixing C, and for all $\alpha_1, \dots, \alpha_r \in k$, we have $\mathrm{val}_z(\alpha_1 B_1 + \cdots + \alpha_r B_r) = \mathrm{val}_{\sigma(z)}(\sigma(\alpha_1)B_1 + \cdots + \sigma(\alpha_r)B_r)$.

Using this assumption, it can further be shown that the unique elements $\alpha_1, \dots, \alpha_{d-1} \in \bar{C}$ from (C′) must in fact belong to $C(z)$ (if they exist at all). This is because if some α_i were in $\bar{C} \setminus C(z)$, then there would be some automorphism $\sigma\colon \bar{C} \to \bar{C}$ fixing $C(z)$ but moving α_i, and (F) would imply that $\sigma(\alpha_1), \dots, \sigma(\alpha_d)$ would be another solution to (C′), in contradiction to the uniqueness.

In order to ensure that the output elements of Alg. 14 are $C(x)$-linear combinations of the input elements, we adjust Alg. 10 as follows. Let G be the Galois group of $C(z)$ over C. In step 2, instead of replacing B_d by $x_z^{-\mathrm{val}_z(B_d)}$, we replace B_d by

$$\left(\prod_{\sigma\in G} \sigma(x_z)^{-\mathrm{val}_z(B_d)}\right) B_d.$$

Note that $\prod_{\sigma\in G} \sigma(x_z) = \prod_{\sigma\in G} \sigma(x-z)$ is the minimal polynomial of z in $C[x]$.

In step 5 of Alg. 10, we choose $\alpha_1, \dots, \alpha_{d-1} \in C(z)$ (if there are any), and instead of replacing B_d by $x_z^{-1}(\alpha_1 B_1 + \cdots + \alpha_{d-1}B_{d-1} + \alpha_d B_d)$ (with $\alpha_d = 1$), we replace B_d by

$$A := \sum_{i=1}^{d} \left(\sum_{\sigma\in G} \sigma\left(\frac{\alpha_i}{x_z}\right)\right) B_i.$$

Proposition 17. *When the steps 2 and 5 of Alg. 10 are adjusted as indicated, Alg. 14 returns an integral basis of V whose elements are $C(x)$-linear combinations of the input elements.*

Proof. By Galois theory, $\prod_{\sigma\in G} \sigma(x_z) = \prod_{\sigma\in G} \sigma(x-z) \in C(x)$ and $\tilde{\alpha}_i := \sum_{\sigma\in G} \sigma(\alpha_i/(x-z)) \in C(x)$ for every i. Therefore, all updates in the modified Alg. 10 replace certain basis elements by $C(x)$-linear combinations of basis elements.

It remains to show that the output is an integral basis for all $z \in Z$. To see this, we have to check the effect of Alg. 10 concerning val_z and concerning val_ζ for $\zeta \in Z \setminus \{z\}$. For the latter, we distinguish the case when ζ is conjugate to z and when it is not.

By part 1 of Lemma 16, for all $\zeta \in Z$ that are not conjugate to z we have $\nu_\zeta(\tilde{\alpha}_i) \geq 0$ for $i = 1, \dots, d-1$ and $\nu_\zeta(\tilde{\alpha}_d) = 0$. Therefore, $B_1, \dots, B_{d-1}$ and A generate the same $O_{(k,\nu_\zeta)}$-module as $B_1, \dots, B_{d-1}$ and B_d, for every $\zeta \in Z$ that is not conjugate to z. This settles the case when ζ is not conjugate to z.

Next, observe that $\mathrm{val}_z(x_z^{-1}(\alpha_1 B_1 + \cdots + \alpha_d B_d)) \geq 0$ by the assumptions on $x_z, \alpha_1, \dots, \alpha_d$. Moreover, by part 1 of Lemma 16, $\nu_z(\sigma(x-z)) = \nu_{\sigma^{-1}(z)}(x-z) = 0$ for every $\sigma \in G \setminus \{\mathrm{id}\}$, and $\nu_z(\sigma(\alpha_i)) = \nu_{\sigma^{-1}(z)}(\alpha_i) \geq 0$ because $\nu_\zeta(\alpha_i) \geq 0$ for all ζ. Therefore $\mathrm{val}_z(\sigma(x_z^{-1})(\sigma(\alpha_1)B_1 + \cdots + \sigma(\alpha_d)B_d) \geq 0$ for every $\sigma \in G \setminus \{\mathrm{id}\}$. It follows that

$$\mathrm{val}_z(A) \geq \max_{\sigma\in G} \mathrm{val}_z\left(\sum_{i=1}^{d} \sigma\left(\frac{\alpha_i}{x-z}\right) B_i\right) \geq 0.$$

Moreover, since $\alpha_d = 1$ and $\mathrm{val}_{\sigma(z)}(x_z) = 0$ for all $\sigma \neq \mathrm{id}$, we have that $B_1, \dots, B_{d-1}$ and A generate the same $O_{(k,\nu_z)}$-module as $B_1, \dots, B_{d-1}$ and $x_z^{-1}(\alpha_1 B_1 + \cdots + \alpha_d B_d)$. This settles the concern about val_z.

Finally, if ζ is conjugate to z, say $\zeta = \sigma(z)$ for some automorphism $\sigma \in G$, then $\mathrm{val}_\zeta(A) = \mathrm{val}_\zeta(\sigma(A)) = \mathrm{val}_z(A) \geq 0$ by assumption (F), because A is a $C(x)$-linear combination of the original basis elements. So A belongs to the $O_{(k,\nu_\zeta)}$-module of all integral elements (w.r.t. val_ζ) of the subspace generated by $B_1, \dots, B_d$ in V, so we are not making the module larger than we should. Conversely, the old B_d belongs to the $O_{(k,\nu_\zeta)}$-module generated by $B_1, \dots, B_{d-1}$ and A, so by updating B_d to A, the module generated by $B_1, \dots, B_d$ does not become smaller. ∎

Informally, what happens by taking the sums over the Galois group is that the algorithm working locally at z simultaneously works at all its conjugates. If for a certain z, the set Z_0 contains z as well as its conjugates, it is fair (and advisable) to discard all the conjugates from Z_0 and only keep z. More precisely, the whole process requires only knowing the minimal polynomial of z in $C[x]$, so for applications where the set Z_0 is computed as the set of roots of some polynomial $p \in C[x]$, the algorithms can proceed with the factors of p instead of all its roots.

4 THE ALGEBRAIC AND D-FINITE CASES

We will see below how the algorithms in [14, 18] for computing integral bases are special cases of the general formulation in Section 3. Let C be a computable subfield of $\mathbb{C}$ and $k = \bar{C}(x)$ with a valuation v_z for $z \in \bar{C}$. The value function val_z on $V = k(\beta)$ with $\beta \in \overline{C(x)}$ is defined in Example 6 and on $V = \bar{C}(x)[D]/\langle L\rangle$ with $L \in C[x][D]$ and all local exponents ν of solutions contained in C is defined in Example 7. We show that the assumptions imposed on value functions in Section 3 are fulfilled in the algebraic and D-finite settings. Note that (B), (C), (D) are subsumed in (B′), (C′), (D′), respectively.

- (A) It is assumed that C is a computable field, so it is clear that arithmetic in $\bar{C}(x)$ and V are computable, and that v_z on $\bar{C}(x)$ is also computable. The value functions val_z for algebraic and D-finite functions are computable since we can determine first few terms of Puiseux or generalized series solutions by algorithms in [8, 13].
- (B′) For every $z \in Z$, we can take $x_z = x - z$ such that $v_z(x_z) = 1$ and $v_\zeta(x_z) = 0$ for all $\zeta \in Z \setminus \{z\}$.
- (C′) Done in [14, Section 4].
- (D′) Clear.
- (E) In the algebraic case, we can choose as Z_0 the set of singularities of $\beta \in \overline{C(x)}$ which is clearly a finite set. In the D-finite case, we can choose as Z_0 the set of zeros of ℓ_r which are the only possible singularities by [14, Lemma 9].
- (F) If α and $\bar{\alpha}$ are conjugates, let σ be an element of the Galois group of $\bar{C}/C$ such that $\bar{\alpha} = \sigma(\alpha)$. In particular $\sigma(L) = L$ and $\sigma(B) = B$. For all $i \in \{1, \dots, r\}$, $\sigma(f_{\alpha,i}) \in \bar{C}[[[x - \bar{\alpha}]]]$ is a solution of $\sigma(L) = L$. Since σ is an automorphism, the $\sigma(f_{\alpha,i})$ form a fundamental system of L in $\bar{C}[[[x - \bar{\alpha}]]]$. For all $i \in \{1, \dots, r\}$, $B \cdot \sigma(f_{\alpha,i}) = \sigma(B) \cdot \sigma(f_{\alpha,i}) = \sigma(B \cdot f_{\alpha,i})$, and the equality of the valuations follows. In the algebraic case, this equality follows from the property of Duval's rational Puiseux series (see the remarks on [8, page 120]).

The termination of the general algorithm 10 in the algebraic and D-finite cases have been shown in [14, 18] by using classical discriminants and generalized Wronskians. The discriminant functions in these cases can be taken as the compositions of the valuation v_z and these functions. More precisely, for a basis $B_1, \dots, B_r$ of $V = k(\beta)$, the discriminant function Disc in the algebraic setting is defined as

$$\mathrm{Disc}(\{B_1, \dots, B_r\}) = v_z(\det(\mathrm{Tr}(B_i B_j))),$$

where Tr is the trace map from V to $\bar{C}(x)$. If $B_1, \dots, B_r$ are integral, $\det(\mathrm{Tr}(B_i B_j)) \in \bar{C}[x]$ and then $\mathrm{Disc}(\{B_1, \dots, B_r\}) \in \mathbb{N}$. Let $\alpha_1, \dots, \alpha_{d-1} \in k$, replacing B_d by $\alpha_1 B_1 + \cdots + \alpha_{d-1} B_{d-1} + B_d$ is equivalent to multiplying the matrix $(\mathrm{Tr}(B_i B_j))$ left and right by elementary transformation matrices with determinant 1, so the determinant (and its valuation) are constant. Similarly, replacing B_d by $(x - z)^{-1} B_d$ is equivalent to multiplying the matrix $(\mathrm{Tr}(B_i B_j))$ left and right by a matrix with determinant $(x - z)^{-1}$, so the discriminant decreases by 2. So Disc is indeed a discriminant function on $k(\beta)$.

In the case of D-finite functions, for any basis $B = \{B_1, \dots, B_r\}$ of $V = \bar{C}(x)[D]/\langle L\rangle$ and fundamental series solutions $b_1, \dots, b_r \in \bar{C}[[[x - z]]]$ of L , the generalized Wronskian is defined as

$$\mathrm{wr}_{L,z}(B) := \det(((B_i \cdot b_j))_{i,j=1}^{r}) \in \bar{C}[[[x - z]]].$$

The discriminant function Disc can be defined as the valuation of $\mathrm{wr}_{L,z}(B)$ at z. By the proof of Theorem 18 in [14], Disc is indeed a discriminant function on $\bar{C}(x)[D]/\langle L\rangle$.

5 THE P-RECURSIVE CASE

5.1 Solution Spaces

For the case of recurrence operators, we use a setting that has already been used for instance in [1, 7, 19] in the context of finding hypergeometric solutions. The relevant parts of the construction are summarized in this section. We consider the Ore algebra $C(x)[S]$ with the commutation rule $Sx = (x + 1)S$. We fix an operator $L = \ell_0 + \ell_1 S + \cdots + \ell_r S^r \in C(x)[S]$ with $\ell_0, \ell_r \neq 0$, and we consider the vector space $V = \bar{C}(x)[S]/\langle L\rangle$, where $\langle L\rangle = \bar{C}(x)[S]L$. The operator L acts on a sequence $f : \alpha + \mathbb{Z} \to \bar{C}$ through $(L \cdot f)(z) := \ell_0(z) f(z) + \cdots + \ell_r(z) f(z + r)$ for all $z \in \alpha + \mathbb{Z}$. This action turns $\bar{C}^{\alpha+\mathbb{Z}}$ into a (left) $C[x][S]$-module, but not to a (left) $C(x)[S]$-module, because a sequence $f : \alpha + \mathbb{Z} \to \bar{C}$ cannot meaningfully be divided by a polynomial which has a root in $\alpha + \mathbb{Z}$. In order to obtain a $C(x)[S]$-module, consider the space $\bar{C}((q))^{\alpha+\mathbb{Z}}$ of all sequences $f : \alpha + \mathbb{Z} \to \bar{C}((q))$ whose terms are Laurent series in a new indeterminate q, and define the action of $L = \ell_0 + \cdots + \ell_r S^r \in C(x)[S]$ on a sequence $f : \alpha + \mathbb{Z} \to \bar{C}((q))$ through $(L \cdot f)(z) := \ell_0(z + q) f(z) + \cdots + \ell_r(z + q) f(z + r)$ for all $z \in \alpha + \mathbb{Z}$. Note that no $\ell_i \in C(x)$ can have a pole at $z + q$ for any $z \in \alpha + \mathbb{Z}$ when $\alpha \in \bar{C}$ and $q \notin \bar{C}$.

For a fixed operator $L = \ell_0 + \cdots + \ell_r S^r \in C[x][S]$ with $\ell_0, \ell_r \neq 0$, the set $\mathrm{Sol}(L) := \{ f : \alpha + \mathbb{Z} \to \bar{C}((q)) : L \cdot f = 0 \}$ is a $\bar{C}((q))$-vector space of dimension r. Indeed, a basis $b_1, \dots, b_r$ is given by specifying the initial values $b_i(\alpha + j) = \delta_{i,j}$ for $i, j = 1, \dots, r$ and observing that the operator L uniquely extends any choice of initial values indefinitely to the left as well as to the right. The reason is again that $q \notin \bar{C}$ implies $\ell_0(z + q), \ell_r(z + q) \neq 0$ for every $z \in \alpha + \mathbb{Z}$, so there is no danger that computing a certain sequence term $b_i(z)$ from $b_i(z + 1), \dots, b_i(z + r)$ or from $b_i(z - 1), \dots, b_i(z - r)$ could produce a division by zero. Instead of a division by zero, we can only observe a division by q.

The valuation $v_q(a)$ of a nonzero Laurent series $a \in \bar{C}((q))$ is the smallest $n \in \mathbb{Z}$ such that the coefficient $[q^n]a$ of q^n in a is nonzero. We further define $v_q(0) = +\infty$. For a nonzero solution $f : \alpha + \mathbb{Z} \to \bar{C}((q))$ of an operator $L \in C[x][S]$, we will be interested in how the valuation changes as z ranges through $\alpha + \mathbb{Z}$. As we have noticed, there can be occasional divisions by q as we extend f towards the left or the right, so $v_q(f(z))$ can go up and down as z moves through $\alpha + \mathbb{Z}$. In fact, it can go up and down arbitrarily often, as the solution $f : \mathbb{Z} \to \bar{C}((q))$, $f(z) = 1 + q + (-1)^z$ of the operator $L = S^2 - 1$ shows. However, only when z is a root of ℓ_0 we can have

$$v_q(f(z)) < \min\{v_q(f(z + 1)), \dots, v_q(f(z + r))\},$$

and only when z is a root of $\ell_r(x - r)$ we can have

$$v_q(f(z)) < \min\{v_q(f(z - 1)), \dots, v_q(f(z - r))\}.$$

Since the nonzero polynomials ℓ_0, ℓ_r have at most finitely many roots in $\alpha + \mathbb{Z}$, we can conclude that both

$$\liminf_{n \to -\infty} v_q(f(\alpha + n)) \quad \text{and} \quad \liminf_{n \to +\infty} v_q(f(\alpha + n))$$

are well-defined for every solution $f : \alpha + \mathbb{Z} \to \bar{C}((q))$ of L. Their difference

$$\operatorname{vg} f := \liminf_{n\to+\infty} v_q(f(\alpha+n)) - \liminf_{n\to-\infty} v_q(f(\alpha+n))$$

is called the *valuation growth* of f.

5.2 A Valuation Function

In our context, solutions with negative valuation growth are troublesome, because we want to define the valuation of a residue class $B \in \bar{C}(x)[S]/\langle L\rangle$ at z in terms of the valuations of the sequence terms $(B \cdot b)(z) \in \bar{C}((q))$, where b runs through $\mathrm{Sol}(L)$. When $b \in \mathrm{Sol}(L)$ has negative valuation growth, then we can have $v_q((B \cdot b)(z)) < 0$ for infinitely many z, which makes it hard to meet assumption (E). Moreover, if all solutions have positive valuation growth, we have $v_q((B \cdot b)(z)) > 0$ for infinitely many z, which is also in conflict with assumption (E). In order to circumvent this problem, we let $Z \subseteq \bar{C}$ be such that for each orbit $\alpha + \mathbb{Z}$ with $Z \cap (\alpha + \mathbb{Z}) \neq \emptyset$ and for which L has a solution in $\bar{C}((q))^{\alpha+\mathbb{Z}}$ with nonzero valuation growth, the set $Z \cap (\alpha + \mathbb{Z})$ has a (computable) right-most element. We then define the value function $\mathrm{val}_z : V \to \mathbb{Z} \cup \{\infty\}$ by

$$\mathrm{val}_z(B) := \min_{b\in\mathrm{Sol}(L)} \Big(v_q((B \cdot b)(z)) - \liminf_{n\to\infty} v_q(b(z-n)) \Big).$$

We use the convention $\infty - \infty = \infty$.

Proposition 18. *val_z is a value function for every $z \in Z$.*

Proof. We check the conditions of Def. 2.

(*i*) If $B = 0$, then $B \cdot b$ is the zero sequence for every $b \in \mathrm{Sol}(L)$, so $v_q((B \cdot b)(z)) = \infty$ for all $n \in \mathbb{Z}$.
Conversely, let $B \in \bar{C}(x)[S]$ be such that $\mathrm{val}_z([B]) = \infty$. We may assume that the order of B is less than r, so that $[B] = 0$ is equivalent to $B = 0$. By $\mathrm{val}_z([B]) = \infty$ we have $v_q((B \cdot b)(z)) = \infty$ for all $b \in \mathrm{Sol}(L)$, i.e., $(B \cdot b)(z) = 0$ for all $b \in \mathrm{Sol}(L)$.
If $b_1, \dots, b_r$ is a basis of $\mathrm{Sol}(L)$, then the matrix

$$M = ((b_j(z+i-1)))_{i,j=1}^{r} \in \bar{C}((q))^{r\times r}$$

is regular. Now if B were nonzero and $\beta_k S^k$ is a nonzero term appearing in B, then multiplying the kth row of M by β_k and adding suitable multiples of other rows to the kth row, we obtain a matrix whose kth row is 0, because $(B \cdot b_1)(z) = \cdots = (B \cdot b_r)(z) = 0$. On the other hand, the determinant of this matrix is equal to $\beta_k \det(M) \neq 0$, so B cannot be nonzero.

(*ii*) Clear by $v_q((uf)(z)) = v_q(u) + v_q(f(z))$ for all $u \in \bar{C}((q))$ and $f \in \bar{C}((q))^{z+\mathbb{Z}}$.

(*iii*) Clear by $v_q(((B_1+B_2)\cdot u)(z)) = v_q((B_1\cdot u)(z)+(B_2\cdot u)(z)) \geq \min(v_q((B_1\cdot u)(z)), v_q((B_2\cdot u)(z)))$ for all $u \in \bar{C}((q))^{z+\mathbb{Z}}$. ∎

Next, we show that we can meet the computability assumptions of Section 3. Note again that (B), (C), (D) are subsumed in (B′), (C′), (D′), respectively.

(A) It is assumed that C is a computable field, so it is clear that arithmetic in $\bar{C}(x)$ and V are computable, and that v_z is computable. We show that val_z is computable as well.

Let $\zeta \in z + \mathbb{Z}$ be such that all roots of $\ell_0\ell_r$ contained in $z+\mathbb{Z}$ are to the right of ζ, and consider the basis $b_1, \dots, b_r$ of $\mathrm{Sol}(L)$ in $\bar{C}((q))^{z+\mathbb{Z}}$ defined by the initial values $b_j(\zeta+i-1) = \delta_{i,j}$ $(i, j = 1, \dots, r)$. We shall prove that for all $\eta \in z + \mathbb{Z}$,

$$\mathrm{val}_\eta(B) = \min_{j=1}^{r} v_q((B \cdot b_j)(\eta)).$$

Since we can compute $(B \cdot b_j)(\eta)$ for any $j = 1, \dots, r$ and $\eta \in z + \mathbb{Z}$, this implies that val_η is computable. In particular, val_z is then computable.
We have $\min_{i=1}^{r} v_q(b_j(\zeta + i - 1)) = 0$ for $j = 1, \dots, r$ by construction, and in fact $\liminf_{n\to+\infty} v_q(b_j(\zeta - n)) = 0$ for $j = 1, \dots, r$, because at no position $\zeta - n$ the valuation can be smaller than the minimum valuation of its r neighbors to the right or than the minimum valuation of its r neighbors to the left, due to the lack of roots of $\ell_0\ell_r$ in the range under consideration.
Let now $b = c_1b_1 + \cdots + c_rb_r$ for coefficients $c_1, \dots, c_r \in \bar{C}((q))$. Let $v := \min_{j=1}^{r} v_q(c_j)$. Assume that $v = 0$, and let j_0 be such that $v_q(c_{j_0}) = 0$. Then for all $\eta \in z + \mathbb{Z}$,

$$v_q(b(\eta)) \geq \min_{j=1}^{r} v_q(b_j(\eta))$$

and $v_q((B \cdot b)(\eta)) \geq \min_{j=1}^{r} v_q((B \cdot b_j)(\eta))$.
Furthermore, by construction of the basis of b_j's, for all $i \in \{1, \dots, r\}$, $b(\zeta + i - 1) = c_i$, so $\min_{i=1}^{r} v_q(b(\zeta + i - 1)) = 0$. Again, for lack of roots of $\ell_0\ell_r$ left of ζ, it implies that

$$\liminf_{n\to+\infty} v_q(b(\zeta - n)) = 0.$$

It follows from the above that
$v_q((B \cdot b)(\eta)) - \liminf_{n\to+\infty} v_q(b(\eta - n)) \geq \min_{j=1}^{r} v_q((B \cdot b_j)(\eta))$.
Assume now that $v \neq 0$. In that case, consider $q^{-v}b = q^{-v}c_1b_1 + \cdots + q^{-v}c_rb_r$, with $\min_{j=1}^{r} v_q(q^{-v}c_j) = 0$. From the above,

$$\begin{aligned} &v_q((B \cdot q^{-v}b)(\eta)) - \liminf_{n\to+\infty} v_q(q^{-v}b(\eta - n)) \\ &\geq \min_{j=1}^{r} v_q((B \cdot b_j)(\eta)). \end{aligned}$$

Since for all $\eta \in z + \mathbb{Z}$ we have $v_q(q^{-v}b(\eta)) = v_q(b(\eta)) - v$ and
$v_q((B \cdot q^{-v}b)(\eta)) = v_q((q^{-v}B \cdot b)(\eta)) = v_q((B \cdot b)(\eta)) - v$, it still holds that
$v_q((B \cdot b)(\eta)) - \liminf_{n\to+\infty} v_q(b(\eta - n)) \geq \min_{j=1}^{r} v_q((B \cdot b_j)(\eta))$, so that indeed $\mathrm{val}_\eta(B) = \min_{j=1}^{r} v_q((B \cdot b_j)(\eta))$.

(B′) We can take $x_z = x - z$.

(C′) Let $B_1, \dots, B_d \in \bar{C}(x)[S]/\langle L\rangle$ be given. We can then compute $v := \min_{i=1}^{d} \mathrm{val}_z(B_i)$ and we can find the required $\alpha_1, \dots, \alpha_{d-1} \in \bar{C}$ by equating the coefficients of q^n for $n \leq v$ in the linear combination $\alpha_1(B_1 \cdot b_j)(z) + \cdots + \alpha_{d-1}(B_{d-1} \cdot b_j)(z) + (B_d \cdot b_j)(z)$ to zero and solving the resulting inhomogeneous linear system for $\alpha_1, \dots, \alpha_{d-1}$.

(D′) Clear.

(E) First we shall prove that if $\alpha + \mathbb{Z}$ does not contain a root of $\ell_0\ell_r$, then $\mathcal{B} = \{1, S, \dots, S^{r-1}\}$ is an integral basis for all $z \in Z \cap \alpha + \mathbb{Z}$. For such z, consider the basis $b_1, \dots, b_r$ of

$\mathrm{Sol}(L) \subseteq \bar{C}((q))^{\alpha+\mathbb{Z}}$ with $b_j(z+i-1) = \delta_{i,j}$ $(i,j = 1,\dots,r)$. By the discussion of (A), for any operator $A \in V$, we have

$$\mathrm{val}_z(A) = \min_{j=1}^{r} v_q((A \cdot b_j)(z)).$$

Let $A = p_0 + \cdots + p_{r-1}S^{r-1}$ be an operator in $V = \bar{C}(x)[S]/\langle L\rangle$. By the construction of the basis b_j's, for all $j = \{1,\dots,r\}$, $(A \cdot b_j)(z) = p_{j-1}(x+q-z)$. It imples that

$$\min_{j=1}^{r} v_q((A \cdot b_j)(z)) = \min_{j=0}^{r-1} v_z(p_j).$$

So A is integral if and only if $v_z(p_j) \geq 0$ for all j and $\mathcal{B}$ is an integral basis at z. Since $\ell_0\ell_r$ can only have at finitely many roots, we can restrict Z_0 to finitely many orbits $\alpha + \mathbb{Z}$. In each of these orbits, there is a natural bound for Z_0 to the left after lack of roots of $\ell_0\ell_r$ by the similar argument as above. If L has a solution with nonzero valuation growth, then the bound to the right is given by the choice of Z. Now suppose all solutions of L in $\bar{C}((q))^{\alpha+\mathbb{Z}}$ have zero valuation growth. Let $\zeta \in \alpha + Z$ be such that all roots of $\ell_0\ell_r$ are contained to the left. For each $z = \zeta + n$ with $n \geq 0$, choosing the basis $b_j(z+i-1) = \delta_{i,j}(i,j = 1,\dots,r)$, we get

$$\liminf_{n\to+\infty} v_q(b_j(z+n)) = \min_{i=1}^{r} v_q(b_j(z+i-1)) = 0$$

for all $j = 1,\dots,r$. Then $\liminf_{n\to+\infty} v_q(b_j(z-n)) = 0$. For any operator $A \in V$, it again follows that $\mathrm{val}_z(A) = \min_{j=1}^{r} v_q((A \cdot b_j)(z))$ and hence $\mathcal{B}$ is an integral basis at such a point z for the same reason.

(F) We can take any basis of $V = \bar{C}(x)[S]/\langle L\rangle$ whose basis elements belong to $C(x)[S]/\langle L\rangle$, for example $1, S, \dots, S^{r-1}$. If $z, \tilde{z} \in \bar{C}$ are conjugates, let σ be an element of the Galois group of $\bar{C}$ over C that maps z to $\tilde{z}$. Then for every solution $f \in \bar{C}((q))^{z+\mathbb{Z}}$ of L also $\sigma(f) \in \bar{C}((q))^{\tilde{z}+\mathbb{Z}}$ is a solution of L, because L has coefficients in C, so $\sigma(L) = L$.
Since we have

$$\begin{aligned}&\sigma((\alpha_0 + \cdots + \alpha_{r-1}S^{r-1})(f))\\ &= (\sigma(\alpha_0) + \cdots + \sigma(\alpha_{r-1})S^{r-1})(\sigma(f))\end{aligned}$$

for any $\alpha_0,\dots,\alpha_{r-1} \in \bar{C}(x)$, it follows that $\mathrm{val}_z(\alpha_0 + \cdots + \alpha_{r-1}S^{r-1}) \geq \mathrm{val}_{\tilde{z}}(\sigma(\alpha_0) + \cdots + \sigma(\alpha_{r-1})S^{r-1})$. Equality follows by exchanging z and $\tilde{z}$.

We now define the discriminant function in the shift setting. For each $\alpha \in Z$, by the item (A), we can choose a basis $b_1,\dots,b_r$ of $\mathrm{Sol}(L)$ such that $\mathrm{val}_\alpha(B) = \min_{j=1}^{r} v_q((B \cdot b_j)(\alpha))$. For any k-vector space basis $B = \{B_1,\dots,B_r\}$ of $V = \bar{C}(x)[S]/\langle L\rangle$, we can take

$$\mathrm{Disc}_\alpha(B) := v_q(\det(((B_i \cdot b_j)(\alpha))_{i,j=1}^{r})) \in \mathbb{Z}.$$

It is well-defined since the matrix $((B_i \cdot b_j)(\alpha)) = (p_{i,\ell}) \cdot (b_j(\alpha+\ell-1))$ is regular, where $B_i = \sum_{j=1}^{r} p_{i,\ell}S^{\ell-1}$ with $p_{i,\ell} \in \bar{C}(x)$. If B_i's are integral for α, then $v_q((B_i \cdot b_j)(\alpha)) \geq 0$ for all $i,j = 1,\dots,r$. It follows that $\mathrm{Disc}_\alpha(B) \geq 0$.

Let $\alpha_1,\dots,\alpha_{d-1} \in k$, replacing B_d by $\alpha_1B_1 + \cdots + \alpha_{d-1}B_{d-1} + B_d$ (resp. by $(x-\alpha)^{-1}B_d$) is equivalent to multiplying the matrix $((B_i \cdot b_j)(\alpha))$ by a matrix with determinant 1 (resp. with determinant $(x-\alpha)^{-1}$) and it follows that the valuation of the determinant is constant (resp. is strictly decreasing).

Example 19. *Let $L = (x+2)^2 + xS^2 + (x+2)S^3$. For every $\alpha \notin \mathbb{Z}$, we have that $\{1, S, S^2\}$ is a local integral basis for $V = C(x)[S]/\langle L\rangle$ at $\alpha + \mathbb{Z}$. For the orbit $\mathbb{Z}$, choosing $b_j(-2+i-1) = \delta_{i,j}$ for $i,j = 1,2,3$, we obtain a basis of the solution space in $C((q))^{\mathbb{Z}}$:*

n	$\cdots$	-2	-1	0	1	2	$\cdots$
$b_1(n)$	$\cdots$	1	0	0	$-q$	$\frac{q(q-1)}{q+1}$	$\cdots$
$b_2(n)$	$\cdots$	0	1	0	0	$-q-1$	$\cdots$
$b_3(n)$	$\cdots$	0	0	1	$\frac{-q+2}{q}$	$\frac{q^2-3q+2}{q(q+1)}$	$\cdots$

Then $\mathrm{val}_\alpha(B) = \min_{j=1}^{3} v_q((B \cdot b_j)(\alpha))$ for any operator $B \in V$ and $\alpha \in \mathbb{Z}$. Since the solution b_3 has negative valuation growth, for a global integral basis the set Z has to be bounded on the right in the orbit $\mathbb{Z}$. Take $Z = C \setminus \{1,2,\dots\}$. At $\alpha = 0$, we have 1 is locally integral, but S, S^2 are not since $\mathrm{val}_0(S) = \mathrm{val}_0(S^2) = -1$. However, xS, xS^2 are locally integral. By Alg. 10, we can find a local integral basis at 0:

$$\left\{1, \tfrac{x-2}{x^2} + \tfrac{1}{x}S, \tfrac{-2}{x} + S^2\right\}.$$

Using such a basis as an input, continue to find all locally integral elements at $\alpha = -1$. Similarly replace $B_3 = \frac{-2}{x} + S^2$ by $(x+1)B_3$ since $\mathrm{val}_1(B_3) = -1$. This operation does change the local integrality at $Z \setminus \{-1\}$, because $x+1$ is invertible in the localization of $C[x]$ at any $z \neq -1$. So the output local integral basis at $\alpha = -1$ is also a global integral basis for Z:

$$\left\{1, \tfrac{x-2}{x^2} + \tfrac{1}{x}S, \tfrac{-x+2}{x^2} + \tfrac{-3x-1}{x(x+1)^2}S + \tfrac{1}{x+1}S^2\right\}.$$

Acknowledgement. We thank the referees for their careful reading and their valuable suggestions, in particular the referee who pointed out the implication $(a) \Rightarrow (b)$ in Theorem 13.

REFERENCES

[1] Sergei A. Abramov, Moulay A. Barkatou, Mark van Hoeij, and Marko Petkovsek. Subanalytic solutions of linear difference equations and multidimensional hypergeometric sequences. *J. Symb. Comput.*, 46(11):1205–1228, 2011.
[2] M. Bronstein. The lazy Hermite reduction. Technical Report 3562, INRIA, 1998.
[3] Manuel Bronstein. Symbolic integration tutorial. ISSAC'98, 1998.
[4] Shaoshi Chen, Manuel Kauers, and C. Koutschan. Reduction-based creative telescoping for algebraic functions. In *Proc. ISSAC'16*, pages 175–182, 2016.
[5] Shaoshi Chen, Mark van Hoeij, Manuel Kauers, and Christoph Koutschan. Reduction-based creative telescoping for fuchsian D-finite functions. *J. Symb. Comput.*, 85:108–127, 2018.
[6] Claude Chevalley. Introduction to the Theory of Algebraic Functions of One Variable. AMS Mathematical Surveys, 1951.
[7] Thomas Cluzeau and Mark van Hoeij. Computing hypergeometric solutions of linear recurrence equations. *AAECC*, 17:83–115, 2006.
[8] Dominique Duval. Rational Puiseux expansions. *Compositio Mathematica*, 70(2):119–154, 1989.
[9] Antonio J. Engler and Alexander Prestel. *Valued Fields.* Springer, 2005.
[10] László Fuchs. Vector spaces with valuations. *J. of Algebra*, 35(1-3):23–38, 1975.
[11] Erdal Imamoglu. *Algorithms for Solving Linear Differential Equations with Rational Function Coefficients.* PhD thesis, Florida State University, 2017.
[12] Erdal Imamoglu and Mark van Hoeij. Computing hypergeometric solutions of second order linear differential equations using quotients of formal solutions and integral bases. *J. Symb. Comput.*, 83:254–271, 2007.
[13] Edward L. Ince. *Ordinary Differential Equations.* Dover, 1926.
[14] Manuel Kauers and Christoph Koutschan. Integral D-finite functions. In *Proc. ISSAC'15*, pages 251–258, 2015.
[15] Jean-Pierre Serre. Local fields. *Graduate Texts in Mathematics*, 1979.
[16] Jean Pierre Tignol and Adrian R. Wadsworth. *Value Functions on Simple Algebras, and Associated Graded Rings.* Springer, 2015.
[17] Barry M. Trager. *Integration of Algebraic Functions.* PhD thesis, MIT, 1984.
[18] Mark van Hoeij. An algorithm for computing an integral basis in an algebraic function field. *Journal of Symbolic Computation*, 18(4):353–363, 1994.
[19] Mark van Hoeij. Finite singularities and hypergeometric solutions of linear recurrence equations. *Journal of Pure and Applied Algebra*, 139:109–131, 1999.
[20] G. Zeng. Valuations on a module. *Communic. Algebra*, 35(8):2341–2356, 2007.

A Gröbner-Basis Theory for Divide-and-Conquer Recurrences

Frédéric Chyzak
frederic.chyzak@inria.fr
INRIA

Philippe Dumas
philippe.dumas@inria.fr
INRIA

ABSTRACT

We introduce a variety of noncommutative polynomials that represent divide-and-conquer recurrence systems. Our setting involves at the same time variables that behave like words in purely noncommutative algebras and variables governed by commutation rules like in skew polynomial rings. We then develop a Gröbner-basis theory for left ideals of such polynomials. Strikingly, the nature of commutations generally prevents the leading monomial of a polynomial product to be the product of the leading monomials. To overcome the difficulty, we consider a specific monomial ordering, together with a restriction to monic divisors in intermediate steps. After obtaining an analogue of Buchberger's algorithm, we develop a variant of the F_4 algorithm, whose speed we compare.

CCS CONCEPTS

• **Computing methodologies → Algebraic algorithms**.

KEYWORDS

Gröbner bases, divide-and-conquer recurrences, skew polynomials

ACM Reference Format:
Frédéric Chyzak and Philippe Dumas. 2020. A Gröbner-Basis Theory for Divide-and-Conquer Recurrences. In *International Symposium on Symbolic and Algebraic Computation (ISSAC '20), July 20–23, 2020, Kalamata, Greece.* ACM, New York, NY, USA, 8 pages. https://doi.org/10.1145/3373207.3404055

Divide-and-conquer recurrences appear at the interface between mathematics and (theoretical) computer science, namely, in relation to number systems, formal languages, number theory, and complexity theory. For example, the total number u_n of operations $(+, -, \times)$ in Karatsuba's algorithm when multiplying polynomials of degree less than n satisfies $u_0 = 0$, $u_1 = 1$, and the system of recurrences

$$u_{2n+2} - 3u_{n+1} = 8n+4, \quad u_{2n+3} - 2u_{n+2} - u_{n+1} = 8n+8 \quad \text{for } n \geq 0.$$

So far, the literature has focused almost exclusively on finding the asymptotic behavior of some sequence defined by first-order recurrences; see the references in [5, 9]. In the example above, the sequence undoubtedly exists and is defined uniquely. But can we guarantee that any given divide-and-conquer system actually defines a sequence, and this uniquely? This motivates an algorithmic study of suitable left ideals that encode divide-and-conquer systems.

In Section 1, we explain how divide-and-conquer recurrences can be expressed as polynomials of a noncommutative algebra. In Section 2, we develop a Gröbner-basis theory in it, by using a specific monomial ordering that we call *breadth-first ordering*. This leads to a Buchberger algorithm whose correctness we prove by an analogue of the usual criterion on S-polynomials. We then replace the pair-completion approach by a linear-algebraic one in Section 3, and we develop a variant of the algorithm F_4. Timings are briefly presented in Section 4, together with a speed comparison.

We close this introduction with a short review of related works on Gröbner bases, which we hope the reader will keep in mind and contrast to our contribution. The Gröbner-basis theory for commutative polynomial algebras $k[x_1, \ldots, x_n]$ over a given field k is well understood, see textbooks like [2, 4]. The theory has since long been studied in relation to linear algebra [13]. This led to developments like Faugère's algorithm F_4 [6], a big algorithmic speed-up. Extensions to noncommutative contexts range between two extremes. A first line of research is towards free noncommutative algebras $k\langle a_1, \ldots, a_n\rangle$ [16–18] and path algebras $k\Gamma$ [8, 20], replacing commutative monomials by words on letters a_i or by paths on a graph Γ. Noetherianity is typically lost, but algorithms have been given both for one-sided and two-sided ideals. In these contexts, monomials commute with the coefficients from k and the one-sided case is regarded to be simpler than the two-sided. Another line of research concerns k-algebras given by generators and relations, for well-identified forms of relations. Early works in this direction provided algorithms for Weyl algebras, $k\langle x_1, \ldots, x_n, y_1, \ldots, y_n;\ y_j x_i = x_i y_j + \delta_{i,j},\ y_j y_i = y_i y_j,\ x_j x_i = x_i x_j,\ 1 \leq i, j \leq n\rangle$, where $\delta_{i,j}$ is 1 if $i = j$ and 0 otherwise [7], and for enveloping algebras of Lie algebras, that is, given a finite-dimensional Lie algebra $\mathfrak{g}$ with k-basis $(a_1, \ldots, a_n)$, the associative algebra $k\langle x_1, \ldots, x_n;\ x_j x_i = x_i x_j + [a_j, a_i],\ 1 \leq i, j \leq n\rangle$ [1]. These studies focus on one-sided ideals, which is natural for [7] as Weyl algebras have no nontrivial two-sided ideals. They were extended to noncommutative polynomial rings of solvable type $k\langle x_1, \ldots, x_n;\ x_j x_i = c_{i,j} x_i x_j + p_{i,j},\ 1 \leq i, j \leq n\rangle$, where the $c_{i,j}$ are nonzero and the $p_{i,j}$ are polynomials smaller than $x_i x_j$ in a suitable sense [10, 14]. In all such algebras given by generators and relations, again, the monomials commute with the coefficients from k. In contrast, the rings of difference-differential operators over rational-function coefficients [19] can be obtained by tensoring Weyl algebras or similar algebras with the field $F = k(x_1, \ldots, x_n)$: they involve variables y_i that commute with one another but generally not with the coefficients from the field F. A similar situation occurs with Ore algebras [3], which are generalizations to more types of linear functional operators. A generalization of [10] to a sort of solvable polynomials rings whose monomials in the x_i need not commute with the coefficients from k was developed in [11]. All these cases are (left, right, two-sided) Noetherian rings. For an integer $b \geq 2$, the algebra $A := k\langle x, y; yx = x^b y\rangle$ of linear b-Mahler operators with polynomial coefficients directly relates to the algebras of section operators discussed in the present article; see our Conclusion. Its analogue with rational-function coefficients, $k(x) \otimes_{k[x]} A$, is a case of Ore algebras, while the algebra A itself is non-Noetherian. The theory was adapted to A so as to provide

ISSAC '20, July 20–23, 2020, Kalamata, Greece

ACM ISBN 978-1-4503-7100-1/20/07…$15.00
https://doi.org/10.1145/3373207.3404055

computable Gröbner bases for finitely generated one-sided and two-sided ideals [21]. The setting to be introduced in Section 1 inherits from both the free noncommutative algebras $k\langle a_1, \ldots, a_n\rangle$ and Ore algebras by considering skew polynomials whose monomials are words that have commutation rules with their coefficients in a field $k(x)$. It is non-Noetherian. A similar situation, but distinct in that the monomials are not just noncommutative words but satisfy commutation rules as well, was introduced in an application to the calculation of symmetries of discrete systems [15]; see the generalization [12].

Acknowledgement. Supported in part by ANR-19-CE40-0018.

1 SKEW POLYNOMIALS

In this work, k is a commutative, computable field. The sequences we have in mind are defined on the set of nonnegative integers $\mathbb{Z}_{\geq 0}$. We also see them as those sequences defined on $\mathbb{Z}$ that have their supports in $\mathbb{Z}_{\geq 0}$. To a sequence $(u_n)_{n\in\mathbb{Z}_{\geq 0}}$, we associate the formal power series $\sum_{n\geq 0} u_n x^n$ in $k[[x]]$. The ring $k[[x]]$ is a subring of the field of formal Laurent series $k((x))$, which proves to be the right algebraic set to think of our sequences.

1.1 Section operators

To formalize the study of divide-and-conquer recurrences we introduce what we call *section operators*. We fix an integer $b \geq 2$, which the reader can think of as the radix of a numeration system. For each integer $0 \leq r < b$, we consider, with the same notation, the operators $S_{b,r}$ that act k-linearly on sequences in $k^{\mathbb{Z}}$, and, respectively, on formal Laurent series in $k((x))$ by

$$S_{b,r} \cdot u_n = u_{bn+r}, \qquad S_{b,r} \cdot \sum_n u_n x^n = \sum_n u_{bn+r} x^n, \tag{1}$$

where, in each case, n ranges in $\mathbb{Z}$.

The operators $S_{b,r}$ given by $0 \leq r < b$ generate a monoid of endomorphisms, which, by extension of the notation, are the $S_{b^\ell,r}$ obtained for all integers $\ell \geq 1$ and $0 \leq r < b^\ell$, and are related by the composition rule

$$S_{b^\ell,r} S_{b^{\ell'},r'} = S_{b^{\ell+\ell'},b^{\ell'}r+r'}. \tag{2}$$

Moreover, for any $\ell \geq 1$ and for each $0 \leq r < b^\ell$, there is a 'Leibniz' formula: for any two formal Laurent series $f(x)$ and $g(x)$,

$$\begin{aligned} S_{b^\ell,r} \cdot (f(x)g(x)) = &\sum_{s=0}^{r} \left(S_{b^\ell,r-s} \cdot f(x)\right) \left(S_{b^\ell,s} \cdot g(x)\right) \\ &+ x \sum_{s=r+1}^{b^\ell-1} \left(S_{b^\ell,r-s+b^\ell} \cdot f(x)\right) \left(S_{b^\ell,s} \cdot g(x)\right). \end{aligned} \tag{3}$$

1.2 Skew polynomials

In order to give a polynomial version of the section operators, we introduce the associative $k(x)$-algebra $k(x)\langle T\rangle$ generated by indeterminates $T_{b,r}$ with $0 \leq r < b$, subject to the product rule

$$T_{b,r} \times f(x) = \sum_{s=0}^{r} \left(S_{b,r-s} \cdot f(x)\right) T_{b,s} + x \sum_{s=r+1}^{b-1} \left(S_{b,r-s+b} \cdot f(x)\right) T_{b,s} \tag{4}$$

for all $f(x) \in k(x)$, which reflects (3) when $\ell = 1$. We refer to the elements of $k(x)\langle T\rangle$ as *skew polynomials*.

As this rule can be used to rewrite its left-hand side into its right-hand side, polynomials from the $k(x)$-algebra can be viewed as having monomials that are noncommutative words in the $T_{b,r}$ and coefficient from $k(x)$, written on the left of monomials.

We can view elements $f(x)$ from $k(x)$ as operators on $k((x))$, by considering their action by multiplication, and each $T_{b,r}$ as an operator on $k((x))$ by endowing it with the action of the section operator $S_{b,r}$. Then, the Leibniz formula (3) provides an expression for $T_{b,r} \cdot f(x) \cdot g(x) = S_{b,r} \cdot (f(x)g(x))$, which, by (4), matches the result $(T_{b,r}f(x)) \cdot g(x)$ of the action of the operator $T_{b,r}f(x)$ on $g(x)$. One checks that this defines a left action of $k(x)\langle T\rangle$ on $k((x))$.

1.3 Exponent notation

In the classical commutative case, computations on ideals use monomial orderings and very basic results about the exponents, which are elements of $\mathbb{Z}^m_{\geq 0}$. This leads us to introduce a parallel notation for the monomials. As exponents, we use words over the alphabet $\mathcal{A}$ of the numeration system with radix b, that is, the alphabet $\mathcal{A} = \{0, 1, \ldots, b-1\}$. In other words, we have two notations $T^r = T_{b,r}$ with $0 \leq r < b$ for the generators of $k(x)\langle T\rangle$.

To make an explicit link with section operators, for any word $w \in \mathcal{A}^*$ introduce the integer r whose b-ary expansion is w:

$$r = (w)_b = w_{\ell-1} b^{\ell-1} + \cdots + w_0 b^0.$$

The monomial T^w acts on $k((x))$ as does $S_{b^\ell,r}$, which results from applying Equation (2) iteratively, since the action of a section operator $S_{b^\ell,r}$ on $k((x))$ is the same as the action of T^w, obtained as the composition of the action of $T^{w_{\ell-1}}$ after the action of $w_{\ell-2}, \ldots$, after the action of T^{w_0}.

Hence we can extend the notation $T_{b,r}$ with $0 \leq r < b$ into $T_{b^\ell,r}$ with $0 \leq r < b^\ell$ by the relation

$$T^w = T_{b^\ell,r}, \quad \text{with } \ell = |w|, r = (w)_b. \tag{5}$$

Upon setting $\ell := |w|$, $\ell' := |w'|$, $r := (w)_b$, and $r' := (w')_b$, Equation (2) thus provides the simple formula $T^w T^{w'} = T^{ww'}$, where the word ww' is the concatenation of w and w'. As a consequence, the monoid of words and the monoid of monomials are clearly isomorphic. Furthermore, Formula (4) generalizes by changing b to b^ℓ in the formula, thus mimicking (3) for general ℓ. Written more loosely and after reindexing by words, the formula becomes

$$T^w \times f(x) = \sum_{|w'|=|w|} g_{w'}(x) T^{w'}, \tag{6}$$

for suitable rational functions $g_{w'}(x)$.

For example, with $b = 2$, both formulas $T^{01}T^{101} = T^{01101}$ and $T_{4,1}T_{8,5} = T_{32,13}$ mean the same; after applying to a sequence u, they become $u_{8(4n+1)+5} = u_{32n+13}$.

The length of words plays a role akin to the degree in the commutative case. This leads us to define the degree in $k(x)\langle T\rangle$ by $\deg 0 = -\infty$ and for a nonzero polynomial by the formula

$$\deg \sum_w c_w T^w = \max\{|w| \mid c_w \neq 0\}. \tag{7}$$

It satisfies the usual property of a degree with respect to the multiplication and the addition.

In computational commutative algebra, it is usual to support a piece of reasoning by drawing so-called stairs. A polynomial is seen through its carrier, which is the set of its exponents. Similarly, we

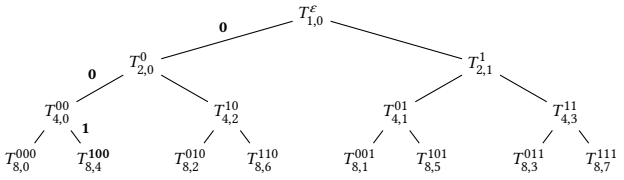


Figure 1: The tree of monomials for $b = 2$. Each node $T^{w}_{b^{\ell},r}$ denotes a monomial in two notations: $T^{w} = T_{b^{\ell},r}$. The path from the root to a monomial follows the word w, read from right to left, that is, from the least significant digits first.

view a polynomial in $k(x)\langle T\rangle$ via the set of its exponents, which are words in $\mathcal{A}^*$. Owing to noncommutativity, these are the nodes of a b-ary tree, instead of the nodes of the square lattice (Fig. 1).

2 GRÖBNER BASES

In this section, we develop a theory for ideals of section operators, adapting what can be of the classical commutative theory [2, 4].

2.1 Monomial ordering

As opposed to the ordinary theories of Gröbner bases, our new theory makes use of a single monomial ordering, which is motivated by two constraints.

First, in our applications to divide-and-conquer recurrences, we do not want to produce recurrence formulas like $u_{2n+1} = u_{8n+3} + u_{4n} + u_n$, where the term $u_{b^{\ell}+r}$ on the left-hand side is defined by using some $u_{b^{\ell'}+r'}$ where ℓ' is larger than ℓ. Hence, we need an ordering that refines the degree.

Our second constraint is technical: Property 3 in Proposition 1 below will prove to be crucial to make our theory possible, in particular by the proof of Lemma 3. In practice, this leads to the choice of a single monomial ordering used in what follows.

Ordering monomials T^{w} is equivalent to ordering words w. In the case $b = 2$, our ordering lists the words involved as superscripts in Figure 1 in the order they appear when read by a breadth-first (left to right) traversal of the tree. We call it the *breadth-first ordering*: $\varepsilon < 0 < 1 < 00 < 10 < 01 < 11 < 000 < 100 < 010 < 110 < 001 < 101 < 011 < 111 < \cdots$. It can be defined formally as follows. First, order the alphabet $\mathcal{A}$ according to $0 < 1 < \cdots < b-1$. Next, words over $\mathcal{A}$ are first ranked by length, with ties broken by the lexicographical ordering on words read from right to left. (With our convention for defining $(w)_b$, this means from the least significant digit to the most significant one.) In other words, we define $w < w'$ if $|w| < |w'|$, or else if $|w| = |w'|$ and the two words can be written $w = ujv$ and $w' = u'j'v$ for words u, u', and v, and letters $j < j'$.

Proposition 1. *The breadth-first ordering on the monoid of monomials satisfies the following properties:*

1. *it is total and refines the degree,*
2. *every set of monomials has a smallest element,*
3. *it is* left compatible *with concatenation, that is if $|v| = |v'|$ and $T^{v} < T^{v'}$, then $T^{uv} < T^{u'v'}$ whenever $|u| = |u'|$.*

Proof. The first and third assertions are direct consequences of the definition of the order on words. The second assertion follows from the first and the fact that there exist only finitely many monomials of a given degree. □

2.2 Leading monomials

With a total ordering on monomials at our disposal, we can consider leading monomials and leading coefficients.

Definition 2. *The leading monomial* $\mathrm{lm}(F)$ *of a nonzero skew polynomial F is the largest monomial in F with respect to breadth-first ordering. The leading coefficient* $\mathrm{lc}(F)$ *is the coefficient of the leading monomial* $\mathrm{lm}(F)$.

A key point in the commutative case is the fact that the leading monomial of a product is the product of the leading monomials. Formula (4) induces a breach to this law, which seems to preclude the translation of the commutative case into our noncommutative case. Indeed, when we multiply a term $c(x)T^{v}$ by a monomial T^{u} on the left, we generally obtain all the monomials $T^{u'v}$ with $|u'| = |u|$ and not only the monomial T^{uv}.

Lemma 3. *The leading monomial w.r.t. breadth-first ordering of the product of two nonzero skew polynomials is the product of their leading monomials whenever the right-hand factor is monic.*

Proof. Let F and G be the two skew polynomials to be multiplied, with F nonzero and G monic. Without loss of generality, we can also assume that F is monic, as changing F into $1/\mathrm{lc}(F)\times F$ does not modify the leading monomial of the left factor, and, by associativity, of the product FG. Let T^{u} and T^{v} be the leading monomials of F and G, respectively. Without loss of generality, we can neglect the terms with degree smaller than the degree of those leading monomials, since the ordering refines the degree. So we consider

$$F = T^{u} + \sum_{u'<u} c_{u'}(x)T^{u'}, \qquad G = T^{v} + \sum_{v'<v} d_{v'}(x)T^{v'},$$

where u' and v' are words subject to $|u'| = |u|$, $|v'| = |v|$. Apart from the monomial $T^{u}T^{v} = T^{uv}$, which bears coefficient 1, the product FG includes two types of terms: first, terms $e(x)T^{u''v'}$ with $|u''| = |u|$, $v' < v$; second, terms $c_{u'}(x)T^{u'v}$ with $u' < u$. In the first case, the monomial $T^{u''v'}$ is smaller than T^{uv} since the breadth-first ordering is left compatible. In the second case, $T^{u'v}$ is smaller than T^{uv} because u' is smaller than u and, again, by the left compatibility with concatenation. Hence the leading monomial of the product is the product of the leading monomial. □

For convenience, we augment the monoid of monomials with the element 0, and its ordering so that 0 becomes its minimal element. We then extend the map $\mathrm{lm}(\cdot)$ by giving it the value 0 at the polynomial 0, so that $\mathrm{lm}(0) = 0 < \mathrm{lm}(F)$ for any nonzero skew polynomial F. The total order on the augmented monoid of monomials then induces a preorder on skew polynomials: for any F and G in $k(x)\langle T\rangle$, we say that F is smaller than G, denoted $F < G$, if $\mathrm{lm}(F) < \mathrm{lm}(G)$, and that F is smaller than or equivalent to G, denoted $F \le G$, if $\mathrm{lm}(F) \le \mathrm{lm}(G)$. Observe that the inequality $F < G$ is equivalent to any of the three inequalities obtained by replacing F by $\mathrm{lm}(F)$, G by $\mathrm{lm}(G)$, or both. We will use this equivalence freely.

The property of left compatibility for monomials extends to skew polynomials in the form of the next lemma.

Input: A dividend A, a list of nonzero divisors $(B_1, \dots, B_s)$.
Output: A list of quotients $(Q_1, \dots, Q_s)$ and a remainder R.

1. For i from 1 to s, do $B'_i := \mathrm{lc}(B_i)^{-1} \times B_i$.
2. $R := 0$. For i from 1 to s, do $Q'_i := 0$.
3. While $A \neq 0$ do
   a. if there exists i between 1 and s such that $\mathrm{lm}(B'_i)$ divides $\mathrm{lm}(A)$ on the right, then:
      (1) pick such an i,
      (2) $M := \mathrm{lc}(A)\,\mathrm{lm}(A)\,\mathrm{lm}(B'_i)^{-1}$,
      (3) $Q'_i := Q'_i + M$, $A := A - MB'_i$;
   b. otherwise: $R := R + \mathrm{lc}(A)\,\mathrm{lm}(A)$, $A := A - \mathrm{lc}(A)\,\mathrm{lm}(A)$.
4. For i from 1 to s, do $Q_i := Q'_i \times \mathrm{lc}(B_i)^{-1}$.
5. Return the list $(Q_1, \dots, Q_s)$ and R.

Algorithm 1: Right division algorithm in $k(x)\langle T\rangle$.

Lemma 4. *For any skew polynomials H, K_1 and K_2 from $k(x)\langle T\rangle$, if $H \neq 0$ and $K_1 < K_2$, then $HK_1 < HK_2$.*

Proof. As $\mathrm{lm}(K_1) < \mathrm{lm}(K_2)$, the second of these monomials is nonzero. Therefore, the polynomial K_2 is nonzero, and so is HK_2. Writing $q = \mathrm{lc}(K_2)$, we have $H \times q \neq 0$ and $q^{-1}K_1 < q^{-1}K_2$, so it is sufficient to prove the result for monic K_2. If $K_1 = 0$, then $HK_1 = 0 < HK_2$ and the result is proved. There remains the case $K_1 \neq 0$. For any term $h(x)T^u$ of H and any term $k(x)T^v$ of K_1, by Formula (6) there exist coefficients $g_{u'}(x)$ such that

$$h(x)T^u k(x)T^v = \sum_{u'} g_{u'}(x)T^{u'v},$$

with a sum over those u' satisfying $|u'| = |u|$. The left-compatibility of breadth-first ordering and the strict inequality $T^v \leq \mathrm{lm}(K_1) < \mathrm{lm}(K_2)$ imply $T^{u'v} < T^u\,\mathrm{lm}(K_2)$ for each u'. Therefore,

$$h(x)T^u k(x)T^v \leq \max_{u'} T^{u'v} < T^u\,\mathrm{lm}(K_2) \leq \mathrm{lm}(HK_2),$$

where the last inequality results from the monicity of K_2. Taking a maximum over u and v, we get $HK_1 \leq \max \mathrm{lm}(h(x)T^u k(x)T^v) < HK_2$, thus proving the result. □

2.3 Division

The needed restriction of right quotients to monic skew polynomials is first involved in right division. To work around the difficulty, we write a right division $A = QB+R$ in the form $A = (Q\times c)(c^{-1}\times B)+R$ where c is the leading coefficient of the polynomial B. Of course, we next adjust the computation by changing the quotient $Q' = Q \times c$ into $Q = Q' \times c^{-1}$. This leads to Algorithm 1, which is a simple adaptation to our setting of the usual division algorithm.

Proposition 5. *Given a tuple $(B_1, \dots, B_s)$ of nonzero polynomials in $k(x)\langle T\rangle$, every $A \in k(x)\langle T\rangle$ can be written $A = Q_1B_1+\cdots+Q_sB_s+R$ for polynomials $Q_1, \dots, Q_s, R$ satisfying the following conditions:*

- *the monomials in the remainder R are not divisible by any of the leading monomials of the divisors $B_1, \dots, B_s$;*
- *furthermore, each Q_iB_i satisfies $Q_iB_i \leq A$.*

Such a division is provided by Algorithm 1, whatever choices are made to resolve nondeterminism at Step 3a(1).

Proof. The proof is based on Algorithm 1. Let T^v be the leading monomial of the dividend A at any stage of the computation. If no divisor has a leading monomial that divides T^v, then the term with this monomial is moved from A to the remainder, so that the dividend is made smaller. If there is a divisor

$$B' = T^u + \sum_{u'<u} c_{u'}(x)T^{u'}$$

whose leading monomial T^u divides T^v, then $v = wu$ for some word w. Then we subtract T^wB' from A so that its leading monomial $T^v = T^wT^u$ is killed. According to Lemma 4,

$$T^w \sum_{u'<u} c_{u'}(x)T^{u'} < T^wT^u = T^v,$$

so the next dividend, $A - T^wB'$, is smaller than A.

This process thus produces a strictly decreasing sequence of monomials, given by the $\mathrm{lm}(A)$, which by Proposition 1 must have a lowest element. The process therefore terminates. The correction of the algorithm results from a loop invariant: the value of $A + Q'_1B'_1 + \cdots + Q'_sB'_s + R$ at each entry into the loop body of Step 3 is equal to the initial value of A. As the final value of A is zero, this proves the existence of the division. As the proof above does not depend on the choice of i at Step 3a(1), the final assertion holds. □

Example 6 (Natural ordering)**.** Instead of breadth-first ordering, we could have considered the 'natural' ordering $<_{\mathrm{nat}}$. As breadth-first ordering, it refines degree and is based on lexicographic ordering. But it compares b-ary expansions of integers from the most significant digit to the least significant digit, that is from left to right, contrary to breadth-first ordering which reads from right to left. In other words, given any ℓ and $0 \leq r$, $r' < b^\ell$, natural ordering has $T_{b^\ell,r} <_{\mathrm{nat}} T_{b^\ell,r'}$ if and only if $r < r'$.

Lemma 3 about the leading monomial of a product does not hold true with natural ordering. For example, with $b = 2$, $F = T_{4,2}$, $G = T_{2,1} + \frac{x^3}{1-x^4}T_{2,0}$, the product is $FG = T_{8,5} + \frac{x}{1-x}T_{8,6}$, with leading monomial $T_{8,6}$ for natural ordering, while the product of the leading monomials is $T_{4,2}T_{2,1} = T_{8,5}$.

Moreover, with $<_{\mathrm{nat}}$, it is possible that the division algorithm does not end. For $b = 2$, consider the dividend $F = T_{8,6}$ and the divisors $F_1 = T_{2,1} - \frac{x^3}{1-x^4}T_{2,0}$, $F_2 = T_{4,2} - \frac{1}{1-x^2}T_{4,1}$. The successive dividends are $P_{2k} = \frac{x^k}{(1-x)^{2k}}T_{8,6}$, $P_{2k+1} = \frac{x^k}{(1-x)^{2k+1}}T_{8,5}$, $k \geq 0$. The carrier of the P_k alternates between $T_{8,6}$ and $T_{8,5}$. Note that for breadth-first ordering, the division process ends immediately, because $\mathrm{lm}(F_1) = T_{2,1}$ and $\mathrm{lm}(F_2) = T_{4,1}$, none of which divides $T_{8,6}$.

2.4 Gröbner bases

In contrast with the commutative case, neither Hilbert's basis theorem nor Dickson's lemma is available. As a consequence, in the sequel we restrict to finitely generated left ideals by requesting that ideals be presented by an explicit finite set of generators.

Definition 7. *A Gröbner basis of a left ideal $\mathcal{I}$ in $k(x)\langle T\rangle$ is a finite subset $\mathcal{G}$ of $\mathcal{I}$ whose elements are monic and such that for every F in $\mathcal{I}$, the leading monomial $\mathrm{lm}(F)$ is a left multiple of the leading monomial $\mathrm{lm}(G)$ of some polynomial G in $\mathcal{G}$.*

Proposition 8. *Let $\mathcal{G}$ be a Gröbner basis for a left ideal $\mathcal{I}$. For every polynomial F, there is a unique polynomial R such that $F \equiv R \bmod \mathcal{I}$ and no monomial of R is divisible by a monomial in $\mathrm{lm}(\mathcal{G})$. As a*

consequence, R is the remainder of the division by $\mathcal{G}$ regardless of the chosen division strategy.

Proof. Let us assume that we have $F \equiv R_1 \equiv R_2 \bmod \mathcal{I}$ for distinct R_1 and R_2, both satisfying the condition with regard to $\mathrm{lm}(\mathcal{G})$. Then $R_1 - R_2$ is in $\mathcal{I}$ and nonzero. By the definition of a Gröbner basis, the leading monomial $\mathrm{lm}(R_1 - R_2)$ is divisible by a monomial in $\mathrm{lm}(\mathcal{G})$. But this is impossible since none of the monomials of R_1 and R_2 is divisible by a monomial in $\mathrm{lm}(\mathcal{G})$. We have thus shown the uniqueness of R.

In addition, whatever choices resolve nondeterminism in the division process, division provides us with some polynomial satisfying the two properties, and as a consequence of uniqueness, this polynomial is independent of the choices. □

A crucial ingredient in the theory of Gröbner bases in polynomial rings is the notion of S-polynomials: for two nonzero polynomials F and G, one considers the least common multiple of their leading monomials and forms a combination of F and G that kills this monomial. Owing to noncommutativity, least common multiples of monomials do not always exist in $k(x)\langle T\rangle$ and they are very specific when they do. In $k(x)\langle T\rangle$, a monomial T^u indeed divides another monomial T^v on the right, meaning there exists a quotient $Q \in k(x)\langle T\rangle$ satisfying $T^u = QT^v$ if and only if v is a suffix of u, in which case there exists a monomial w satisfying $u = wv$ and $Q = T^w$. Thus, when two monomials T^u and T^v have a least common multiple, this is necessarily one of the two monomials.

As is usual in theories of Gröbner bases where monomials need not have common multiples, like in the theory for polynomial modules, we define the S-polynomial of two monic polynomials P and Q of $k(x)\langle T\rangle$, with respective leading monomials T^u and T^v, to be 0 when neither T^u divides T^v nor T^v divides T^u, and to be $P - T^wQ$ when $T^u = T^wT^v$ for some w, respectively $Q - T^wP$ when $T^v = T^wT^u$ for some w. Observe that we restrict the definition to monic polynomials, as nonmonic divisors are ill-behaved.

We next obtain a characterization of Gröbner bases in $k(x)\langle T\rangle$ akin to that in commutative polynomial rings, via S-polynomials.

Theorem 9. *A family $\mathcal{G} = (G_i)_{1\le i\le m}$ of monic polynomials is a Gröbner basis of the left ideal $\mathcal{I}$ it generates if and only if, whenever $i \neq j$, there exists a choice resolving nondeterminism in the division of the S-polynomial of G_i and G_j by $\mathcal{G}$ that leads to a zero remainder.*

In relation to the forward implication, notice that by Proposition 8, any resolution of nondeterminism leads to a zero remainder.

Proof. Given a Gröbner basis $\mathcal{G}$, let us consider any two of its elements, H_1 and H_2, and their S-polynomial H. The division of H by $\mathcal{G}$ produces a remainder R in $\mathcal{I}$. If it was nonzero, by the definition of a Gröbner basis, its leading monomial would be a multiple of an element of $\mathcal{G}$, contradicting that R is a remainder. So R is nothing but 0, and more generally so do all S-polynomials.

Conversely, let $\mathcal{G} = (G_i)_{1\le i\le m}$ be a family of monic polynomials whose S-polynomials all admit zero as a remainder upon division by $\mathcal{G}$. Further, let F be a nonzero polynomial in the left ideal $\mathcal{I}$ generated by $\mathcal{G}$, which can be written

$$F = \sum_{i=1}^{m} H_iG_i. \tag{8}$$

In particular, H_i is a nonzero polynomial for at least one i. We set $M_i := \mathrm{lm}(H_iG_i)$ for $1 \le i \le m$, and $M := \max_{1\le i\le m} M_i$. Note the inequalities $0 < \mathrm{lm}(F) \le M$. We will show that if $\mathrm{lm}(F) < M$, then we can change the representation of F so as to reduce M. Postponing the proof, we therefore assume the equality $\mathrm{lm}(F) = M$, implying that M is one of the M_i, and that $\mathrm{lm}(F)$ is right divisible by $\mathrm{lm}(G_i)$. This proves that $\mathcal{G}$ is a Gröbner basis.

When $\mathrm{lm}(F) < M$, we can without loss of generality assume that for some integer s,

$$M = M_1 = \cdots = M_s > M_{s+1} \ge \cdots \ge M_m \ge 0,$$

and that $\mathrm{lm}(G_s) = \min_{1\le i\le s} \mathrm{lm}(G_i)$. Then, for each $\ell < s$, there exists w_ℓ such that $\mathrm{lm}(G_\ell) = T^{w_\ell}\,\mathrm{lm}(G_s)$, so that, by assumption, the S-polynomial $G_\ell - T^{w_\ell}G_s$ admits zero as a remainder upon division by $\mathcal{G}$: there is an exact-division formula

$$G_\ell - T^{w_\ell}G_s = \sum_{i=1}^{m} A_{\ell,i}G_i,$$

where the inequality $A_{\ell,i}G_i < \mathrm{lm}(G_\ell) = \mathrm{lm}(T^{w_\ell}G_s)$ holds for each i. By rewriting the G_ℓ in terms of those new sums into (8), we deduce the new expression

$$F = \sum_{\ell=1}^{s-1} H_\ell\left(T^{w_\ell}G_s + \sum_{i=1}^{m} A_{\ell,i}G_i\right) + \sum_{i=s}^{m} H_iG_i = QG_s + R$$

for

$$Q := H_s + \sum_{\ell=1}^{s-1} H_\ell T^{w_\ell}, \qquad R := \sum_{\ell=1}^{s-1}\sum_{i=1}^{m} H_\ell A_{\ell,i}G_i + \sum_{i=s+1}^{m} H_iG_i.$$

Since $M > 0$, note that H_ℓ is nonzero if $\ell < s$. So, for $\ell < s$ and any i, this and the inequality $A_{\ell,i}G_i < \mathrm{lm}(G_\ell)$ imply by Lemma 4 the strict inequality $H_\ell A_{\ell,i}G_i < \mathrm{lm}(H_\ell G_\ell) = M$. For $i > s$, the inequality $H_iG_i < M$ is strict as well. Adding all terms, this implies $R < M$, then, because $F < M$, also $QG_s = F - R < M$. Up to reordering, this makes $QG_s + R$ a new representation of F, with lowered maximal monomial M. □

2.5 A variant of Buchberger's algorithm

Buchberger's algorithm generalizes with minimal alterations.

Theorem 10. *The noncommutative variant of Buchberger's algorithm provided by Algorithm 2 terminates. Moreover, with the breadth-first ordering, it computes a Gröbner basis for the left ideal generated by the input $(F_i)_{1\le i\le s}$.*

Proof. According to Proposition 5, the calls to the division algorithm in Step 3b(2) return. The set $\mathcal{G} := (G_i)_{i=1,\dots,m}$ can change only in Step 3b(3)iii, if the remainder R is nonzero. In this case, the set of the leading monomials of the elements of $\mathcal{G}$ increases at this point. But all encountered monomials in the algorithm have a degree that is not more than the maximal degree in $\mathcal{F}$, primarily because the S-polynomials H considered at Step 3b(2) have this property. So the set $\mathrm{lm}(\mathcal{G})$ cannot grow indefinitely, proving that Step 3b(3)iii can happen only finitely many times. After that, for each S-polynomial H there exists a division of H by $\mathcal{G}$ with remainder 0, so that the algorithm terminates.

Let $\mathcal{G}_f$ be the value of $\mathcal{G}$ output from the algorithm, and consider a pair (G, G') in it, with G appearing as at a smaller index than G'

Input: A finite list $\mathcal{F} = (F_i)_{i=1,\dots,s}$ of nonzero skew polynomials.
Output: A finite list $\mathcal{G} = (G_i)_{i=1,\dots,m}$ of nonzero skew polynomials.

1. $m := s$. For i from 1 to m, do $G_i := \mathrm{lc}(F_i)^{-1} \times F_i$.
2. $\mathcal{P} := \{(G_i, G_j) \mid 1 \le i < j \le m\}$.
3. While $\mathcal{P} \neq \varnothing$ do:
   a. choose a pair (H_1, H_2) and remove it from $\mathcal{P}$;
   b. if one of the leading monomials of the pair divides the other, say, if $\mathrm{lm}(H_2) = T^w \mathrm{lm}(H_1)$:
      (1) compute the S-polynomial $H = H_2 - T^w H_1$,
      (2) divide H by $(G_i)_{i=1,\dots,m}$,
      (3) if the remainder R is not 0 then
         i. $R := \mathrm{lc}(R)^{-1} \times R$,
         ii. $\mathcal{P} := \mathcal{P} \cup \{(G_i, R) \mid 1 \le i \le m\}$,
         iii. set $m := m + 1$, then $G_m := R$.
4. Return $(G_i)_{i=1,\dots,m}$.

Algorithm 2: A variant of Buchberger's algorithm for the noncommutative algebra $k(x)\langle T\rangle$.

in $\mathcal{G}_f$. As the algorithm never removes any element from $\mathcal{G}$, the pair must have been introduced into $\mathcal{P}$ during the execution, and must have later been dealt with. Let $\mathcal{G}_0$ be the value of $\mathcal{G}$ at the time the pair has been considered, and considering the S-polynomial H of G and G'. A possible choice for the division of H by the final set $\mathcal{G}_f$ is, first, to reuse the exact same division steps that led to R, thus using only elements from $\mathcal{G}_0$, and, second, in case R is nonzero, to add with one division step, dividing by the element R of $\mathcal{G}_f$. In all cases, the division obtains zero as its remainder. Therefore, the output $\mathcal{G}$ is a Gröbner basis, as a consequence of Theorem 9. □

2.6 Reduced Gröbner bases

We continue by exploring properties of Gröbner bases that ensure their uniqueness for a fixed ideal I (and breadth-first ordering). The results and proofs of the present section are very similar to those of the classical commutative case.

Definition 11. *A Gröbner basis $\mathcal{G}$ is minimal, respectively reduced, if, for any two polynomials F and G in $\mathcal{G}$, the leading monomial of F does not divide the leading monomial of G, respectively any monomial of G.*

Proposition 12. *Every Gröbner basis $\mathcal{G}$ of a given left ideal I contains a minimal Gröbner basis for the same ideal I. Furthermore, any two minimal Gröbner bases for I have the same number of elements and the same set of leading monomials.*

Proof. Suppose F and G in $\mathcal{G}$ are such that $\mathrm{lm}(G)$ is a left multiple of $\mathrm{lm}(F)$. By transitivity of right divisibility, $\mathcal{G}' := \mathcal{G} \setminus \{G\}$ is another Gröbner basis. Let $H = G - T^w F$ be the S-polynomial of F and G. The division of H by $\mathcal{G}$ cannot involve the divisor G, as leading monomials exclude this possibility, and it has a zero remainder, because the set $\mathcal{G}$ is a Gröbner basis. So G is in the ideal generated by $\mathcal{G}'$. The latter is also a Gröbner basis for I.

Let $\mathcal{F} = (F_i)_{i=1,\dots,n}$ and $\mathcal{G} = (G_i)_{i=1,\dots,m}$ be two minimal Gröbner bases of I. Because $\mathcal{G}$ is a Gröbner basis, the leading monomial $\mathrm{lm}(F_1)$ is divisible by the leading monomial of some G_i. Without loss of generality, we can reindex the family $\mathcal{G}$ so that $\mathrm{lm}(G_1)$ divides $\mathrm{lm}(F_1)$. But $\mathrm{lm}(G_1)$ is by the same argument divisible by some $\mathrm{lm}(F_i)$, so that $\mathrm{lm}(F_i)$ divides $\mathrm{lm}(F_1)$, hence $i = 1$ as $\mathcal{F}$ is minimal. Consequently, $\mathrm{lm}(G_1)$ divides $\mathrm{lm}(F_1)$ and $\mathrm{lm}(F_1)$ divides $\mathrm{lm}(G_1)$, so that they are equal. We continue with $\mathrm{lm}(F_2)$, which neither divides $\mathrm{lm}(F_1)$ on the right nor is a left multiple of it, because the Gröbner basis $\mathcal{F}$ is minimal. As previously, up to some reindexation, we get G_2 satisfying $\mathrm{lm}(F_2) = \mathrm{lm}(G_2)$. The process continues until one of the lists is finished. If there remains an element in the other, say G_m in $\mathcal{G}$, we obtain a contradiction to minimality: $\mathrm{lm}(G_m)$ would be divisible by some leading monomial $\mathrm{lm}(F_i)$, that it to say by $\mathrm{lm}(G_i)$ with $i < m$. □

Input: A Gröbner basis $\mathcal{F} = (F_i)_{i=1,\dots,m}$ of an ideal of $k(x)\langle T\rangle$.
Output: A reduced Gröbner basis $\mathcal{G} = (G_i)_{i=1,\dots,r}$ of the same ideal.

1. $r := m$.
2. While some $\mathrm{lm}(F_i)$ is a left multiple of some $\mathrm{lm}(F_j)$ with $j \neq i$, set $\mathcal{F} := (F_1, \dots, F_{i-1}, F_{i+1}, \dots, F_r)$ and $r := r - 1$.
3. Set $\mathcal{G} := \mathcal{F}$.
4. For i from 1 to r:
   a. $\mathcal{G}' := (G_1, \dots, G_{i-1}, G_{i+1}, \dots, G_r)$;
   b. set G_i to the remainder R of G_i upon division by $\mathcal{G}'$.
5. Return $\mathcal{G} = (G_i)_{i=1,\dots,r}$.

Algorithm 3: Gröbner-basis reduction algorithm.

Both following propositions show that a reduced Gröbner basis of a left ideal generated by a finite set of skew polynomials exists and is unique. Note that the monomial ordering used is the breadth-first ordering and only this one.

Proposition 13. *A reduced Gröbner basis of a left ideal of $k(x)\langle T\rangle$ is unique.*

Proof. Observe that reduced Gröbner bases are minimal, so that their cardinality is fixed, and so are their set of leading monomials. Let $\mathcal{G} = \{G_1, G_2 \dots G_s\}$ and $\mathcal{G}' = \{G'_1, G'_2 \dots G'_s\}$ be two reduced Gröbner bases of the same left ideal. Without loss of generality, we can assume $\mathrm{lm}(G_i) = \mathrm{lm}(G'_i)$ for each i. Suppose that G_i and G'_i are different for some i. Then, the difference $G_i - G'_i$ is in the ideal, and its leading monomial M appears in at least one of G_i and G'_i, strictly below their leading monomials. If in G_i, M is a left multiple of some $\mathrm{lm}(G_j)$ for $j \neq i$, contradicting that $\mathcal{G}$ is reduced. If in G'_i, a similar argument applies. Therefore, $\mathcal{G} = \mathcal{G}'$. □

Proposition 14. *Algorithm 3 computes a reduced Gröbner basis from a Gröbner basis.*

Proof. Let $\mathcal{I}$ be the input ideal. The first two steps of Algorithm 3 replace $\mathcal{F}$ by some minimal Gröbner basis generating the same ideal by the method implicit in the proof of Proposition 12. Observe that the successive values of $\mathcal{G}$ along the loop at Step 4 are all minimal Gröbner bases of $\mathcal{I}$, with the family $\mathrm{lm}(\mathcal{G})$ kept invariant, as a result of G_i and R sharing the same leading monomial at Step 4b. Additionally, for each i, the remainder R write $\mathrm{lm}(G_i) - Q$ where Q involves no left multiple of any element of $\mathrm{lm}(\mathcal{G}')$, and in fact of any element of $\mathrm{lm}(\mathcal{G})$ as $Q < G_i$. As a result, the final family $\mathcal{G}$ is a reduced Gröbner basis of $\mathcal{I}$. □

Example 15. Let us consider the family of skew polynomials

$$T_{4,3} + \frac{1}{1-2x}T_{4,2} + \frac{x}{1-x^2}T_{2,1} + T_{2,0}, \quad T_{8,3} + \frac{1}{1-x}T_{8,2},$$
$$xT_{8,4} + \frac{2-x}{1-x}T_{4,2} + T_{4,0}, \quad T_{8,1} + T_{8,0} + T_{4,3}, \quad T_{8,1} + T_{8,2} + T_{8,0},$$
$$-\frac{x^3}{1-x^4}T_{8,2} + T_{8,4} + T_{2,1}, \quad T_{8,5} + \frac{x^2}{1-3x}T_{8,4} + T_{2,1}.$$

An instance of execution of Algorithm 2 begins by considering the pair between the first two polynomials, because of the relation $T_{8,3} = T_{2,0}T_{4,3}$. This provides the S-polynomial

$$T_{8,6} - \frac{2x^5 - 8x^4 - 3x^3 - 3x^2 - 9x + 6}{2x^4(2x^2 - 3x + 1)}T_{4,2} + \frac{4x-1}{2-2x}T_{2,0}$$
$$- \frac{4x^4 + 2x^3 + 3x^2 + 3x + 3}{2x^4}T_{4,0} + \frac{3x^6 + 4x^5 - 4x^4 - 3x^2 + 3}{2x^3(x^3 - x^2 - x + 1)}T_{2,1}.$$

The computation results in a Gröbner basis with 14 elements, whose leading monomials are: $T_{8,3}$, $T_{8,5}$, $T_{8,1}$, $T_{8,6}$, $T_{8,2}$, $T_{8,4}$, $T_{8,0}$, $T_{4,3}$, $T_{4,1}$, $T_{4,2}$, $T_{4,0}$, $T_{2,1}$, $T_{2,0}$. One of the polynomials in the basis has rational functions coefficients with degree 31 and numerical coefficients of order 10^{11}. There were 81 pairs dealt with. Among them, 21 gave S-polynomials and 14 of the 21 S-polynomials reduced to 0. After reduction by Algorithm 3, we find the Gröbner basis $\{T_{2,0}, T_{2,1}\}$.

3 THE LINEAR ALGEBRA APPROACH

In this section, we develop an algorithm reminiscent of Faugère's algorithm F_4 [6], but properties of section operators departing from those of commutative polynomials make specific variations needed. First, it results directly from the properties of division and the definition of S-polynomials that our variant of Buchberger's algorithm, Algorithm 2, performs all its calculations on an input $\mathcal{F}$ in the $k(x)$-vector space generated by the monomials $T^w \leq \max \mathrm{lm}(\mathcal{F})$. Second, divisions tend to involve dense polynomials, owing to the relation (6), which is amplified by the exponential growth with d of the number of monomials T^w of degree $|w| = d$.

Consequently, it seems adequate to perform a calculation that is incremental in the way of F_4, but with the unusual property of being confined in a finite-dimensional vector space known beforehand.

Given a finite set of generators of an ideal, we use the basis $\mathcal{B}$ of all monomials that are not larger than an adequate monomial T^u. Then, any polynomial $F \leq T^u$ in $k(x)\langle T\rangle$ can be represented by the row vector $V = \mathrm{mat}_{\mathcal{B}}(F)$, and conversely, any row vector V represents a polynomial $F = \mathrm{poly}_{\mathcal{B}}(V) = \sum_{v \leq u} V_v T^v$. By viewing matrices as families of rows, indexed by integers, a similar bijection is in place between families $\mathcal{F}$ of s polynomials and rectangular matrices M with s rows. We write $M = \mathrm{mat}_{\mathcal{B}}(\mathcal{F})$ and $\mathcal{F} = \mathrm{poly}_{\mathcal{B}}(M)$ accordingly. Furthermore, we extend the notion of leading monomial to vectors through these bijections, that is, we define $\mathrm{lm}(V) := \mathrm{lm}(\mathrm{poly}_{\mathcal{B}}(V))$. In the previous discussion, all (row) vectors and matrices have columns indexed by the words v such that $\varepsilon \leq v \leq u$. For pivoting considerations in linear algebra, we view those columns as sorted according to decreasing v, that is, so to say with u to the left and ε to the right.

As already emphasized in Section 2.4, the notion of S-polynomial is very particular in our context. A pair (H_1, H_2) of polynomials admits a nonzero S-polynomial $H_2 - T^w H_1$ only if $\mathrm{lm}(H_2) = T^w \mathrm{lm}(H_1)$ (up to order). A direct analogue of Faugère's "half pairs" would therefore consist of both polynomials H_2 and $T^w H_1$. But when we get to consider a pair (H_1, H_2) in our variant of the F_4 algorithm, the polynomial H_2 is already in the polynomial list $\mathrm{poly}_{\mathcal{B}}(R)$ of polynomials available as divisors. So, it suffices to add $T^w H_1$ to the list P of new half pairs. This motivates that our definition of HalfPairs intentionally forgets H_2.

Input: A finite family $\mathcal{F}$ of skew polynomials.
Output: A Gröbner basis $\mathcal{G}$ of the left ideal generated by $\mathcal{F}$.

1. Set $\mathcal{B} := (T^v)_{u \geq v \geq \varepsilon}$ for u such that $T^u = \max \mathrm{lm}(\mathcal{F})$.
2. $R := \mathrm{RowEchelon}((\mathrm{mat}_{\mathcal{B}}(\mathrm{lc}(F)^{-1} \times F))_{F \in \mathcal{F}})$.
3. $P := \mathrm{Preproc}(\mathrm{HalfPairs}(R, R), R)$.
4. While $P \neq \varnothing$ do
   a. $R^0 :=$ R augmented by stacking it above P,
   b. $R := \mathrm{RowEchelon}(R^0)$,
   c. $R^+ :=$ the rows V of R such that $\mathrm{lm}(V)$ is not in $\mathrm{lm}(R^0)$,
   d. $P := \mathrm{Preproc}(\mathrm{HalfPairs}(R, R^+), R)$.
5. Return $\mathcal{G} = \mathrm{poly}_{\mathcal{B}}(R)$.

where:

* $\mathrm{HalfPairs}(R^1, R^2)$ returns the rows $\mathrm{mat}_{\mathcal{B}}(T^w \mathrm{poly}_{\mathcal{B}}(V'))$ satisfying $\mathrm{lm}(V) = T^w \mathrm{lm}(V')$ for some word w, some row V in R^1 or R^2, and some row V' in the other one.
* $\mathrm{RowEchelon}(M)$ returns the variant of a row echelon form of M obtained by reducing each row by the rows above it, without interchanging any rows, but removing null rows, and by using leading coefficients of rows as pivots.
* $\mathrm{Preproc}(P, R)$ takes a monomial that appears in $\mathrm{poly}_{\mathcal{B}}(P)$ but not in $\mathrm{lm}(R \cup P)$, and is expressible as a product $T^w \mathrm{lm}(V)$ for some word w, some row V in R, then adds $\mathrm{mat}_{\mathcal{B}}(T^w \mathrm{poly}_{\mathcal{B}}(V))$ to P, and repeats until no such product can be added.

Algorithm 4: A variant of the F_4 algorithm for the noncommutative algebra $k(x)\langle T\rangle$.

Theorem 16. *The variant of the F_4 algorithm provided by Algorithm 4 terminates and returns a Gröbner basis of the left ideal of $k(x)\langle T\rangle$ generated by its input.*

Proof. The successive matrices R at Step 4b generate an increasing family of $k(x)$-vector spaces of rows, all of dimension at most the cardinality of $\mathcal{B}$. The termination of the algorithm is then immediate: as the span of R cannot grow indefinitely, at some point R^+ is empty, forcing P to be empty as well at the next step.

After the initialization of R at Step 2, the left ideal generated by $\mathrm{poly}_{\mathcal{B}}(R)$ is exactly the ideal generated by the input $\mathcal{F}$. Whether it be after Step 3 or Step 4d, $\mathrm{poly}_{\mathcal{B}}(P)$ contains only elements of the ideal generated by $\mathrm{poly}_{\mathcal{B}}(R)$. The ideal generated by $\mathrm{poly}_{\mathcal{B}}(R^0)$ is therefore equal to that generated by $\mathrm{poly}_{\mathcal{B}}(R)$, and this ideal is left unchanged upon changing R^0 to $\mathrm{RowEchelon}(R^0)$. We get that the ideal generated by $\mathrm{poly}_{\mathcal{B}}(R)$ remains unchanged after Steps 4a and 4b. By induction, this ideal, and therefore the ideal generated by the output $\mathcal{G}$, is the ideal generated by the input $\mathcal{F}$.

Next, the construction of R^0 at Step 4a and the definition of RowEchelon are so that the matrix R obtained at Step 4b is equal to the matrix R before, stacked above the matrix R^+ that will be extracted at Step 4c. Thus, any row vector introduced into R by Step 2 or 4b will remain there until the end of the algorithm.

problem	01	35	38	14	39	42	18	15	43
radix	2	2	3	2	3	2	3	2	2
deg/dim	3/14	6/127	4/161	5/63	5/485	4/31	4/161	6/127	5/63
#in/#out	7/2	5/5	5/5	5/5	5/5	24/1	4/4	6/6	48/1
Buchberger	0.29	1.89	2.09	0.46	9.10	4.90	1.64	1.98	69.95
F4	0.26	0.65	0.77	2.76	2.86	5.39	9.68	25.50	77.41
speed-up	1.09	2.91	2.70	0.17	3.18	0.91	0.17	0.08	0.90

Table 1: Selected timings, comparing the speeds of Algorithm 4 (F4) and Algorithm 2 (Buchberger). Our running example (Examples 15 and 17) corresponds to problem 01.

Finally, consider any two polynomials H_1 and H_2 of the output $\mathcal{G}$ satisfying $\operatorname{lm}(H_1) = T^w \operatorname{lm}(H_2)$ for some word w. This w is nonempty since $\operatorname{lm}(\mathcal{G})$ has no repeated no element. If both row vectors $V_1 = \operatorname{mat}_{\mathcal{B}}(H_1)$ and $V_2 = \operatorname{mat}_{\mathcal{B}}(H_2)$ were introduced at Step 2, they are considered at Step 3 to produce the half pair $\operatorname{mat}_{\mathcal{B}}(T^w H_2)$. Otherwise, the most recent of V_1 and V_2 was introduced at Step 4b and both vectors are considered at Step 4d to produce the half pair $\operatorname{mat}_{\mathcal{B}}(T^w H_2)$. In both cases, P is thus nonempty and the calculation continues to Step 4a with both V_1 and V_2 in R. After 4b, they are still in R, and $\operatorname{mat}_{\mathcal{B}}(T^w H_2)$ is a linear combination of the rows of R. As a consequence, the remainder of the division of the S-polynomial $H_1 - T^w H_2$ by $\operatorname{poly}_{\mathcal{B}}(R)$ is zero, and so is the remainder under division by $\mathcal{G}$. By Theorem 9, $\mathcal{G}$ is a Gröbner basis. □

Example 17 (Example 15 continued)**.** The maximum monomial is $T_{8,3} = T^{011}$ so that we use the basis of the T^w with $w = 011, 101, 001, 110, \dots, 0, \varepsilon$, that is $T_{8,3}, T_{8,5}, T_{8,1}, T_{8,6}, \dots, T_{2,0}, T_{1,0}$. The leading monomials of the input polynomials are $T_{4,3}$ and $T_{8,r}$ with $1 \le r \le 5$. The row echelon reduction at Step 2 brings up the monomial $T_{4,2}$. As $T_{4,3}$ divides $T_{8,3}$ and $T_{4,2}$ divides $T_{8,2}$, Step 3 computes two half pairs that are rows with leading monomials $T_{8,3}$ and $T_{8,2}$. Preprocessing adds a row with leading monomial $T_{8,6}$, resulting in P consisting of 3 rows. After stacking R and P at the first execution of the loop, the reduction at Step 4b discovers the monomials $T_{8,0}$ and $T_{4,1}$, hence the matrix R^+ at Step 4c has two rows. Next, Step 4d produces three half pairs with leading monomials $T_{8,6}$, $T_{8,1}$, and $T_{8,5}$, before preprocessing finds no row to be added, resulting in P consisting of only 3 rows. At this point, the matrix contains polynomials with maximal degree 14. It takes 3 executions of the main loop before the computation ends and returns a Gröbner basis with 13 polynomials, whose leading monomials are in fact all the elements of the basis $\mathcal{B}$ except for $T^\varepsilon = T_{1,0}$. The polynomials in intermediate calculations have degrees up to 19 and use integer coefficients up to $\approx 3.7\,10^{19}$.

4 IMPLEMENTATION AND EXPERIMENT

We implemented Algorithms 2 and 4 in Maple and computed reduced Gröbner bases of over 40 ideals. The script and the data are available at https://specfun.inria.fr/chyzak/gbdacr/. The timings obtained (Table 1) do not indicate any clear advantage of F4.

5 CONCLUSION

We have achieved our initial goal of a theory of Gröbner bases for divide-and-conquer systems. To the best of our knowledge, this is the first time such a theory has been developed in a context involving noncommutative words and twisted commutation rules simultaneously. We could overcome the difficulty that the leading monomial of a polynomial product need not be the product of the leading monomials.

As to efficiency, the contribution of the F_4 algorithm is unclear. It needs to be further studied in relation to other ingredients: rejection criteria; an incremental selection strategy of half pairs; modular variants of $k(x)$ compatible with the action of sections operators.

On the other hand, our theory extends to an algorithmic module theory, which we use in applications involving nonhomogeneous recurrence equations and systems. This will be developed elsewhere.

Finally, remark that Example 15 provides a system whose series solutions are all zero, although the ideal does not contain 1. Any series annihilated by the computed Gröbner basis, $\{T^0, T^1\}$, has indeed odd and even parts that are zero, and so is zero. Recovering 1 in the ideal is possible if one extends the algebra with a generator M to represent the Mahler operator, acting on series by $M \cdot f(x) = f(x^b)$. When $b = 2$, the action on series leads to the identity $1 = MT^0 + xMT^1$ to be enforced in the algebra. However, it also leads to $T^0 M = 1$ and $T^1 M = 0$, hence to an algebra with zero divisors. We have not tried to develop a Gröbner-basis theory for it.

REFERENCES

[1] J. Apel and W. Lassner. An extension of Buchberger's algorithm and calculations in enveloping fields of Lie algebras. *J. Symbolic Comput.*, 6(2-3):361–370, 1988.
[2] T. Becker and V. Weispfenning. *Gröbner Bases.* Springer, 1993.
[3] F. Chyzak and B. Salvy. Non-commutative elimination in Ore algebras proves multivariate identities. *Journal of Symbolic Computation*, 26(2):187–227, 1998.
[4] D. A. Cox, J. Little, and D. O'Shea. *Ideals, Varieties, and Algorithms.* Springer, 2015.
[5] P. Dumas. Asymptotic expansions for linear homogeneous divide-and-conquer recurrences. *Theoretical Computer Science*, 548:25–53, Sept. 2014.
[6] J.-C. Faugère. A new efficient algorithm for computing Gröbner bases (F4). *Journal of Pure and Applied Algebra*, 139(1–3):61–88, June 1999.
[7] A. Galligo. Some algorithmic questions on ideals of differential operators. In *Eurocal'85, Vol. 2 (Linz, 1985)*, volume 204 of *Lecture Notes in Comput. Sci.*, pages 413–421. Springer, 1985.
[8] E. L. Green. An introduction to noncommutative Gröbner bases. In K. G. Fischer, P. Loustaunau, J. Shapiro, E. L. Green, and D. Farkas, editors, *Computational Algebra*, volume 151 of *Lecture Notes in Pure and Appl. Math.*, pages 167–190. Dekker, 1994. Proc. of the 5th Mid-Atlantic Algebra Conference (1993).
[9] H.-K. Hwang, S. Janson, and T.-H. Tsai. Exact and asymptotic solutions of a divide-and-conquer recurrence dividing at half. *ACM Transactions on Algorithms*, 13(4):1–43, Dec. 2017.
[10] A. Kandri-Rody and V. Weispfenning. Noncommutative Gröbner bases in algebras of solvable type. *Journal of Symbolic Computation*, 9(1):1–26, 1990.
[11] H. Kredel. *Solvable Polynomial Rings.* Reihe Mathematik. Shaker, Germany, 1993.
[12] R. La Scala and V. Levandovskyy. Skew polynomial rings, Gröbner bases and the letterplace embedding of the free associative algebra. *J. Symbolic Comput.*, 48:110–131, 2013.
[13] D. Lazard. Gröbner bases, Gaussian elimination and resolution of systems of algebraic equations. In *Lecture Notes in Comput. Sci.*, pages 146–156. 1983.
[14] V. Levandovskyy and H. Schönemann. Plural: a computer algebra system for noncommutative polynomial algebras. In *Proceedings of ISSAC'03 (Philadelphia, USA).* ACM Press, 2003.
[15] E. L. Mansfield and A. Szanto. Elimination theory for differential difference polynomials. In *Proceedings of ISSAC'03 (Philadelphia, USA).* ACM Press, 2003.
[16] F. Mora. Groebner bases for non-commutative polynomial rings. In *Algebraic Algorithms and Error-Correcting Codes*, pages 353–362. Springer, 1986.
[17] T. Mora. Standard bases and non-noetherianity: Non-commutative polynomial rings. In T. Beth and M. Clausen, editors, *Applicable Algebra, Error-Correcting Codes, Combinatorics and Computer Algebra*, pages 98–109. Springer, 1988.
[18] T. Mora. Groebner bases in non-commutative algebras. In *Proceedings of ISSAC'89 (Portland, USA)*, pages 150–161. Springer, 1989.
[19] N. Takayama. Gröbner basis and the problem of contiguous relations. *Japan Journal of Applied Mathematics*, 6(1):147–160, 1989.
[20] V. A. Ufnarovskiĭ. On the use of graphs for computing a basis, growth and Hilbert series of associative algebras. *Math. Sb.*, 68(2):417–428, 1991.
[21] V. Weispfenning. Finite Gröbner bases in non-Noetherian skew polynomial rings. In *Proceedings of ISSAC'92 (Berkeley, USA).* ACM Press, 1992.

Bounds for Degrees of Minimal $\mu-$bases of Parametric Surfaces

Teresa Cortadellas
Universitat de Barcelona, Facultat d'Educació
Barcelona, Spain
terecortadellas@ub.edu

Carlos D'Andrea
Universitat de Barcelona, Departament de Matemàtiques i Informàtica
Barcelona, Spain
cdandrea@ub.edu

M. Eulàlia Montoro
Universitat de Barcelona, Departament de Matemàtiques i Informàtica
Barcelona, Spain
eula.montoro@ub.edu

ABSTRACT

By adapting the effective version of Quillen-Suslin Theorem given in [8], we show that if the ideal defining a rational parametrization of degree d of an algebraic surface in 3-dimensional space is radical and has D points, then a μ-basis of this parametrization can be found of degree bounded by $5\max(1, D-1)^4(2d+1)^4$. This bound improves those obtained recently in [4] in our setup, and it is also sensitive to the number of base points.

KEYWORDS

μ-bases, syzygies, parametrization, Quillen-Suslin Theorem, effective bounds

ACM Reference Format:
Teresa Cortadellas, Carlos D'Andrea, and M. Eulàlia Montoro. 2020. Bounds for Degrees of Minimal $\mu-$bases of Parametric Surfaces. In *International Symposium on Symbolic and Algebraic Computation (ISSAC '20), July 20–23, 2020, Kalamata, Greece.* ACM, New York, NY, USA, 7 pages. https://doi.org/10.1145/3373207.3404039

1 INTRODUCTION

The concept of $\mu-$basis was introduced in [5] in the case of parametrized rational curves. Let $\mathbb{K}$ be a field, s an indeterminate over $\mathbb{K}$ and $n \in \mathbb{N}$. An $(n+1)$-tuple $P(s) = (a_1(s), \ldots, a_{n+1}(s)) \in \mathbb{K}[s]^{n+1}$ can be regarded as the parametrization of a rational curve in $\mathbb{K}^n$ via the map $\mathbb{K} \dashrightarrow \mathbb{K}^n$ given by $\left(\frac{a_1(s)}{a_{n+1}(s)}, \ldots \frac{a_n(s)}{a_{n+1}(s)}\right)$. With this in mind, we can assume w.l.o.g. that $gcd(a_i(s)) = 1$. The sygyzy module of P over $\mathbb{K}[s]$ is defined as

$$Syz(P) = \{(A_1(s), \ldots, A_{n+1}(s)) \in \mathbb{K}[s]^{n+1} : \sum_{i=1}^{n+1} A_i(s)a_i(s) = 0\}.$$

This module is free of rank n.

Assume that $d = \max(\deg(a_i)) \geq 1$. By applying the Extended Euclidean Algorithm to the input (see [5] and [10]) one can find a basis $\{p_1(s), \ldots, p_n(s)\}$ of $Syz(P)$ such that $\deg(p_i(s)) = \mu_i$ with $\mu_1 + \ldots + \mu_n = d$. Such a basis is called a *μ-basis* in the literature. So, essentially, μ-bases are bases of $Syz(P)$ of controlled degree. For algorithms to compute $\mu-$bases of curves, see [3, 5, 10, 12].

ISSAC '20, July 20–23, 2020, Kalamata, Greece

ACM ISBN 978-1-4503-7100-1/20/07…$15.00
https://doi.org/10.1145/3373207.3404039

To show that the degree of a μ-basis is sharp in the sense that the condition $\sum_{i=1}^{n} \deg(p_i(s)) < d$ can never happen for any basis $\{p_1(s), \ldots, p_n(s)\}$ of $Syz(P)$, some finer results from Commutative Algebra are needed. We state it here for convenience of the reader, as it will also be used later in the proof of our main result.

Theorem 1.1. *(Hilbert-Burch Theorem [7, Theorem 3.2]) Suppose that an ideal I in a Noetherian ring R admits a free resolution of length 1 as follows:*

$$0 \to F \xrightarrow{M} G \to I \to 0.$$

If the rank of the free module F is n, then the rank of G is $n+1$, and there exists a nonzero divisor $a \in R$ such that I is equal to a times the ideal of $n \times n$ minors of the matrix M with respect with any given bases of F and G. The generator of I that is the image of the i-th basis vector of G is $\pm a$ times the determinant of the submatrix of M formed from all except the i-th row. Moreover, the grade of the ideal of maximal minors is 2.

Conversely, given a $(n+1) \times n$ matrix M with entries in R such that the grade of the ideal of $n \times n$ minors of M is at least 2, and a given nonzero divisor $a \in R$, the ideal I generated by a times the $n \times n$ minors of M admits a free resolution of length one as above. It has grade 2 if and only if a is a unit.

By applying the Hilbert-Burch Theorem to the case $R = \mathbb{K}[s, s_0]$ with s_0 being a new indeterminate over $\mathbb{K}$, and I being the ideal generated by the homogeneization of the $a_i(s)$, $1 \leq i \leq n+1$, we will deduce that

$$P(s, s_0) = p_1(s, s_0) \wedge \ldots \wedge p_n(s, s_0), \tag{1}$$

where each of the objects appearing above are the homogeneizations of those defined in one variable. The wedge product notation here only means that the coordinates of $P(s, s_0)$ are -up to a nonzero constant in $\mathbb{K}$- equal to the signed maximal minors of the $(n+1) \times n$ matrix having in the i-th column the coordinates of $p_i(s, s_0)$.

From (1) we deduce straightforwardly that

$$\deg(P(s, s_0)) = \sum_{i=1}^{n+1} \deg(p_i(s, s_0)),$$

which shows that the degree in μ-bases for curves is sharp.

For parametric surfaces the situation is more complicated as no Euclidean Algorithm is possible in more than one variable. Yet μ-bases exist, and the Hilbert-Burch Theorem still applies in this case. Let t be another indeterminate over $\mathbb{K}$. An $(n+1)$-tuple $P(s, t) = (a_1(s, t), \ldots, a_{n+1}(s, t)) \in \mathbb{K}[s, t]^{n+1}$ can be regarded as the parametrization of a surface in $\mathbb{K}^n$ as before. So, we can assume again that $gcd(a_i(s, t)) = 1$. In the Appendix of [2], it is shown that

the syzygy module

$$Syz(P) = \{(A_1(s,t), \ldots, A_{n+1}(s,t)) \in \mathbb{K}[s,t]^{n+1} : A_1(s,t)a_1(s,t) + \ldots + A_{n+1}(s,t)a_{n+1}(s,t) = 0\}$$

is also free of rank n. In that paper, a *μ-basis* of $P(s,t)$ was defined as *any* basis of $Syz(P)$. No were neither required nor deduced on the degrees of any of these bases. A *minimal μ-basis* was defined as a basis $\{p_1(s,t), \ldots, p_n(s,t)\}$ of $Syz(P)$ such that $\sum_{i=1}^{n} \deg(p_i(s,t))$ is minimal among all the bases of $Syz(P)$, and the question on explicit bounds on the degree of such a minimal μ-basis was raised. Algorithms to compute μ-bases for this case can be found in [6], but no bounds on the degree of these elements can be easily derived from these algorithms.

In [4] the first of such bounds is produced for surfaces in $\mathbb{K}^3$, i.e. when $n = 3$. Indeed, it is shown in [4, Theorem A] that a minimal μ-basis in this situation has degree bounded by $O(d^{33})$. Several subcases were considered with better bounds in all of them. However, it is not clear yet whether these bounds are sharp, and there is definitely a lot of room for improvements.

In this paper, we present one of such sharpenings. To keep the notation simple and also to compare with previous results, we set ourselves in the case $n = 3$, but the generalization to any n is straightforward. So, for the rest of the text, we will deal with a parametrization

$$P(s,t) = (a_1(s,t), a_2(s,t), a_3(s,t), a_4(s,t)) \in \mathbb{K}[s,t]^4 \qquad (2)$$

with $\deg(P) = \max(\deg(a_i(s,t))) = d$ and $gcd(a_i(s,t)) = 1$. We denote with $I_P = \langle a_1(s,t), a_2(s,t), a_3(s,t), a_4(s,t)\rangle \subset \mathbb{K}[s,t]$ the ideal defined by these polynomials, and $V_P \subset \overline{\mathbb{K}}^2$ the variety defined by this ideal in the algebraic closure of $\mathbb{K}$. Note that the condition $gcd(a_i(s,t)) = 1$ implies that V_P is a finite set. Let D be its *degree*, meaning the number of points in V_P counted with their corresponding multiplicities.

It is known that a radical zero-dimensional ideal $I \subset \mathbb{K}[s,t]$ has -after possibly a linear change of coordinates- system of generators of the form $\{p(s), t - q(s)\}$. The converse of course holds for a larger class of ideals containing those radical and zero-dimensional, and have been characterized geometrically in [1]. These ideals are said to have a *shape basis*.

We will use the effective version of Quillen-Suslin Theorem given in [8] to give an algorithmic proof of the following result:

Theorem 1.2. *Let $P(s,t)$ be as in (2) with $d = \deg(P)$ and $D = \deg(V_P)$. If I_P has a shape basis, then a μ−basis $\{p_1(s,t), p_2(s,t), p_3(s,t)\}$ of $P(s,t)$ can be found with degree bounded by*

$$5\max(1, D-1)^4(2d+1)^4. \qquad (3)$$

As a consequence of this result, if $V_P = \emptyset$ (i.e. $I_P = \mathbb{K}[s,t]$ thanks to Hilbert's Nullstellensatz), we have that $D = 0$ and hence (3) boils down to a bound of the size $O(d^4)$ which is the one appearing in [8] for this situation, and refines the amount $O(d^{22})$ obtained in [4] for this case.

In order to obtain a bound only depending on d, note that by Bézout Theorem we always have $D \leq d^2$, and hence (3) is always bounded by a quantity of the size of $O(d^{12})$. This is also a major improvement over the results in [4]. It should be said however, that our results are restricted to the case of I_P having a shape basis, and our techniques depend strongly on the particular properties of this kind of ideals, so no extensions to the general case seem to be deduced from our approach and extra ideas are needed to improve the bounds already known. On the plus side, by following the steps of our proof one can compute a μ-basis of $P(s,t)$ with such bounds in the degree, see Algorithm 4.1.

The paper is organized as follows: in Section 2 we will revisit the effective version of Quillen-Suslin Theorem given in [8] to obtain bounds for a minimal μ-basis in the case $V_P = \emptyset$. In Section 3 we will prove the general case by reducing it to the situation of Quillen-Suslin. All our steps are computationally feasible so we collect all of them in Algorithm 4.1 in Section 4, where a running example is also provided.

2 THE UNIMODULAR CASE

Recall that a matrix in $\mathbb{K}[s,t]^{n\times m}$ is said to be *unimodular* if the ideal generated by its maximal minors is the whole ring $\mathbb{K}[s,t]$. We will consider P as a 1×4 matrix, and consider the situation when P is unimodular which equivalently means that $I_P = \mathbb{K}[s,t]$ or $V_P = \emptyset$.

Theorem 2.1. *[8] If $P(s,t)$ being as in (2) is a unimodular matrix, there exists a unimodular matrix $M \in \mathbb{K}[s,t]^{4\times 4}$ of degree $O(d^4)$, such that*

$$P(s,t)\,M = (1,0,0,0). \qquad (4)$$

Corollary 2.2. *Let M be the matrix of above, and write $M = (M^1\, M^2\, M^3\, M^4)$, with M^i being the i-th column of M. Then, the set $\{M^2, M^3, M^4\}$ is a μ−basis of $Syz(P)$.*

Proof. We clearly have $P \cdot M^i = 0$ for $i = 2,3,4$. Also, these columns are $\mathbb{K}[s,t]$-linearly independent as they are part of a matrix of full rank. To show that they generate $Syz(P)$, let A be any element in this module, as M is unimodular, we have that $\{M^1, M^2, M^3, M^4\}$ generates $\mathbb{K}[s,t]^4$, and hence we get $A = \sum_{j=1}^{4} p_j M^j$ with $p_j \in \mathbb{K}[s,t]$, $j = 1,2,3,4$. As $P \cdot A = 0$, from (4) we deduce straightforwardly that $p_1 = 0$, which then implies that $A \in \langle M^2, M^3, M^4\rangle$. This concludes with the proof of the claim. □

We will review now the algorithm proposed in [8] to compute such a matrix M. This will give explicit bounds for the degrees of the elements of a μ-basis thanks to Corollary 2.2. We will assume that $\mathbb{K}$ is infinite, otherwise we can work in an extension of it. Suppose w.l.o.g. that $\deg_t(a_1) = d$. If this is not the case, we can redefine a_1 with this property after a linear combination of the a_i's, and if necessary also after applying the change of variable $\tilde{s} = s + \lambda t$ with $\lambda \in \mathbb{K} \setminus \{0\}$.

Lemma 2.3. *If $I_P = \mathbb{K}[s,t]$, then there exist $\alpha_3, \alpha_4 \in \mathbb{K}$ and $\beta_2, \beta_3, \beta_4 \in \mathbb{K}$, such that by setting $\tilde{a}_2 := a_2 + \alpha_3 a_3 + \alpha_4 a_4$, and $\tilde{a}_3 := \beta_2 a_2 + \beta_3 a_3 + \beta_4 a_4$, the polynomials $r_{12}(s) := Res_t(a_1(s,t), \tilde{a}_2(s,t))$ and $r_{13}(s) := Res_t(a_1(s,t), \tilde{a}_3(s,t))$ are coprimes.*

Proof. If $r_{12}(s) = Res_t(a_1(s,t), \tilde{a}_2(s,t)) \equiv 0\, \forall \alpha_3, \alpha_4 \in \mathbb{K}$, then $a_1(s,t)$ and $\tilde{a}_2(s,t)$ would have a common factor $g(s,t)$ of positive degree in t which divides a_1, a_2, a_3 and a_4. This is a contradiction because we are assuming $V(a_1, a_2, a_3, a_4) = \emptyset$.

So, we can find $\alpha_3, \alpha_4 \in \mathbb{K}$ such that $r_{12}(s) \neq 0$, and this implies that $V(a_1, \tilde{a}_2)$ is finite. If it is empty, the claim follows straightforwardly. Otherwise, we list its elements:

$$V(a_1, \tilde{a}_2) = \{(s_i, t_i), i = 1, \ldots, l\} \subset \mathbb{K}^2.$$

Set now $S = \{(s_i, t_{ij}) \mid a_1(s_i, t_{ij}) = 0,\ i = 1, \ldots, l\}$.

Since $V(a_1, \tilde{a}_2, a_3, a_4) = \emptyset$ and $\mathbb{K}$ is infinite, there must exist $(\beta_2, \beta_3, \beta_4)$ such that

$$(\beta_2, \beta_3, \beta_4) \notin \cup_{(s_i,t_{ij})\in S} \left\langle \left(\tilde{a}_2(s_i, t_{ij}), a_3(s_i, t_{ij}), a_4(s_i, t_{ij})\right)\right\rangle^{\perp}.$$

For these values of the β_i's, the claim follows straitghtforwardly. □

Theorem 2.4. *Let $P(s,t)$ be as in (2) with $1 \in I_P$. Then, a μ−basis of $Syz(P)$ has degree bounded by $4d^4$.*

Proof. Assume w.l.o.g. that a_1, a_2 and a_3 have already been modified according to the hypothesis of Lemma 2.3. We will apply the constructive method given in [8] for the proof of Theorem 2.1 with two steps:

$$\begin{aligned}&(a_1(s,t), a_2(s,t), a_3(s,t), a_4(s,t))\\ \xrightarrow{step1}\ &(a_1(s,B'), a_2(s,B'), a_3(s,B'), a_4(s,B'))\\ \xrightarrow{step2}\ &(a_1(s,0), a_2(s,0), a_3(s,0), a_4(s,0))\end{aligned}$$

For the first step we compute

$$r_{12}(s) = Res_t(a_1(s,t), a_2(s,t)) \text{ and } r_{13}(s) = Res_t(a_1(s,t), a_3(s,t)).$$

By Bézout's Identities, there exist $A_1(s,t)$, $A_2(s,t)$, $A_1^*(s,t)$, and $A_2^*(s,t) \in \mathbb{K}[s,t]$ such that

$$A_1(s,t)a_1(s,t) + A_2(s,t)a_2(s,t) = r_{12}(s)$$

and

$$A_1^*(s,t)a_1(s,t) + A_2^*(s,t)a_3(s,t) = r_{13}(s)$$

with

$$\boxed{\deg(r_{12}), \deg(r_{13}) \leq d^2, \quad \deg(A_i(s,t)), \deg(A_i^*(s,t)) \leq d^2 - d.}$$

By Lemma 2.3, $r_{12}(s)$ and $r_{13}(s)$ are coprimes. So, again by Bézout's Identity, there exist $R_{12}(s)$, $R_{13}(s) \in \mathbb{K}[s]$ such that

$$R_{12}(s)r_{12}(s) + R_{13}(s)r_{13}(s) = 1, \tag{5}$$

and multiplying by t the two sides we get

$$R_{12}(s)r_{12}(s)t + R_{13}(s)r_{13}(s)t = t.$$

Therefore in Step 1 we can take $\begin{cases} B = t, \\ B' = R_{13}(s)r_{13}(s)t \end{cases}$

Note that we have

$$\max(\deg(R_{12}(s)r_{12}(s)), \deg(R_{13}(s)r_{13}(s))) \leq 2d^2 - 1,$$

therefore

$$\boxed{\deg(B) = 1, \deg(B') \leq 2d^2}$$

Continuing with step 1 of the algorithm, we compute a unimodular matrix $N_1 \in \mathbb{K}[s,t]^{4\times 4}$ such that

$$P(s,t)N_1 = P(s,B').$$

This matrix will be obtained as a product of two matrices, $N_1 = E_1S_1$, where $E_1 \in \mathbb{K}[s,t]^{4\times 4}$ satisfies

$$P(s,t)E_1 = (a_1(s,t), a_2(s,t), a_3(s,B'), a_4(s,B')),$$

and $S_1 \in \mathbb{K}[s,t]^{4\times 4}$ is such that

$$(a_1(s,t), a_2(s,t), a_3(s,B'), a_4(s,B'))S_1 = P(s,B').$$

To be more precise,

$$E_1 = \begin{pmatrix} I_2 & E_{12} \\ 0 & I_2 \end{pmatrix},$$

where I_2 is the identity 2×2 matrix, and

$$E_{12} = \begin{pmatrix} -\alpha(s,t)A_1(s,t) & -\beta(s,t)A_1(s,t) \\ -\alpha(s,t)A_2(s,t) & -\beta(s,t)A_2(s,t) \end{pmatrix},$$

with

$$\alpha(s,t) = \frac{1}{r_{12}(s)}(a_3(s,t) - a_3(s,B'))$$

and

$$\beta(s,t) = \frac{1}{r_{12}(s)}(a_4(s,t) - a_4(s,B')).$$

The fact that both $\alpha(s,t)$, $\beta(s,t) \in \mathbb{K}[s,t]$ can be deduced straightforwardly from (5) because

$$t - B' = t - R_{13}(s)r_{13}(s)t = (1 - R_{13}(s)r_{13}(s))t = R_{12}(s)r_{12}(s)t,$$

which implies that $a_i(s,t) - a_i(s,B')$ is a multiple of $r_{12}(s)$ for all $i = 1, 2, 3, 4$.

Estimating degrees, we get

$$\boxed{\deg(\alpha) \leq d\deg(B') = 2d^3, \quad \deg(\beta) \leq d\deg(B') = 2d^3}$$

and therefore

$$\boxed{\deg(E_1) \leq 2d^3 + d^2 - d}.$$

The matrix $S_1 \in \mathbb{K}[s,t]^{4\times 4}$ is of the form

$$S_1 = \begin{pmatrix} S_{11} & 0 \\ 0 & I_2 \end{pmatrix},$$

with

$$S_{11} = \begin{pmatrix} \frac{A_1(s,t)a_1(s,B')+A_2(s,B')a_2(s,t)}{r_{12}(s)} & \frac{A_1(s,t)a_2(s,B')-A_1(s,B')a_2(s,t)}{r_{12}(s)} \\ \frac{A_2(s,t)a_1(s,B')-A_2(s,B')a_1(s,t)}{r_{12}(s)} & \frac{A_2(s,t)a_2(s,B')+A_1(s,B')a_1(s,t)}{r_{12}(s)} \end{pmatrix}.$$

Again the fact that the entries of S_{11} are polynomials can be deduced from (5). We compute

$$\boxed{\begin{aligned}\deg(S_1) &\leq \max\{d^2 - d + d\deg(B'), (d^2 - d)\deg(B') + d\} \\ &= \\ &2d^4 - 2d^3 + d\end{aligned}}$$

Finally, we have

$$N_1 = E_1S_1 = \begin{pmatrix} S_{11} & E_{12} \\ 0 & I_2 \end{pmatrix},$$

and hence

$$\boxed{\deg(N_1) \leq 2d^4 - 2d^3 + d}$$

Now we pass to step 2 of the algorithm, where we compute a unimodular matrix $N_2 \in \mathbb{K}[s,t]^{4\times 4}$ such that

$$P(s,B')N_2 = P(s,0).$$

As before, this matrix is obtained as a product of two unimodular matrices $N_2 = E_2S_2$, where $E_2 \in \mathbb{K}[s,t]^{4\times 4}$ satisfies

$$P(s,B')E_2 = (a_1(s,B'), a_2(s,0), a_3(s,B'), a_4(s,0)),$$

and $S_2 \in \mathbb{K}[s,t]^{4\times 4}$ is such that

$$(a_1(s,B'), a_2(s,0), a_3(s,B'), a_4(s,0))S_2 = P(s,0).$$

To be more precise, we have

$$E_2 = \begin{pmatrix} 1 & -\tilde{\alpha}(s,t)A_1^*(s,B') & 0 & -\tilde{\beta}(s,t)A_1^*(s,B') \\ 0 & 1 & 0 & 0 \\ 0 & -\tilde{\alpha}(s,t)A_2^*(s,B') & 1 & -\tilde{\beta}(s,t)A_2^*(s,B') \\ 0 & 0 & 0 & 1 \end{pmatrix},$$

where

$$\tilde{\alpha}(s,t) = \frac{1}{r_{13}(s)}(a_2(s,B') - a_2(s,0))$$

and

$$\tilde{\beta}(s,t) = \frac{1}{r_{13}(s)}(a_4(s,B') - a_4(s,0)),$$

Computing the degrees, we get

$$\boxed{\deg(\tilde{\alpha}(s,t)),\ \deg(\tilde{\beta}(s,t)) \le d\deg(B') = 2d^3}$$

and therefore

$$\boxed{\deg(E_2) \le (2d^4 - 2d^3) + 2d^3 = 2d^4}.$$

The other matrix is

$$S_2 = \begin{pmatrix} \frac{A_1^*(s,B')a_1(s,0)+A_2^*(s,0)a_3(s,B')}{r_{13}(s)} & 0 & \frac{A_1^*(s,B')a_3(s,0)-A_1^*(s,0)a_3(s,B')}{r_{13}(s)} & 0 \\ 0 & 1 & 0 & 0 \\ \frac{A_2^*(s,B')a_1(s,0)-A_2^*(s,0)a_1(s,B')}{r_{13}(s)} & 0 & \frac{A_2^*(s,B')a_3(s,0)+A_1^*(s,0)a_1(s,B')}{r_{13}(s)} & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

with $\deg(S_2) \le 2d^4 + d$. So, we have that $N_2 = E_2S_2$, and as before the coefficients do not get mixed in the product, so we have

$$\boxed{\deg(N_2) \le 2d^4 + d.}$$

Therefore, we have that

$$P(s,t)N_1N_2 = P(s,0)$$

with degree

$$\boxed{\deg(N_1N_2) \le 4d^4 - 2d^3 + 2d}.$$

Note that $P(s,0) \in \mathbb{K}[s]^4$ is also a unimodular matrix of degree d. It is known that (see for instance [10]) there exists a unimodular matrix $M \in \mathbb{K}[s]^{4\times4}$ with $\deg(M) \le d$ such that

$$P(s,t)N_1N_2M = P(s,0)M = (1,0,0,0).$$

As a consequence of Corollary 2.2, we get that the last three columns of N_1N_2M are a $\mu-$basis of $Syz(P)$ of degree is bounded by

$$\boxed{\deg(N_1N_2M) \le 4d^4 - 2d^3 + 3d \le 4d^4.}$$

□

Example 2.5. Let us consider

$$P(s,t) = (s^2, t^2, s^2 - 1, s^2 + 1).$$

Applying the constructive proof of Theorem 2.4 we have that $B' = t - s^4t$. As the last two polynomials do not depend on t, we get that $E_1 = I_4$,

$$S_1 = \begin{pmatrix} 1 & s^2(-2+s^4)t^2 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix},$$

and hence

$$N_1 = E_1S_1 = \begin{pmatrix} 1 & -2s^2t^2 + s^6t^2 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}.$$

Similarly, we compute E_2 and S_2 in the second step of the algorithm to get the matrix

$$N_2 = E_2S_2 = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & t^2 + s^2t^2 - s^4t^2 - s^6t^2 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}.$$

We also compute

$$M = \begin{pmatrix} -1 & 0 & -2 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & s^2+1 \\ 1 & 0 & 1 & -s^2+1 \end{pmatrix},$$

so we have that

$$N_1N_2M = \begin{pmatrix} -1 & -2s^2t^2 + s^6t^2 & -2 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & t^2 + s^2t^2 - s^4t^2 - s^6t^2 & 1 & 1+s^2 \\ 1 & 0 & 1 & 1-s^2 \end{pmatrix}$$

and a $\mu-$basis $\{p,\ q,\ r\}$ of $P(s,t)$ is given by the last three columns of the above matrix, that is
$p_1(s,t) = (-2s^2t^2 + s^6t^2, 1, t^2 + s^2t^2 - s^4t^2 - s^6t^2, 0)$,
$p_2(s,t) = (-2, 0, 1, 1)$, and $p_3(s,t) = (0, 0, 1+s^2, 1-s^2)$.

Example 2.6. Let us consider

$$P(s,t) = (2st, 2t, 2s, s^2 + t^2 + 1).$$

Note that in this case $V(a_1, a_2, a_3) \neq \emptyset$, but we can resort the sequence and get

$$P(s,t) = (s^2 + t^2 + 1, 2t, 2s, 2st).$$

which suits better to our computations. Now $B' = -s^2t$ and we compute

$$N_1 = \begin{pmatrix} s^2t^2+1 & -2t & 0 & -2st \\ -\frac{1}{2}t\left(s^2t^2+s^2+1\right) & t^2+1 & 0 & st^2 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix},$$

$$N_2 = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ -\frac{1}{2}s^3t^2 & st & 1 & s^2t \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

and

$$M = \begin{pmatrix} 1 & 0 & 0 & -2s \\ 0 & 1 & 0 & 0 \\ -s/2 & 0 & 0 & s^2+1 \\ 0 & 0 & 1 & 0 \end{pmatrix}.$$

Therefore the three last columns of

$$N_1N_2M = \begin{pmatrix} s^2t^2+1 & -2t & -2st & -2s^3t^2 - 2s \\ -\frac{1}{2}s^2t^3 - \frac{s^2t}{2} - \frac{t}{2} & t^2+1 & st^2 & s^3t^3 + s^3t + st \\ -\frac{1}{2}s^3t^2 - \frac{s}{2} & st & s^2t & s^4t^2 + s^2 + 1 \\ 0 & 0 & 1 & 0 \end{pmatrix},$$

are a μ−basis of $P(s,t)$, i.e. we have in this case
$p_1(s,t) = (-2t, t^2+1, st, 0)$, $p_2(s,t) = (-2st, st^2, st, 0)$, and
$p_3(s,t) = (-2s^3t^2 - 2s, s^3t^3 + s^3t + st, s^4t^2 + s^2 + 1, 0)$.

3 PROOF OF THEOREM 1.2

Now we will consider the case when I_P has -maybe after a linear change of coordinates- a shape basis, i.e.

$$\langle a_1(s,t), a_2(s,t), a_3(s,t), a_4(s,t)\rangle = \langle p(s), t - q(s)\rangle \tag{6}$$

with

$$\deg(p(s)) = D, \quad \deg(q(s)) \le D-1.$$

By Bézout's Theorem, we have straightforwardly that $D \le d^2$. This setup contains the case when I_P is a radical ideal.

From (6) we get that there exist $A_i(s,t), B_i(s)$, $i = 1,2,3,4$ such that

$$a_i(s,t) = A_i(s,t)(t - q(s)) + B_i(s)p(s). \tag{7}$$

We can compute explicitly these polynomials and bound their degrees as follows. Set $D^* := \max(1, D-1)$, so that $\deg(t-q(s)) \le D^*$. By applying the Division Algorithm between $a_i(s,t)$ and $t - q(s)$, we have

$$a_i(s,t) = A_i(s,t)(t - q(s)) + r_i(s)$$

with $r_i(s) = a_i(s, q(s))$, and then

$$\deg(r_i(s)) = \deg(a_i(s,q(s))) \le dD^*.$$

Moreover, from (7) we deduce that $r_i(s) = p(s)B_i(s)$. Therefore,

$$\deg(B_i(s)) \le \deg(r_i(s)) - \deg(p(s)) = dD^* - D \le D^*(d-1),$$

and $\deg(A_i(s,t)) \le dD^*$.

Then, we can write $P(s,t)$ as

$$(t - q(s) \;\; p(s)) \cdot \begin{pmatrix} A_1(s,t) & A_2(s,t) & A_3(s,t) & A_4(s,t) \\ B_1(s) & B_2(s) & B_3(s) & B_4(s) \end{pmatrix}$$

with

$$\boxed{\deg(A_i, B_i) \le dD^*.}$$

From (6) we have also that

$$P(s,t) \cdot \begin{pmatrix} \alpha_1(s,t) & \beta_1(s,t) \\ \alpha_2(s,t) & \beta_2(s,t) \\ \alpha_3(s,t) & \beta_3(s,t) \\ \alpha_4(s,t) & \beta_4(s,t) \end{pmatrix} = (t - q(s) \;\; p(s))$$

for suitable polynomials $\alpha_i(s,t), \beta_i(s,t) \in \mathbb{K}[s,t]$, $i = 1,2,3,4$.

So, we get

$$t - q(s) = \sum_{i=1}^{4} \alpha_i(s,t)a_i(s,t)$$

and

$$p(s) = \sum_{i=1}^{4} \beta_i(s,t)a_i(s,t) =$$

$$= \sum_{i=1}^{4} \beta_i(s,t)(A_i(s,t)(t-q(s)) + B_i(s)p(s)) \Rightarrow$$

$$\Rightarrow \left(1 - \sum_{i=1}^{4} \beta_i(s,t)B_i(s)\right)p(s) = \sum_{i=1}^{4} \beta_i(s,t)A_i(s,t)(t-q(s)) \Rightarrow$$

$$\Rightarrow \begin{cases} 1 - \sum_{i=1}^{4} \beta_i(s,t)B_i(s) = A(s,t)(t-q(s)) \\ \sum_{i=1}^{4} \beta_i(s,t)A_i(s,t) = A(s,t)p(s) \end{cases} \tag{8}$$

for a suitable $A(s,t) \in \mathbb{K}[s,t]$. We set $t = q(s)$ in the first equation of (8) and get

$$\sum_{i=1}^{4} \beta_i(s,q(s))B_i(s) = 1,$$

that is $\gcd(B_1(s), B_2(s), B_3(s), B_4(s)) = 1$ and by the results of [10] we can compute a unimodular matrix $M_B \in \mathbb{K}[s]^{4\times 4}$ such that

$$\begin{pmatrix} A_1(s,t) & A_2(s,t) & A_3(s,t) & A_4(s,t) \\ B_1(s) & B_2(s) & B_3(s) & B_4(s) \end{pmatrix} \cdot M_B = \begin{pmatrix} \tilde{A}_1(s,t) & \tilde{A}_2(s,t) & \tilde{A}_3(s,t) & \tilde{A}_4(s,t) \\ 1 & 0 & 0 & 0 \end{pmatrix}, \tag{9}$$

with

$$\boxed{\deg(M_B) \le D^*(d-1),\ \deg(\tilde{A}_i) \le 2dD^*.}$$

Set now

$$(\tilde{a}_1(s,t), \tilde{a}_2(s,t), \tilde{a}_3(s,t), \tilde{a}_4(s,t)) := P(s,t) \cdot M_B. \tag{10}$$

From (9), we have that I_P is actually equal to the ideal generated by

$$\tilde{A}_1(s,t)(t-q(s)) + p(s), \tilde{A}_2(s,t)(t-q(s)), \tilde{A}_3(s,t)(t-q(s)), \tilde{A}_4(s,t)(t-q(s)),$$

and therefore, there exist $\gamma_i(s,t) \in \mathbb{K}[s,t]$, $i = 1,2,3,4$, such that

$$(t-q(s)) = \gamma_1(s,t)(\tilde{A}_1(s,t)(t-q(s))+p(s))+\sum_{i=2}^{4} \gamma_i(s,t)\tilde{A}_i(s,t)(t-q(s)).$$

Replacing $t = q(s)$ in the above equation, we obtain that $\gamma_1(s,q(s)) = 0$ and therefore we have $\gamma_1(s,t) = \tilde{\gamma}_1(s,t)(t-q(s))$, so

$$1 = \tilde{\gamma}_1(s,t)(\tilde{A}_1(s,t)(t-q(s)) + p(s)) + \sum_{i=2}^{4} \gamma_i(s,t)\tilde{A}_i(s,t),$$

that is, $\langle(\tilde{A}_1(s,t)(t-q(s))+p(s), \tilde{A}_2(s,t), \tilde{A}_3(s,t), \tilde{A}_4(s,t))\rangle = \mathbb{K}[s,t]$. So, we can apply Theorem 2.1 to

$$\tilde{A} = (\tilde{A}_1(s,t)(t-q(s)) + p(s), \tilde{A}_2(s,t), \tilde{A}_3(s,t), \tilde{A}_4(s,t)), \tag{11}$$

which has $\deg(\tilde{A}) \le 2d(D-1) - 1$, to get a unimodular matrix $M_{\tilde{A}} \in \mathbb{K}[s,t]^{4\times 4}$ such that

$$(\tilde{A}_1(s,t)(t-q(s)) + p(s), \tilde{A}_2(s,t), \tilde{A}_3(s,t), \tilde{A}_4(s,t))M_{\tilde{A}} = (1,0,0,0) \tag{12}$$

with

$$\boxed{\deg(M_{\tilde{A}}) \le 4(2dD^* + D^*)^4 = 4D^{*4}(2d+1)^4.}$$

If we denote with $M^2_{\tilde{A}}, M^3_{\tilde{A}}, M^4_{\tilde{A}}$ the three last columns of $M_{\tilde{A}}$, then by the Hilbert-Burch Theorem (Theorem 1.1) we have that -up to a nonzero constant in $\mathbb{K}$-

$$M^2_{\tilde{A}} \wedge M^3_{\tilde{A}} \wedge M^4_{\tilde{A}} = (\tilde{A}_1(s,t)(t-q(s))+p(s), \tilde{A}_2(s,t), \tilde{A}_3(s,t), \tilde{A}_4(s,t)). \tag{13}$$

Write $M_{\tilde{A}} = \left(m_{ij}\right)_{1\le i,j\le 4}$, and set

$$M_{\tilde{P}} = \begin{pmatrix} m_{12}(t-q(s)) & m_{13}(t-q(s)) & m_{14}(t-q(s)) \\ m_{22} & m_{23} & m_{24} \\ m_{32} & m_{33} & m_{34} \\ m_{42} & m_{43} & m_{44} \end{pmatrix}. \tag{14}$$

We clearly have

$$\boxed{\deg(M_{\tilde{P}}) \le 4D^{*4}(2d+1)^4 + D^*}.$$

and from (13) we deduce that $M^2_{\tilde{P}} \wedge M^3_{\tilde{P}} \wedge M^4_{\tilde{P}}$ is the vector with coordinates $\tilde{A}_1(s,t)(t-q(s))+p(s)$, $(t-q(s))\tilde{A}_2(s,t)$, $(t-q(s))\tilde{A}_3(s,t)$, and $(t-q(s))\tilde{A}_4(s,t)$).

So, by the converse of the Hilbert-Burch Theorem (Theorem 1.1) we deduce that $\{M^2_{\tilde{P}}, M^3_{\tilde{P}}, M^4_{\tilde{P}}\}$ is a μ-basis of the parametrization given by these four components.

To conclude, we set $\tilde{M} = M_B M_{\tilde{P}} \in \mathbb{K}[s,t]^{4\times 3}$. From (9) and (10) we deduce that

$$P(s,t)\cdot \tilde{M} = (0,0,0). \tag{15}$$

As M_B is unimodular, and the columns of $M_{\tilde{P}}$ a μ-basis of the aforementioned parametrization, we deduce straightforwardly that the columns of $\tilde{M}$ are a μ–basis of $P(s,t)$. Computing it straightforwardly we get

$$\boxed{\deg(\tilde{M}) \le D^*(d-1) + 4D^{*4}(2d+1)^4 + D^* \le 5D^{*4}(2d+1)^4,}$$

which concludes with the proof of the Theorem.

4 ALGORITHM AND RUNNING EXAMPLE

We collect here all the steps needed to compute a μ-basis of a given parametrization $P(s,t)$.

Algorithm 4.1.
Input: A parametrization $P(s,t) := (a_1(s,t),\ a_2(s,t),\ a_3(s,t),\ a_4(s,t))$ of polynomials in $\mathbb{K}[s,t]^4$ such that $\gcd(a_i(s,t)) = 1$ and that the ideal I_P they generate is radical.
Output: A matrix $\tilde{M} = \left(m_{ij}\right)_{1\le i\le 4, 1\le j\le 3}$ from (15) such that its three columns are a μ-basis of $P(s,t)$.
Procedure:

1. Compute $p(s),\ q(s) \in \mathbb{K}[s]$ such that $I_P = \langle p(s),\ t-q(s)\rangle$. Some linear change of variables may be needed at this step.
2. For $i = 1,2,3,4$, compute $A_i(s,t),\ B_i(s)$ as in (7).
3. Compute the unimodular matrix M_B from (9). Note that at these steps the polynomials $\tilde{A}_i(s,t)$ are defined, $1 \le i \le 4$.
4. By applying the effective Quillen-Suslin (Theorem 2.1), compute the matrix $M_{\tilde{A}}$ from (12).
5. By using (14), compute $M_{\tilde{P}}$.
6. Set $\tilde{M} := M_B M_{\tilde{P}} \in \mathbb{K}[s,t]^{4\times 3}$.

We conclude the paper by illustrating the algorithm with a concrete case.

Example 4.1. Consider the following parametrization:

$$\begin{cases} a_1(s,t) &= 11 - 4s + 3s^2 + 4t \\ a_2(s,t) &= 5 - 4s + 2s^2 + 4t - 2st + t^2 \\ a_3(s,t) &= 1 + 3s^2 - s^3 + s^2 t \\ a_4(s,t) &= 7 - 3s + s^2 + 3t. \end{cases}$$

(1) By computing a Gröbner Basis of these polynomials with respect to lex $t > s$, we get that $I_P = \langle 2 - s + t,\ 1 + s^2\rangle$, so we have $d = 3,\ D = 2$ in this case, and $p(s) = 1+s^2,\ q(s) = -s+2$.
(2) We compute explicitly $A_i(s,t),\ B_i(s),\ 1 \le i \le 4$, as in (7) to get

$$\begin{pmatrix} A_1(s,t) & B_1(s) \\ A_2(s,t) & B_2(s) \\ A_3(s,t) & B_3(s) \\ A_4(s,t) & B_4(s) \end{pmatrix} = \begin{pmatrix} 4 & 3 \\ 2-s+t & 1 \\ s^2 & 1 \\ 3 & 1 \end{pmatrix}$$

(3) A unimodular matrix M_B as in (9) can be the following:

$$M_B = \begin{pmatrix} 0 & 1 & 1 & 1 \\ 1 & -3 & 0 & 0 \\ 0 & 0 & -3 & 0 \\ 0 & 0 & 0 & -3 \end{pmatrix}.$$

With this choice of M_B, we have

$$\left(\tilde{A}_1(s,t), \tilde{A}_2(s,t), \tilde{A}_3(s,t), \tilde{A}_4(s,t)\right) = (A_1(s,t), A_2(s,t), A_3(s,t), A_4(s,t))\cdot M_B = \left(2-s+t, -2+3s-3t, 4-3s^2, -5\right),$$

so $\tilde{A}$ from (11) is equal to

$$(5 - 4s + 2s^2 + 4t - 2st + t^2, -2 + 3s - 3t, 4 - 3s^2, -5),$$

which we can confirm (by computing Gröbner bases for instance) that it is a unimodular matrix.
(4) The matrix $M_{\tilde{A}}$ from (12) is equal to

$$\begin{pmatrix} 0 & 5 - \frac{150t}{37} + \frac{45st}{37} - \frac{540t^2}{1369} + \frac{405s^2t^2}{1369} & \frac{135t}{37} & 0 \\ 0 & m_1 & 5 + \frac{150t}{37} - \frac{45st}{37} + \frac{45t^2}{37} & 0 \\ 0 & -\frac{60t}{37} + \frac{30st}{37} - \frac{180t^2}{1369} + \frac{135s^2t^2}{1369} & \frac{45t}{37} & 5 \\ -\frac{1}{5} & 5 - 4s + 2s^2 & -2 + 3s & 4 - 3s^2 \end{pmatrix}$$

with

$$m_1 = -\frac{125t}{37} + \frac{100st}{37} - \frac{2450t^2}{1369} + \frac{735st^2}{1369} + \frac{450s^2t^2}{1369} - \frac{135s^3t^2}{1369} - \frac{180t^3}{1369} + \frac{135s^2t^3}{1369}$$

(5) $M_{\tilde{P}}$ is deduced easily from $M_{\tilde{A}}$, as shown in (14).
(6) By computing $M_B M_{\tilde{P}}$ we get $\tilde{M} = \left(m_{ij}\right)_{1\le i\le 4, 1\le j\le 3}$, with

$$\begin{aligned} m_{11} &= 5 - 4s + 2s^2 - 5t + \frac{130st}{37} - \frac{2630t^2}{1369} + \frac{735st^2}{1369} + \\ &\quad + \frac{585s^2t^2}{1369} - \frac{135s^3t^2}{1369} - \frac{180t^3}{1369} + \frac{135s^2t^3}{1369}, \\ m_{21} &= 10 - 5s + \frac{260t}{37} - \frac{60st}{37} - \frac{45s^2t}{37} + \frac{720t^2}{1369} - \frac{540s^2t^2}{13} \\ m_{31} &= \frac{180t}{37} - \frac{90st}{37} + \frac{540t^2}{1369} - \frac{405s^2t^2}{1369}, \\ m_{41} &= -15 + 12s - 6s^2, \end{aligned}$$

$$\begin{aligned} m_{12} &= 3 + 3s + \frac{195t}{37} - \frac{45st}{37} + \frac{45t^2}{37}, \\ m_{22} &= -15 - \frac{180t}{37} \\ m_{32} &= -\frac{135t}{37}, \\ m_{42} &= 6 - 9s, \end{aligned}$$

$$\begin{aligned} m_{13} &= 9 - 3s^2, \\ m_{23} &= 0, \\ m_{33} &= -15, \\ m_{43} &= -12 + 9s^2. \end{aligned}$$

ACKNOWLEDGMENTS

All our computations were done with the aid of `Macaulay2` ([9]) and `Mathematica` ([11]). T. Cortadellas is supported by the Spanish MEC research project MTM2013-40775-P, C. D'Andrea and E. Montoro are supported by the Spanish MINECO/FEDER research project MTM 2015-65361-P. C. D'Andrea is also supported by the European Union's Horizon 2020 research and innovation programme under the Marie Skłodowska-Curie grant agreement "GRAPES" No. 860843.

REFERENCES

[1] Eberhard Becker, Maria Marinari, Teo Mora, and Carlo Traverso. 1994. The shape of the Shape Lemma. *ISSAC'94: Proceedings of the International Symposium on Symbolic and Algebraic Computation* (08 1994), 129–133. DOI: http://dx.doi.org/https://doi.org/10.1145/190347.190382

[2] Falai Chen, David Cox, and Yang Liu. 2005. The μ-basis and implicitization of a rational parametric surface. *Journal of Symbolic Computation* 39, 6 (2005), 689 – 706. DOI: http://dx.doi.org/https://doi.org/10.1016/j.jsc.2005.01.003

[3] Falai Chen and Wenping Wang. 2002. The μ-basis of a planar rational curve - properties and computation. *Graph. Model.* 64, 6 (2002), 368–381. DOI: http://dx.doi.org/10.1016/S1077-3169(02)00017-5

[4] Yairon Cid-Ruiz. 2019. Bounding the degrees of a minimal μ-basis for a rational surface parametrization. *Journal of Symbolic Computation* 95 (2019), 134 – 150. DOI: http://dx.doi.org/https://doi.org/10.1016/j.jsc.2019.02.003

[5] David A. Cox, Thomas W. Sederberg, and Falai Chen. 1998. The moving line ideal basis of planar rational curves. *Computer Aided Geometric Design* 15, 8 (1998), 803 – 827. DOI: http://dx.doi.org/https://doi.org/10.1016/S0167-8396(98)00014-4

[6] Jiansong Deng, Falai Chen, and Li-Yong Shen. 2005. Computing μ-bases of rational curves and surfaces using polynomial matrix factorization. In *ISSAC'05*. ACM, New York, 132–139. DOI: http://dx.doi.org/10.1145/1073884.1073904

[7] David Eisenbud. 2005. *The geometry of syzygies*. Graduate Texts in Mathematics, Vol. 229. Springer-Verlag, New York. A second course in commutative algebra and algebraic geometry.

[8] Noaï Fitchas and André Galligo. 1990. Nullstellensatz effectif et Conjecture de Serre (Théorème de Quillen-Suslin) pour le Calcul Formel. *Mathematische Nachrichten* 149, 1 (1990), 231–253.

[9] Daniel R. Grayson and Michael E. Stillman. 2018. Macaulay2, a software system for research in algebraic geometry. (2018). Available at http://www.math.uiuc.edu/Macaulay2/.

[10] Hoon Hong, Zachary Hough, and Irina A. Kogan. 2017. Algorithms for computing μ-bases of univariate polynomials. *Journal of Symbolic Computation* 80 (2017), 844 – 874. DOI: http://dx.doi.org/https://doi.org/10.1016/j.jsc.2016.08.013

[11] Wolfram Research, Inc. 2018. Mathematica, Version 11. (2018). Champaign, IL.

[12] Ning Song and Ron Goldman. 2009. μ-bases for polynomial systems in one variable. *Comput. Aided Geom. Design* 26, 2 (2009), 217–230. DOI: http://dx.doi.org/10.1016/j.cagd.2008.04.001

On A Non-Archimedean Broyden Method

Xavier Dahan
Tohoku University, IEHE
Sendai, Japan
xdahan@gmail.com

Tristan Vaccon
Université de Limoges; CNRS, XLIM UMR 7252
Limoges, France
tristan.vaccon@unilim.fr

ABSTRACT

Newton's method is an ubiquitous tool to solve equations, both in the archimedean and non-archimedean settings — for which it does not really differ. Broyden was the instigator of what is called "quasi-Newton methods". These methods use an iteration step where one does not need to compute a complete Jacobian matrix nor its inverse. We provide an adaptation of Broyden's method in a general non-archimedean setting, compatible with the lack of inner product, and study its Q and R convergence. We prove that our adapted method converges at least Q-linearly and R-superlinearly with R-order $2^{\frac{1}{2m}}$ in dimension m. Numerical data are provided.

CCS CONCEPTS

• **Computing methodologies** → *Exact arithmetic algorithms.*

KEYWORDS

System of equations, Broyden's method, Quasi-Newton, p-adic approximation, Power series, Symbolic-numeric, p-adic algorithm

ACM Reference Format:
Xavier Dahan and Tristan Vaccon. 2020. On A Non-Archimedean Broyden Method. In *International Symposium on Symbolic and Algebraic Computation (ISSAC '20), July 20–23, 2020, Kalamata, Greece*. ACM, New York, NY, USA, 9 pages. https://doi.org/10.1145/3373207.3404045

1 INTRODUCTION

In the numerical world. Quasi-Newton methods refer to a class of variants of Newton's method for solving square nonlinear systems, with the twist that the inverse of the Jacobian matrix is "approximated" by another matrix. When compared to Newton's method, they benefit from a cheaper update at each iteration (See *e.g.* [10, p.49-50, 53]), but suffer from a smaller rate of convergence. They were mainly introduced by Broyden in [6], which has sparked numerous improvements, generalizations, and variants (see the surveys [10, 19]). It is now a fundamental numerical tool (that finds its way in entry level numerical analysis textbooks [8, § 10.3]). To some extent, this success stems from: the specificities of machine precision arithmetic as commonly used in the numerical community, the fact that Newton's method is usually not quadratically convergent from step one, and that the arithmetic cost of an iteration is independent of the quality of the approximation reached. In another direction, variants of Broyden's method have known dramatic success for unconstrained optimization — the target system is the gradient of the objective function, the zeros are then critical points— where it takes advantage of the special structure of the Hessian (see Sec. 7 of [10]). Another appealing feature of Broyden's method is the possibility to design derivative-free methods generalizing to the multivariate case the classical secant method (which can be thought of as Broyden's in dimension one). This feature is a main motivation for this work.

Non-archimedean. It is a natural wish to transpose such a fundamental numerical method to the non-archimedean framework, offering new tools to perform *exact* computations, typically for systems with p-adic or power series coefficients. For this adaptation, several non-trivial difficulties have to be overcome: *e.g.* no inner products, a more difficult proof of convergence, or a management of arithmetic at finite precision far more subtle. This article presents satisfactory solutions for all these difficulties, which we believe can be expanded to a broader variety of quasi-Newton methods.

Bach proved in [1] that in dimension one, the secant method can be on an equal footing with Newton's method in terms of complexity. We investigate how this comparison is less engaging in superior dimension (see Section 6). To our opinion, this is due to the remarkable behavior of Newton's method in the non-archimedean setting. No inversion of the Jacobian is required at each iteration (simply a matrix multiplication, this is now classical see [5, 16, 17]). The evaluation of the Jacobian is also efficient for polynomial functions (in dimension m, it involves only $O(m)$ evaluations, instead of m^2 over $\mathbb{R}$, see [2]). It displays also true quadratic behavior from step one which, when combined with the natural use of finite precision arithmetic (against machine precision over $\mathbb{R}$), offers a ratio cost/precision gained that is hard to match.

And indeed, our results show that for large dimension m and polynomials as input, there is little hope for Broyden to outperform Newton, although it depends on the order of superlinear convergence of Broyden's method. In this respect more investigation is necessary, but for now the interest lies more in the theoretical advances and in the situations mentioned in "*Motivations*" thereafter.

Relaxed arithmetic. Since the cost of one iteration of Broyden's method involves m^2 instead of m^ω for Newton, we should mention the *relaxed* framework (a.k.a online [11]) which show essentially the same decrease of complexity, while maintaining quadratic convergence. It has been implemented efficiently for power series [23], and for p-adic numbers [3]. In case of a smaller m and a larger precision of approximation required, FFT trading [24] has to be mentioned. These techniques are however unlikely to be suited to the Broyden iteration, since it is *a priori* not described by a fixed-point equation, a necessity for the relaxed machinery.

ISSAC '20, July 20–23, 2020, Kalamata, Greece

ACM ISBN 978-1-4503-7100-1/20/07…$15.00
https://doi.org/10.1145/3373207.3404045

Motivations. As explains Remark 6.4, it seems unlikely in the non-archimedean world that with polynomials or rational fractions, a quasi-Newton method meets the standard of Newton's method. The practical motivations concern:

1/ Derivative-free method: instead of starting with the Jacobian at precision one, use a divided-difference matrix. A typical application is when the function is given by a "black-box" and there is no direct access to the Jacobian.

2/ When computing the Jacobian does not allow shortcuts like in the case of rational fractions [2], evaluating it may require up to Lm^2 operations, where L is the complexity of evaluation of the input function. Regarding the complexity of Remark 6.4, Broyden's method then becomes beneficial when $L \gtrsim m^2 - m^{\omega-1}$.

3/ While Newton's method over general Banach spaces of infinite dimension can be made effective when the differential is effectively representable (integral equations [15, § 5][14] are a typical example), it is in general difficult or impossible to compute it. On the other hand, Broyden's method or its variants have the ability to work with approximations of the differential, including of *finite rank*, by considering a projection (as shown in [14, 15] and the references therein; the dimension of the projection is increased at each iteration). In the non-archimedean context, ODEs with parameters, for example initial conditions, constitute a natural application.

Organization of the paper. Definitions and notations are introduced in Section 2. Section 3 explains how Broyden's method can be adapted to an ultrametric setting. In Section 4, we study the Q and R-order of convergence of Broyden's method (see Definition 2.1), presenting our main results. It is followed by Section 5, where are introduced developments and conjectures on Q-superlinearity. Finally, in Section 6, we explain how our Broyden's method can be implemented with dynamical handling of the precision, and we conclude with some numerical data in Section 7.

2 BROYDEN'S METHOD AND NOTATIONS

2.1 General notations

Throughout the paper, K refers to a complete, discrete valuation field, $\mathrm{val} : K \twoheadrightarrow \mathbb{Z} \cup \{+\infty\}$ to its valuation, O_K its ring of integers and π a uniformizer.[1] For $k \in \mathbb{N}$, we write $O(\pi^k)$ for $\pi^k O_K$.

Let $m \in \mathbb{Z}_{\geq 1}$. We are interested in computing an approximation of a non-singular zero $x^\star$ of $f : K^m \to K^m$ through an iterative sequence of approximations, $(x_n)_{n\in\mathbb{N}} \in (K^m)^{\mathbb{N}}$. Note that all our vectors are column-vectors. For any $x \in K^m$ where it is well-defined, we denote by $f'(x) \in M_m(K)$ the Jacobian matrix of f at x. We will use the following notations (borrowed from [13]):

$$f_n = f(x_n), \quad y_n = f_{n+1} - f_n, \quad s_n = x_{n+1} - x_n \tag{1}$$

We denote by $(e_1, \ldots, e_m)$ the canonical basis of K^m. In K^m, $O(\pi^k)$ means $O(\pi^k)e_1 + \cdots + O(\pi^k)e_m$.

Newton's iteration produces a sequence $(x_n)_{n\in\mathbb{N}}$ given by:

$$x_{n+1} = x_n - f'(x_n)^{-1} \cdot f(x_n). \tag{N}$$

For quasi-Newton methods, the iteration is given by:

$$x_{n+1} = x_n - B_n^{-1} \cdot f(x_n), \quad (\Rightarrow s_n = -B_n^{-1} \cdot f_n) \tag{QN}$$

with B_n presumably not far from $f'(x_n)$. More precisely, it is a generalization of the design of the secant method over K where one approximates $f'(x_n)$ by $\frac{f(x_n)-f(x_{n-1})}{x_n-x_{n-1}}$. In quasi-Newton, it is thus required that:

$$B_n \cdot (x_n - x_{n-1}) = f(x_n) - f(x_{n-1}) \quad (\Rightarrow B_n \cdot s_{n-1} = y_{n-1}) \tag{2}$$

By this condition alone, B_n is obviously underdetermined. To mitigate this issue, B_n is taken as a one-dimensional modification of B_{n-1} satisfying (2). Concretely, a sequence $(u_n)_{n\in\mathbb{N}} \in (K^m)^{\mathbb{N}}$ is introduced such that:

$$B_n = B_{n-1} + (y_{n-1} - B_{n-1}s_{n-1}) \cdot {u_{n-1}}^t. \tag{3}$$

$$1 = {u_{n-1}}^t \cdot s_{n-1}. \tag{4}$$

In Broyden's method over $\mathbb{R}$, u_{n-1} is defined by:

$$u_{n-1} = \frac{s_{n-1}}{{s_{n-1}}^t \cdot s_{n-1}}. \tag{5}$$

The computation of the inverse of B_n can then be done using the Sherman-Morrison formula (see [22]):

$$B_n^{-1} = B_{n-1}^{-1} + \frac{(s_{n-1} - B_{n-1}^{-1}y_{n-1}) \cdot {s_{n-1}}^t B_{n-1}^{-1}}{{s_{n-1}}^t B_{n-1}^{-1} y_{n-1}}. \tag{6}$$

This formula gives rise to the so-called "good Broyden's method". Using [22] provides the following alternative formulae:

$$B_n = B_{n-1} + f_n \cdot {u_{n-1}}^t, \quad B_n^{-1} = B_{n-1}^{-1} - \frac{B_{n-1}^{-1} f_n \cdot {u_{n-1}}^t B_{n-1}^{-1}}{{u_{n-1}}^t B_{n-1}^{-1} y_{n-1}}. \tag{7}$$

2.2 Convergence

We recall some notions on convergence of sequences commonly used in the analysis of the behavior of Broyden's method.

Definition 2.1 ([20] Chapter 9). *A sequence $(x_k)_{k\in\mathbb{N}} \in (K^m)^{\mathbb{N}}$ has Q-order of convergence $\mu \in \mathbb{R}_{>1}$ to a limit $x^\star \in K^m$, if:*

$$\exists r \in \mathbb{R}_+, \quad \forall k \text{ large enough}, \quad \frac{\|x_{k+1} - x^\star\|}{\|x_k - x^\star\|^\mu} \leq r.$$

If we can take $\mu = 1$ and $r < 1$ in the previous inequality, we say that $(x_k)_{k\in\mathbb{N}}$ has Q-linear convergence. For $\mu = 2$, we say it has Q-quadratic convergence. The sequence is said to have Q-superlinear convergence if

$$\lim_{k\to+\infty} \frac{\|x_{k+1} - x^\star\|}{\|x_k - x^\star\|} = 0.$$

It is said to have R-order of convergence[2] $\mu \in \mathbb{R}_{\geq 1}$ if

$$\limsup \|x_k - x^\star\|^{1/\mu^k} < 1.$$

Remark 2.2. For both Q and R, we write *has convergence μ* to mean *has convergence at least μ*.

Broyden's method satisfies the following convergence results:

Theorem 2.3. *Over $\mathbb{R}^m$, under usual regularity assumptions, Broyden's method defined by Eq.* (5) *converges locally[3] Q-superlinearly [7], exactly in $2m$ steps for linear systems, and with R-order at least $2^{\frac{1}{2m}} > 1$ [13].*

[1] Discrete valuation is only needed in Section 6. For the rest complete and ultrametric is enough.

[2] R-convergence is a weaker notion, aimed at sequences not monotonically decreasing.

[3] By locally, we mean that for any x_0 and B_0 in small enough balls around $x^\star$ and $f'(x^\star)$, the following convergence property is satisfied.

Unfortunately, for general K, Eq. (5) is not a good fit. Indeed, the quadratic form $x \mapsto x^t x$ can be isotropic over K^m, *i.e.* there can be an $s_n \neq 0$ such that ${s_n}^t \cdot s_n = 0$. This is the case, for example if $s_n = (X, X)$ in $\mathbb{F}_2[\![X]\!]^2$. Consequently, (5) has to be modified. Trying to seek for another quadratic form that would not be isotropic is pointless, since for example there is none over $\mathbb{Q}_p^m$ for $m \geq 5$ [21].

Remark 2.4. In the sequel, all the B_i's will be invertible matrices. Consequently, $s_{n+1} = 0$ if and only if $f(x_n) = 0$. We therefore adopt the convention that if for some x_n, we have $f(x_n) = 0$, then the sequences $(x_v)_{v \geq n}$ and $(B_v)_{v \geq n}$ will be constant, and this case does not require any further development.

3 NON-ARCHIMEDEAN ADAPTATION

3.1 Norms

We use the following natural (non-normalized) norm on K defined from its valuation: for any $x \in K$, $\|x\| = 2^{-\mathrm{val}(x)}$, except for $K = \mathbb{Q}_p$, where we take the more natural $p^{-\mathrm{val}(x)}$ over $\mathbb{Q}_p$. Our norm[4] on K can naturally be extended to K^m: for any $x = (x_1, \ldots, x_m) \in K^m$, $\|x\| = \max_i |x_i|$. We denote by $\mathrm{val}(x)$ the minimal valuation among the $\mathrm{val}(x_i)$'s. It defines the norm of x.

Lemma 3.1. *Let $|||\cdot|||$ be the norm on $M_m(K)$ induced by $\|\cdot\|$. Let us abuse notations by denoting with $\|\cdot\|$ the max-norm on the coefficients of the matrices of $M_m(K)$. Then $|||\cdot||| = \|\cdot\|$.*

Proof. Let $A \in M_n(K)$. If $x \in K^m$ is such that $\|x\| \leq 1$, then by ultrametricity, it is clear that $\|Ax\| \leq \|A\|$, hence $|||A||| \leq \|A\|$. If $i \in \mathbb{N}$ is such that $\|A\|$ is obtained with a coefficient on the column of index i, then $\|Ae_i\| = \|A\|$, whence the equality. □

Consequently, the *max*-norm on the coefficients of a matrix is a matrix norm. For rank-one matrices, the computation of the norm can be made easy using the following corollary of Lemma 3.1.

Corollary 3.2. *Let $a, b \in K^m$ be two vectors. Then*

$$\|a^t \cdot b\| = \|a\| \cdot \|b\|. \tag{8}$$

3.2 Constraints and optimality

For the sequence $(x_n)_{n \in \mathbb{N}}$ to be well defined, the sequence $(u_n)_{n \in \mathbb{N}}$ must satisfy Eqs (3)-(4) and also:

$${s_n}^t B_n^{-1} y_n \neq 0, \tag{9}$$

to ensure Eq. (6) makes sense. Many different u_n's can satisfy those conditions. Over $\mathbb{R}$, Broyden's choice of u_n defined by (5) can be characterized by minimizing the Frobenius norm of $B_{n+1} - B_n$. We can proceed similarly over K.

Lemma 3.3. *If B_{n+1} satisfies* (2)*, then:*

$$\|B_{n+1} - B_n\| \geq \frac{\|y_n - B_n s_n\|}{\|s_n\|}. \tag{10}$$

Proof. It is clear as in this case, $(B_{n+1} - B_n)s_n = y_n - B_n s_n$. □

This inequality can become an equality with a suitable choice of u_n as shown in the following lemma.

[4]Over $\mathbb{R}$, it is of course denoted by $\|\cdot\|_\infty$, but when based on a non-archimedean absolute value, this notation is not used since it is implicitly unambiguous: other norms such as the $\|\cdot\|_p$ are mostly useless.

Lemma 3.4. *Let l be such that* $\mathrm{val}(s_{n,l}) = \mathrm{val}(s_n)$*. Then*

$$u_n = s_{n,l}^{-1} e_l$$

satisfies (4) *and reaches the bound in* (10)*.*

Nevertheless, this is not enough to have B_n invertible in general, as we can see from the Sherman-Morrison formula (7):

Lemma 3.5. *B_n defined by Eq.*(3) *is invertible if and only if*

$${u_{n-1}}^t B_{n-1}^{-1} y_{n-1} \neq 0. \tag{11}$$

The next lemma shows how to choose l, up to the condition $(B_{n-1}^{-1} y_{n-1})_l \neq 0$, which actually never occurs after Corollary 4.3.

Lemma 3.6. *Let l be the* smallest *index such that* $\mathrm{val}(s_{n,l}) = \mathrm{val}(s_n)$*. If $\left(B_{n-1}^{-1} y_{n-1}\right)_l \neq 0$, then*

$$u_n = s_{n,l}^{-1} e_l \tag{12}$$

satisfies Eq. (4)*, reaches the bound in Eq.* (10) *and satisfies Eq.*(11)*.*

4 LOCAL CONVERGENCE

4.1 Local Linear convergence

Let E and F be two finite-dimensional normed vector spaces over K We denote by $L(E, F)$ the space of K-linear mappings from E to F.

Definition 4.1. *Let U be an open subset of E. A function $f : U \to F$ is* strictly differentiable *at $x \in U$ if there exists an $f'(x) \in L(E, F)$ satisfying the following property: for all $\varepsilon > 0$, there exists a neighborhood $U_{x,\varepsilon} \subset U$ of x, on which for any y, $z \in U_{x,\epsilon}$:*

$$\|f(z) - f(y) - f'(x) \cdot (z-y)\|_F \leq \varepsilon \cdot \|z-y\|_E. \tag{13}$$

Note that both z and y can vary. This property is natural in the ultrametric context (see 3.1.3 of [9]), as the counterpart of Fréchet differentiability over $\mathbb{R}$ does not provide meaningful local information. Polynomials and converging power series satisfy strict differentiability everywhere they are defined.

We can then adapt Theorem 3.2 of [7] in our ultrametric setting.

Theorem 4.2. *Let $f : K^m \to K^m$ and $x^\star \in U$ be such that f is strictly differentiable at $x^\star$, $f'(x^\star)$ is invertible and $f(x^\star) = 0$. Then any quasi-Newton method whose choice of u_n yields for all n, $\|u_n\| = \|s_n\|^{-1}$ (which includes Broyden's choice of Eq.* (12)*), is locally Q-linearly converging to $x^\star$ with ratio r for any $r \in (0, 1)$.*

Proof. Let $r \in (0, 1)$. Let the constants γ, δ, and λ be satisfying:

$$\gamma \geq \|f'(x^\star)^{-1}\|, \quad 0 < \delta \leq \frac{r}{\gamma(1+r)(3-r)}, \quad 0 < \lambda \leq \delta(1-r). \tag{14}$$

Let $\eta > 0$ be given by the strict differentiability at $x^\star$ and such that on the ball $B(x^\star, \eta)$,

$$\|f(z) - f(y) - f'(x^\star) \cdot (z-y)\| \leq \lambda \cdot \|z-y\|.$$

We restrict further η so as to have: $\eta \leq \delta(1-r)$. Let us assume that

$$\|B_0 - f'(x^\star)\| \leq \delta, \qquad \|x_0 - x^\star\| < \eta.$$

We have from the condition on δ that $\delta\gamma(1+r)(3-r) \leq r$. Since $3 - r > 2$, then $2\delta\gamma(1+r) \leq r$. Consequently,

$$\frac{1}{1 - 2\delta\gamma} \leq 1 + r,$$

the denominator being non zero because $\delta < (2\gamma)^{-1}$.

Since $\|f'(x^\star)^{-1}\| \leq \gamma$ and $\|B_0 - f'(x^\star)\| < 2\delta$, the Banach Perturbation Lemma ([20] page 45) in the Banach algebra $M_m(K)$ implies that B_0 is invertible and:

$$\|B_0^{-1}\| \leq \frac{\gamma}{1-2\gamma\delta} \leq (1+r)\gamma.$$

We can now estimate what happens to $x_1 = x_0 - B_0^{-1}f(x_0)$.

$$\begin{aligned}
\|x_1 - x^\star\| &= \|x_0 - x^\star - B_0^{-1}f(x_0)\|, \qquad (15)\\
&= \| - B_0^{-1}\left(f(x_0) - f(x^\star) - f'(x^\star)\cdot(x_0 - x^\star)\right)\\
&\quad - B_0^{-1}\left(f'(x^\star)(x_0 - x^\star) - B_0(x_0 - x^\star)\right)\|,\\
&= \| - B_0^{-1}\left(f(x_0) - f(x^\star) - f'(x^\star)\cdot(x_0 - x^\star)\right)\\
&\quad - B_0^{-1}\left((f'(x^\star) - B_0)(x_0 - x^\star)\right)\|,\\
&\leq \|B_0^{-1}\|\left(\lambda\|x_0 - x^\star\| + 2\delta\|x_0 - x^\star\|\right),\\
&\leq \|B_0^{-1}\|(\lambda + 2\delta)\|x_0 - x^\star\|,\\
&\leq \gamma(1+r)(\delta(1-r) + 2\delta)\|x_0 - x^\star\|,\\
&\leq \gamma(1+r)\delta(3-r)\|x_0 - x^\star\| \qquad \text{by Eq. (14) (middle)}\\
&\leq r\|x_0 - x^\star\|. \qquad (16)
\end{aligned}$$

Consequently, $\|x_1 - x^\star\| \leq r\|x_0 - x^\star\|$ and $\|x_1 - x^\star\| \leq r\eta < \eta$, *i.e.* $x_1 \in B(x^\star, \eta)$.

Eq. (3) defines B_1 by $B_1 = B_0 - (y_1 - B_0 s_1)\cdot u_1{}^t$ for some u_1 verifying $\|u_1\| = \|s_1\|^{-1}$ (see Eqs. (4), Corollary 3.2). Then:

$$\|B_1 - B_0\| = \|f(x_1) - f(x_0) - B_0(x_1 - x_0)\| \cdot \|x_1 - x_0\|^{-1}.$$

Therefore,

$$\begin{aligned}
\|B_1 - f'(x^\star)\| &\leq \max\Big(\quad \|B_0 - f'(x^\star)\|\,, \qquad (17)\\
&\qquad \|f(x_1) - f(x_0) - B_0(x_1 - x_0)\|\|x_1 - x_0\|^{-1}\Big),\\
&\leq \max\Big(\quad \|B_0 - f'(x^\star)\|\,,\\
&\qquad \|\left(B_0 - f'(x^\star)\right)(x_1 - x_0)\|\|x_1 - x_0\|^{-1},\\
&\qquad \|f(x_1) - f(x_0) - f'(x^\star)(x_1 - x_0)\|\|x_1 - x_0\|^{-1}\Big),\\
&\leq \max(\delta, \lambda) \leq \delta.
\end{aligned}$$

We can then carry on and prove by induction that for all k,

$$\text{(i)}\ \ \|x_k - x^\star\| \leq r^k\|x_0 - x^\star\|, \quad \text{and} \quad \text{(ii)}\ B_k \in B(f'(x^\star), \delta). \quad (18)$$

Heredity for Inequality (18)-(i) comes from: a same use of the Banach Perturbation Lemma on B_k so that B_k is invertible; that $\|B_k^{-1}\| \leq (1+r)\gamma$ and by repeating the computations (15) to (16):

$$\begin{aligned}
\|x_{k+1} - x^\star\| &\leq \|B_k\|^{-1}(\lambda + 2\delta)\|x_k - x^\star\|,\\
&\leq (1+r)\gamma\delta(3-r)\|x_k - x^\star\|,\\
&\leq r\|x_k - x^\star\|.
\end{aligned}$$

We can deal with (18)-(ii) using a similar computation as (17):

$$\begin{aligned}
\|B_{k+1} - f'(x^\star)\| \leq\ & \max\left(\|B_k - f'(x^\star)\|, \qquad (19)\right.\\
&\left.\|f(x_{k+1}) - f(x_k) - B_k(x_{k+1} - x_k)\|\|x_{k+1} - x_k\|^{-1}\right)\\
\leq\ & \max\left(\|B_k - f'(x^\star)\|,\right.\\
&\left.\|f(x_{k+1}) - f(x_k) - f'(x^\star)(x_{k+1} - x_k)\|\|x_{k+1} - x_k\|^{-1}\right),\\
\leq\ & \max(\delta, \lambda) \leq \delta. \qquad \square
\end{aligned}$$

Corollary 4.3. *Locally, one can take definition* (12) *to define all the u_n's and all the B_n's will still be invertible.*

Proof. With the assumptions of the proof of Theorem 4.2, for u_n defined by (12), $\|u_{n-1}\| = \|s_{n-1}\|^{-1}$ and (4) are satisfied, and by the Banach Perturbation Lemma, B_n defined by (3) is invertible. $\square$

Remark 4.4. The fact that Broyden's method has locally Q-linear convergence with ratio r for any r is not enough to prove that ithas Q-superlinear convergence. Indeed, as x_k is going closer to $x^\star$, there is no reason for B_k to get closer to $f'(x^\star)$. Consequently, we cannot expect from the previous result that x_k and B_k enter loci of smaller ratio of convergence as k goes to infinity. In fact, in general, B_k does not converge to $f'(x^\star)$.

Finally, the next lemma, consequence of the previous theorem, will be useful in the next subsection to obtain the R-superlinear convergence.

Lemma 4.5. *Using the same notations as in the proof of Theorem 4.2, if $r \leq \left(\frac{\gamma\|f'(x^\star)\|}{2}\right)^{-1}$, and $\|B_0 - f'(x^\star)\| < \delta$ and $\|x_0 - x^\star\| < \eta$, then for all $n \in \mathbb{N}$,*

$$\|f_{n+1}\| \leq \|f_n\|.$$

Proof. Let $n \in \mathbb{N}$. We have $\|s_n\| \leq r\|s_{n-1}\|$. Indeed, from $\|x_{n+1} - x_n\| \leq \max(\|x_{n+1} - x^\star\|,\ \|x^\star - x_n\|)$, and $\|x_{n+1} - x_n\| < \|x_n - x^\star\|$, we see that $\|s_n\| = \|x^\star - x_n\| \leq r\|x^\star - x_{n-1}\| = r\|s_{n-1}\|$.

Then using (QN) and the Q-linear convergence with ratio r, we get that $\|f_{n+1}\| \leq r\|B_{n+1}\|\|B_n^{-1}\|\|f_n\|$. Using (19), the definition of δ, γ in (14), and the fact that $0 < r < 1$, we get that $\|B_{n+1}\|\|B_n^{-1}\| \leq 2\gamma\|f'(x^\star)\|$, which concludes the proof. $\square$

4.2 Local R-superlinear convergence

We first remark that the $2n$-step convergence in the linear case proved by Gay in [13] is still valid. Indeed, it is only a matter of linear algebra.

Theorem 4.6 (Theorem 2.2 in [13]). *If f is defined by $f(x) = Ax - b$ for some $A \in GL_m(K)$, then any quasi-Newton method converges in at most $2m$ steps (i.e. $f(x_{2m}) = 0$).*

With this and under a stronger differentiability assumption on f, we can obtain R-superlinearity, similarly to Theorem 3.1 of [13]. The proof also follows the main steps thereof.

Theorem 4.7. *Let us assume that on a neighborhood U of $x^\star$, there is a $c_0 \in \mathbb{R}_{>0}$ such that f satisfies*[5]

$$\forall x, y \in U, \|f(x) - f(y) - f'(x^\star)\cdot(x - y)\| \leq c_0\|x - y\|^2. \quad (20)$$

Then there are η, δ and Γ in $\mathbb{R}_{>0}$ such that if $x_0 \in B(x^\star, \eta)$ and $B_0 \in B(f'(x^\star), \delta)$, then for any $w \in \mathbb{Z}_{\geq 0}$,

$$\|x_{w+2m} - x^\star\| \leq \Gamma\|x_w - x^\star\|^2.$$

Proof. *Step 1: Preliminaries.* Condition (20) is stronger than strict differentiability as stated in Theorem 4.2. From its proof and Lemma 4.5, let $r \in (0,1)$ and $\gamma \geq \|f'(x^\star)^{-1}\|$, as well as η and δ such that: $r \leq \left(\frac{\gamma\|f'(x^\star)\|}{2}\right)^{-1}$, and if $x_0 \in B(x^\star, \eta)$ and $B_0 \in B(f'(x^\star), \delta)$, the sequences $(x_n)_{n\in\mathbb{N}}$ and $(B_n)_{n\in\mathbb{N}}$ defined by Broyden's method

[5] This condition is satisfied by polynomials or converging power series.

(using (12)) are well defined and moreover the four following inequalities are satisfied: for any $k \in \mathbb{N}$,

$$\|B_k - f'(x^\star)\| \le \delta, \quad \|x_{k+1} - x^\star\| \le r\|x_k - x^\star\|,$$
$$\|B_k^{-1}\| \le (1+r)\gamma, \quad \|f(x_{k+1})\| \le \|f(x_k)\|.$$

Let $x_0 \in B(x^\star, \eta)$, $B_0 \in B(f'(x^\star), \delta)$, and $(x_n)_{n\in\mathbb{N}}$ and $(B_n)_{n\in\mathbb{N}}$ be defined by Broyden's method. Let $w \in \mathbb{N}$ and $h = \|x_w - x^\star\|$. We must show that there is a Γ, independent of w such that $\|x_{w+2m} - x^\star\| \le \Gamma h^2$.

Step 2: reference to a linear map. Let the linear affine map $\hat{f}(x) = f'(x^\star)\left(x - x^\star\right)$, and $\hat{x_0} = x_w$ and $\hat{B}_0 = B_w$. Broyden's method (using first (12)) applied to those data produces the sequences $(\hat{x}_n)_{n\in\mathbb{N}}$ and $(\hat{B}_n)_{n\in\mathbb{N}}$, which are constant for $n \ge 2m$, as a result of Theorem 4.2. We define similarly $\hat{s}_n = \hat{x}_{n+1} - \hat{x}_n$. We have again for all $k \in \mathbb{N}$ the four inequalities:

$$\|\hat{B}_k - f'(x^\star)\| \le \delta, \quad \|\hat{x}_{k+1} - x^\star\| \le r\|\hat{x}_k - x^\star\|,$$
$$\|\hat{B}_k^{-1}\| \le (1+r)\gamma \quad \|\hat{f}(x_{k+1})\| \le \|\hat{f}(x_k)\|.$$

The key to the proof is that $\hat{x}_{2m} = x^\star$ and $\hat{x}_k$ and x_{w+k} are not too much far apart.

Step 3: Statement of the induction. More concretely, we prove by induction on j that there exist $\gamma_{1,j}$ and $\gamma_{2,j}$, independent of w, such that for $0 \le j \le 2m$, we have the two inequalities:

$$\|B_{w+j} - \hat{B}_j\| \cdot \|f_{w+j}\| \le \gamma_{1,j} h^2, \tag{$E_{1,j}$}$$

$$\|x_{w+j} - \hat{x}_j\| \le \gamma_{2,j} h^2. \tag{$E_{2,j}$}$$

Step 4: Base case. Since $B_w = \hat{B}_0$ and $x_w = \hat{x}_0$, $(E_{1,0})$ and $(E_{2,0})$ are clear, with $\gamma_{1,0} = \gamma_{2,0} = 0$. Now, let us assume that $(E_{1,k})$ and $(E_{2,k})$ are true for a given k such that $0 \le k < 2m$.

Step 5: We first prove $(E_{2,k+1})$. One part of the inequality (21) is obtained thanks to: $B_{w+k}^{-1} - \hat{B}_k^{-1} = B_{w+k}^{-1}(\hat{B}_k - B_{w+k})\hat{B}_k^{-1}$.

$$\begin{aligned}\|s_{w+k} - \hat{s}_k\| &= \|B_{w+k}^{-1} f_{w+k} - \hat{B}_k^{-1}\hat{f}(\hat{x}_k)\| \\ &\le \max\Big(\|B_{w+k}^{-1}\| \cdot \|\hat{B}_k^{-1}\| \cdot \|B_{w+k} - \hat{B}_k\| \cdot \|f_{w+k}\|, \quad (21)\\ &\qquad \|\hat{B}_k^{-1}\| \cdot \|f_{w+k} - \hat{f}(\hat{x}_k)\|\Big) \\ &\le \|\hat{B}_k^{-1}\| \max\Big(\|B_{w+k}^{-1}\| \cdot \|B_{w+k} - \hat{B}_k\| \cdot \|f_{w+k}\|, \\ &\qquad \|f_{w+k} - \hat{f}(x_{w+k})\|, \ \|\hat{f}(x_{w+k}) - \hat{f}(\hat{x}_k)\|\Big) \quad (22)\end{aligned}$$

The first term on the r.h.s. of (22) is upper-bounded by $(1+r)^2\gamma^2\gamma_{1,k}h^2$ using $(E_{1,k})$ and $\|B_{w+k}^{-1}\| \le (1+r)\gamma$.

For the second term of (22), using (20):

$$\|f_{w+k} - f(x^\star) - f'(x^\star)\cdot(x_{w+k} - x^\star)\| \le c_0\|x_{w+k} - x^\star\|^2$$

and $\|x_{w+k} - x^\star\| \le \|x_w - x^\star\| = h$, it is upper-bounded by c_0h^2. Finally, the last term is equal to $f'(x^\star)(x_{w+k} - \hat{x}_k)$ whose norm is upper-bounded by $\|f'(x^\star)\|\gamma_{2,k}h^2$ thanks to $(E_{2,k})$. This is enough to define $\gamma_{3,k}$ such that $\|s_{w+k} - \hat{s}_k\| \le \gamma_{3,k}h^2$ (‡). Consequently, with $\gamma_{2,k+1} = \max(\gamma_{3,k}, \gamma_{2,k})$, we do have $\|x_{w+k+1} - \hat{x}_{k+1}\| \le \gamma_{2,k+1}h^2$, and $(E_{2,k+1})$ is satisfied.

Step 6.0: We now prove $(E_{1,k+1})$. We first deal with some preliminary cases. If $s_{w+k} = 0$, (that is $x_{w+k+1} = x_{w+k}$) then the property (2) $s_{w+k} = -B_{w+k}^{-1}f_{w+k}$ implies that $f_{w+k} = 0$, and the property $B_{w+k+1}s_{w+k} = y_{w+k}$ implies that $f_{w+k} = f_{w+k+1} = 0$. Thus $(E_{1,k+1})$ is satisfied with $\gamma_{1,k+1} = 0$. If $\hat{s}_k = 0$, then similarly $\hat{f}(\hat{x}_{w+k}) = \hat{f}(\hat{x}_{w+k+1}) = 0$. Therefore, as we have seen before,

$$\begin{aligned}\|f_{w+k+1}\| &= \|f_{w+k+1} - \hat{f}(x_{w+k+1}) + \hat{f}(x_{w+k+1}) - \hat{f}(\hat{x}_{k+1})\|, \\ &\le \max\left(c_0, \|f'(x^\star)\|\gamma_{2,k+1}\right)h^2.\end{aligned}$$

Then, using that $\|B_{w+k+1} - \hat{B}_{k+1}\| \le \max(\|B_{w+k+1} - f'(x^\star)\|, \|\hat{B}_{k+1} - f'(x^\star)\|) \le \delta$, $(E_{1,k+1})$ is satisfied with:

$$\gamma_{1,k+1} = \delta h^2 \max\left(c_0, \|f'(x^\star)\|\gamma_{2,k+1}\right).$$

Step 6.1 : We can now assume that both s_k and $\hat{s}_k$ are non zero. To prove that there is a $\gamma_{1,k+1}$ (independent of w) such that $(E_{1,k+1})$ holds, then in view of the fact that $\|f_{w+k+1}\| \le \|f_{w+k}\|$ (Lemma 4.5) of $(E_{1,k})$ and of the definition (Eq. (3)) of B_{k+1} and $\hat{B}_{k+1}$, it is enough to prove that there is some $\gamma_{4,k+1}$ (independent of w) such that:

$$\|\left(y_{w+k} - B_{w+k}s_{w+k}\right)u_{w+k}{}^t - \left(\hat{y}_k - \hat{B}_k\hat{s}_k\right)\hat{u}_k{}^t\| \cdot \|f_{w+k+1}\| \le \gamma_{4,k+1}h^2. \tag{23}$$

Using that $\|f_{w+k+1}\| \le \|f_{w+k}\|$ (by Lemma 4.5), we obtain:

$$\begin{aligned}&\|f_{w+k+1}\| \cdot \|\left(y_{w+k} - B_{w+k}s_{w+k}\right)u_{w+k}{}^t - \left(\hat{y}_k - \hat{B}_k\hat{s}_k\right)\hat{u}_k{}^t\| \\ &\le \|f_{w+k}\| \max\Big(\|y_{w+k} - f'(x^\star)s_{w+k}\| \cdot \|u_{w+k}{}^t\|, \\ &\qquad \|(f'(x^\star) - B_{w+k})s_{w+k}u_{w+k}{}^t - (f'(x^\star) - \hat{B}_k)\hat{s}_k\hat{u}_k{}^t\|\Big) \\ &\le \|f_{w+k}\| \max\Big(\|y_{w+k} - f'(x^\star)s_{w+k}\| \cdot \|u_{w+k}{}^t\|, \quad (24)\\ &\qquad \|(f'(x^\star) - \hat{B}_k)(s_{w+k}u_{w+k}{}^t - \hat{s}_k\hat{u}_k{}^t)\|, \quad (25)\\ &\qquad \|(B_{w+k} - \hat{B}_k)s_{w+k}u_{w+k}{}^t\|\Big). \quad (26)\end{aligned}$$

Step 6.2: From $f_{w+k} = -B_{w+k}s_{w+k}$, we have $\|f_{w+k}\| \le \|s_{w+k}\| \cdot \max(\|B_{w+k} - f'(x^\star)\|, \|f'(x^\star)\|) \le \|s_{w+k}\| \cdot \max(\delta, \|f'(x^\star)\|)$ (•). Otoh by (20), $\|y_{w+k} - f'(x^\star)s_{w+k}\| \le c_0\|s_{w+k}\|^2$. It follows that the first term (24) can be upper-bounded in the following way:

$$(24) \le c_0\|s_{w+k}\|^3\|u_{w+k}{}^t\| \max(\delta, \|f'(x^\star)\|) \le c_0h^2 \max(\delta, \|f'(x^\star)\|),$$

the rightmost inequality being obtained from $\|u_{w+k}{}^t\| = \|s_{w+k}\|^{-1}$ and $\|s_{w+k}\| \le \max(\|x_{w+k+1} - x^\star\|, \|x_{w+k} - x^\star\|) = \|x_{w+k} - x^\star\| \le \|x_w - x^\star\| = h$.

Step 6.3: The third one (26) can be upper-bounded using $(E_{1,k})$:

$$(26) \le \|f_{w+k}\|\|(B_{w+k} - \hat{B}_k)s_{w+k}u_{w+k}{}^t\| \le \gamma_{1,k}h^2.$$

Step 6.4: For the second one (25), observe that:

$$s_{w+k}u_{w+k}{}^t - \hat{s}_k\hat{u}_k{}^t = (s_{w+k} - \hat{s}_k)u_{w+k}{}^t - \hat{s}_k(u_{w+k}{}^t - \hat{u}_k{}^t). \tag{27}$$

The first term is easy to manage using the previous inequality (•) on $\|f_{w+k}\|$, the inequality (‡) on $\|s_{w+k} - \hat{s}_k\|$ and $\|s_{w+k}\|\|u_{w+k}{}^t\| = 1$:

$$\|f_{w+k}\| \cdot \|(s_{w+k} - \hat{s}_k)u_{w+k}{}^t\| \le \max(\delta, \|f'(x^\star)\|)\gamma_{3,k}h^2. \tag{28}$$

The second one of Eq. (27) is a little bit trickier. Define as in (12), $u_{w+k} = s_{w+k,l}^{-1}e_l$ and $\hat{u}_k = \hat{s}_{k,\hat{l}}^{-1}e_{\hat{l}}$ for some given l and $\hat{l}$.

If $l = \hat{l}$, we have: (the last inequality below follows from (‡)).

$$\begin{aligned}\|u_{w+k} - \hat{u}_k\| &= |s_{w+k,l}^{-1} - \hat{s}_{k,l}^{-1}| = \frac{|s_{w+k,l} - \hat{s}_{k,l}|}{|s_{w+k,l}| \cdot |\hat{s}_{k,l}|} = \frac{|s_{w+k,l} - \hat{s}_{k,l}|}{\|s_{w+k}\| \cdot \|\hat{s}_k\|} \\ &\le \frac{\|s_{w+k} - \hat{s}_k\|}{\|s_{w+k}\| \cdot \|\hat{s}_k\|} \le \frac{\gamma_{3,k}h^2}{\|s_{w+k}\| \cdot \|\hat{s}_k\|}.\end{aligned}$$

From this and from $\|f_{w+k}\| = \|B_{w+k}\| \cdot \|s_{w+k}\|$ we get:

$$\|f_{w+k}\| \cdot \|u_{w+k} - \hat{u}_k\| \cdot \|\hat{s}_k\| \leq \gamma_{3,k} \max\left(\delta, \|f'(x^\star)\|\right) h^2. \quad (29)$$

If $l \neq \hat{l}$, then either $\|s_{w+k} - \hat{s}_k\| = \|s_{w+k}\|$, if $\|\hat{s}_k\| \leq \|s_{w+k}\|$, or $\|s_{w+k} - \hat{s}_k\| = \|\hat{s}_k\|$, if $\|s_{w+k}\| \leq \|\hat{s}_k\|$. In the first case, we have

$$\|u_{w+k} - \hat{u}_k\| = \|\hat{s}_k\|^{-1},$$

and then, the second term of (27) multiplied by $\|f_{w+k}\|$ verifies:

$$\begin{aligned}\|f_{w+k}\| \cdot \|u_{w+k} - \hat{u}_k\| \cdot \|\hat{s}_k\| &\leq \max\left(\delta, \|f'(x^\star)\|\right) \|s_{w+k}\| \\ &\leq \max\left(\delta, \|f'(x^\star)\|\right) \gamma_{3,k} h^2. \quad (30)\end{aligned}$$

The second case follows with the same computation. Eqs (30) (29) (28) prove together the bound on the expression (25) in (27). In turn with the bounds on the terms (24) and (26), prove (23). This concludes the proof of $(E_{1,k+1})$, and finally the induction.

Step 7: Consequently, $\|x_{w+2m} - \hat{x}_{2m}\| \leq \gamma_{2,2m} h^2$. Thanks to Theorem 4.2, $\hat{x}_{2m} = x^\star$, and thus, we have proved that for any w,

$$\|x_{w+2m} - x^\star\| \leq \gamma_{2,2m} \|x_w - x^\star\|^2. \qquad \square$$

Theorem 4.7 has for immediate consequence:

Theorem 4.8. *Broyden's method has locally R-order of convergence* $2^{\frac{1}{2m}}$.

Proof. Let us take x_0 and B_0 as in the proof of the previous theorem, and same constants and notations. For any w, $\|x_{w+2m} - x^\star\| \leq \Gamma \|x_w - x^\star\|^2$.

Consequently, for $0 \leq k < 2m$, $l \in \mathbb{N}$, and $\mu = 2^{1/2m}$,

$$\begin{aligned}\|x_{2lm+k} - x^\star\|^{\mu^{-2lm-k}} &\leq \|x_k - x^\star\|^{2^l \mu^{-2lm-k}} \Gamma^{(2^l-1)\mu^{-2lm-k}} \\ &\leq \|x_k - x^\star\|^{2^l 2^{-l-\frac{k}{2m}}} \Gamma^{(2^l-1)2^{-l-\frac{k}{2m}}} \\ &\leq \|x_k - x^\star\|^{2^{-\frac{k}{2m}}} \Gamma^{(1-2^{-l})2^{-\frac{k}{2m}}}.\end{aligned}$$

For simplicity, we can assume that $\Gamma \geq 1$. Thus,

$$\begin{aligned}\|x_{2lm+k} - x^\star\|^{\mu^{-2lm-k}} &\leq \|x_k - x^\star\|^{2^{-\frac{k}{2m}}} \Gamma^{2^{-\frac{k}{2m}}}. \\ &\leq \|x_0 - x^\star\|^{2^{-\frac{k}{2m}}} \Gamma^{2^{-\frac{k}{2m}}}.\end{aligned}$$

Therefore, for $\|x_0 - x^\star\|$ small enough, we get that for all k such that $0 \leq k < 2m$, $\|x_0 - x^\star\|^{2^{-\frac{k}{2m}}} \Gamma^{2^{-\frac{k}{2m}}} < 1$, and hence, $\limsup_s \|x_s - x^\star\|^{\mu^s} < 1$. From 9.2.7 of [20], we then obtain that Broyden's method do have locally R-order of convergence $2^{\frac{1}{2m}}$. $\square$

5 QUESTIONS ON Q-SUPERLINEARITY

A Q-order of μ implies an R-order of μ. The converse is not true. Over $\mathbb{R}$, one of the most important result concerning Broyden's method is that it is Q-superlinear. The extension of this result to the non-archimedean case remains an open question.

5.1 Dimension 1: secant method

In dimension one, Broyden's method reduces to the secant method.

It is known since [1] that the p-adic secant method applied on polynomials has order Φ, the golden ratio. Its generalization to a general non-archimedean context is straightforward.

Proposition 5.1. *Let us assume that $m = 1$ and on a neighborhood U of $x^\star$, there is a $c_0 \in \mathbb{R}_{>0}$ such that f satisfies (20) on U. Then the secant method has locally Q-order of convergence Φ.*

Proof. Let us assume that we are in the same context as in the proof of Theorem 4.7, with some Q-linear convergence of ratio $r < 1$. Let us define $\varepsilon_k = x_k - x^\star$ for $k \in \mathbb{N}$. For all $k \in \mathbb{N}$, $|\varepsilon_{k+1}| < |\varepsilon_k|$. Then by ultrametricity, $|x_{k+1} - x_k| = |\varepsilon_k|$. Also, we further assume that $c_0|\varepsilon_0| < |f'(x^\star)|$ so that for all $k \in \mathbb{N}$, $|f'(x^\star) \times (x_{k+1} - x_k)| > c_0|(x_{k+1} - x_k)|^2$, which also implies by ultrametricity and (20) that for all $k \in \mathbb{N}$,

$$|f(x_{k+1}) - f(x_k)| = |f'(x^\star) \times (x_{k+1} - x_k)|.$$

Similarly, $|f(x_k)| = |f'(x^\star)||\varepsilon_k|$.

Now, let $n \in \mathbb{Z}_{>0}$. Broyden's iteration is given by:

$$x_{n+1} = x_n - \frac{x_n - x_{n-1}}{f(x_n) - f(x_{n-1})}.$$

It rewrites as:

$$\begin{aligned}|\varepsilon_{n+1}| = |\varepsilon_n - \frac{\varepsilon_n f(x_n) - \varepsilon_{n-1} f(x_n)}{f(x_n) - f(x_{n-1})}| &= |\frac{\varepsilon_{n-1} f(x_n) - \varepsilon_n f(x_{n-1})}{f(x_n) - f(x_{n-1})}| \\ \leq c_0 \frac{\max\left(|\varepsilon_{n-1}||\varepsilon_n|^2,\ |\varepsilon_{n-1}|^2|\varepsilon_n|\right)}{|f(x_n) - f(x_{n-1})|} &\leq \frac{c_0}{|f'(x^\star)|} |\varepsilon_n||\varepsilon_{n-1}|.\end{aligned}$$

Let us write $C = \frac{c_0}{|f'(x^\star)|}$ and $v_n = C\varepsilon_n$. Then, $v_{n+1} \leq v_n v_{n-1}$ for any $n > 0$ and consequently,

$$v_{n+1}/(v_n^\Phi) \leq v_n^{1-\Phi} v_{n-1} \leq \left(v_n / v_{n-1}^\Phi\right)^{1-\Phi},$$

as $\Phi^2 = \Phi + 1$. If we define $(Y_n)_{n \in \mathbb{Z}_{\geq 1}}$ by $Y_1 = \frac{v_1}{v_0^\Phi}$ and $Y_{n+1} = Y_n^{1-\Phi}$, then $\frac{v_{n+1}}{v_n^\Phi} \leq Y_n$. Since $|1 - \Phi| < 1$, then Y_n converges to 1. Therefore, it is bounded by some $D \in \mathbb{R}_+$, and $\frac{v_{n+1}}{v_n^\Phi} \leq D$ for all $n \in \mathbb{Z}_{\geq 1}$. This concludes the proof. $\square$

5.2 General case

Over $\mathbb{R}$, Broyden's method is known to converge Q-superlinearly. The key point is that for any $E \in M_m(\mathbb{R})$ and $s \in \mathbb{R}^m \setminus \{0\}$,

$$\|E\left(I - \frac{s \cdot s^t}{(s^t \cdot s)}\right)\|_F^2 = \|E\|_F^2 - \left(\frac{\|Es\|_2}{\|s\|_2}\right)^2, \quad (31)$$

equation (5.5) of [10]. The minus sign is a blessing as it allows the appearance of a telescopic sum which plays a key role in proving that $\frac{\|x_{n+1} - x^\star\|}{\|x_n - x^\star\|}$ converges to zero. Unfortunately, there does not seem to be a non-archimedean analogue to this equality. Thanks to Theorem 4.7, we nevertheless believe in the following conjecture.

Conjecture 5.2. *In the same setting as Theorem 4.7, Broyden's method has locally Q-superlinear convergence.*

6 FINITE PRECISION

6.1 Design and notations

One remarkable feature of Newton's method in an ultrametric context is the way it can handle precision. For example, if π is a uniformizer, if we assume that $\|f'(x^\star)^{-1}\| = 1$, x_n known at precision $O(\pi^{2^n})$ is enough to obtain x_{n+1} at precision $O(\pi^{2^{n+1}})$. To that intent, it thus suffices to double the precision at each new iteration. Hence the working precision of Newton's method can be taken to grow at the same rate as the rate of convergence.

The handling of precision is more subtle in Broyden. This is however crucial to design efficient implementations. Note that in the real numerical setting, most works using Broyden's methods

are employing fixed finite precision arithmetic, and do not address precision. Additionally, the lack of a knowledge of a precise exponent of convergence requires special care, and the presence of a division also complicates the matter. We explain hereafter how to cope with those issues.

We will make the following hypotheses throughout this section, which correspond to the standard ones in the Newton-Hensel method. They are that the starting x_0 and B_0 are in a basin of convergence at least linear. This allows us to replace any encountered x_n by its lift $\tilde{x}_n$ to a higher precision (and same for B_n). Indeed, $\tilde{x}_n$ will still be in the basin of convergence and then follows the same convergence property. These liftings allow to mitigate the fact that some divisions are reducing the amount of precision so that only arbitrary added digits are destroyed by the divisions. [6]

Assumption 6.1. *We assume that x_0 and $x^\star$ are in O_K, and that $\|f'(x^\star)\| = \|f'(x^\star)^{-1}\| = \|B_0\| = \|B_0^{-1}\| = 1$. We also assume that some $\rho_1 \le 1$ and $\rho_2 \le 1$ are given such that $B(x^\star, \rho_1) \times B(f'(x^\star), \rho_2)$, is a basin of convergence at least linear and for any $x \in B(x^\star, \rho_1)$, and $\rho \le \rho_1$, $f(x + B(0,\rho)) = f(x) + f'(x^\star) \cdot B(0,\rho)$ (see the Precision Lemma 3.16 of [9])*

The assumption on B_0 and $f'(x^\star)$ states that they are unimodular, which is the best one can assume regarding to conditioning and precision. Indeed if $M \in GL_m(K)$ is unimodular ($\|M\| = \|M^{-1}\| = 1$), then for any $x \in K^m$, $\|Mx\| = \|x\|$. Over $\mathbb{Q}_p$, $M \in M_m(\mathbb{Z}_p)$ is unimodular if and only if its reduction in $M_m(\mathbb{Z}/p\mathbb{Z})$ is invertible (and *idem* for $\mathbb{Q}[\![T]\!]$ and $\mathbb{Q}$). The last assumption is there to provide the precision on the evaluations $f(x_k)$'s. It is satisfied if $f \in O_K[X_1, \ldots, X_m]$.

Precision and complexity settings. Let $\mathsf{M}(N)$ be a superadditive upper-bound on the arithmetic complexity over the residue field of O_K for the computation of the product of two elements in O_K at precision $O(\pi^N)$, and L be the size of a straight-line program that computes the system f. One can take $\mathsf{M}(N) \in \tilde{O(N)}$.

Working over K with *zealous* arithmetic, the ultrametric counterpart of interval arithmetic [9, § 2.1], the interval of integers $[\![a,\ b[\![$ indicates the coefficients of an element $x \in K$ represented in the computer as $x = \sum_{i=a}^{b-1} x_i \pi^i$, with $x_i \in O_K/\langle\pi\rangle$. In this way $\mathrm{val}(x) = a$, its *absolute precision* is $\mathrm{abs}(x) = b$, and its *relative precision* is $\mathrm{rel}(x) = b - a$. We recall the usual precision formulae, and assume in the algorithm below that it is how the software manages zealous arithmetic (as in Magma, SageMath, Pari). See *loc. cit.* for more details.

$$[\![a,\ b[\![\times [\![c,\ d[\![= [\![a + c,\ \min(a + d,\ b + c)[\![$$
$$[\![a,\ b[\![/ [\![c,\ d[\![= [\![a - c,\ \min(a + d - 2c,\ b - c)[\![\qquad \text{(P)}$$

The cost of multiplying two elements of relative precision a and b is within $\mathsf{M}(\max(a,\ b))$, and to divide one by the other is in $4\mathsf{M}(\max(a,\ b)) + \max(a,\ b)$ [25, Thm 9.4].

To perform changes in the precision, we use the same notation as Magma's function for doing so. If x has interval $[\![a,\ b[\![$, the (destructive) procedure "ChangePrec(~x, c)" either truncates x to absolute precision c if $c \le b$, or lifts with zero coefficients $0\pi^b + \cdots + 0\pi^{c-1}$ to fit the interval $[\![a,\ c[\![$, if $c > b$. The non-destructive counterpart is denoted "ChangePrec(x, c)" without ~.

[6] This an example of an adaptive method, which can also be used in Newton's method when divisions occur.

6.2 Effective Broyden's method

We start from an initial approximation x_0 at precision one, for example given by a modular method. The inverse of the Jacobian at precision one provides B_0^{-1}. It yields a cost of $O(m^\omega)$, but the complexity analysis of Remark 6.4 shows that it is negligible. Obtaining these data is not always obvious [12], but is the standard hypothesis in the context of modular methods. We write $v_k = \mathrm{val}(f_k)$,

In an ideal situation. Assume an oracle provides the valuations $v_0, v_1, v_2, \ldots, v_n, \ldots$ (computed by a Broyden method at arbitrarily large precision). From this ideal situation, we derive the simple and costless modifications required in reality. This analysis allows us to know how efficient can a Broyden method be, which is noteworthy for comparing it to Newton's. The implementation of Iteration n ($n = 0$ included) follows the lines hereunder. The rightmost interval indicates the output interval precision of the object computed (following (P)), while the middle indicates a complexity estimate.

Input: (1) B_n^{-1} has interval $[\![0,\ v_n[\![$ and is unimodular.
(2) x_n has interval $[\![0,\ v_n + v_{n+1}[\![$ (non-zero entries in $[\![0,\ v_{n-1} + v_n[\![$).
(3) f_n has interval $[\![v_n,\ v_n + v_{n+1}[\![$.

Output: (i) B_{n+1}^{-1} with interval $[\![0,\ v_{n+1}[\![$, ($\mathrm{val}(\det(B_n^{-1})) = 0$).
(ii) x_{n+1} in the interval $[\![0,\ v_{n+1}+v_{n+2}[\![$ (non-zero entries in $[\![0,\ v_n+v_{n+1}[\![$).
(iii) f_{n+1} in the interval $[\![0,\ v_{n+1} + v_{n+2}[\![$.

(1)	ChangePrec(~B_n^{-1}, v_{n+1}) ;		$[\![0,\ v_{n+1}[\![$
(2)	$s_n \leftarrow -B_n^{-1} \cdot f_n$;	$m^2\mathsf{M}(v_{n+1})$	$[\![v_n,\ v_n + v_{n+1}[\![$
(3)	$x_{n+1} \leftarrow x_n + s_n$;		$[\![0,\ v_n + v_{n+1}[\![$
(4)	ChangePrec(~x_{n+1}, $v_{n+1} + v_{n+2}$) ;		$[\![0,\ v_{n+1} + v_{n+2}[\![$
(5)	$f_{n+1} \leftarrow f(x_{n+1})$;	$L \cdot \mathsf{M}(v_{n+1} + v_{n+2})$	$[\![v_{n+1},\ v_{n+1} + v_{n+2}[\![$
(6)	$\overline{f_{n+1}} \leftarrow$ ChangePrec(f_{n+1}, $v_n + v_{n+1}$) ;		$[\![v_{n+1},\ v_{n+1} + v_n[\![$
(7)	$h_n \leftarrow B_n^{-1} \cdot \overline{f_{n+1}}$;	$m^2\mathsf{M}(v_{n+1})$	$[\![v_{n+1},\ v_n + v_{n+1}[\![$
(8)	$u_n \leftarrow$ Eq.(12) ;	(negligible)	$[\![-v_n,\ v_{n+1} - v_n[\![$
(9)	$r_n \leftarrow u_n^T \cdot$ ChangePrec(B_n^{-1}, v_n) ;	$m^2\mathsf{M}(v_{n+1})$	$[\![-v_n,\ 0[\![$
(10)	ChangePrec(~$\overline{f_{n+1}}$, $2v_n$) ;		$[\![v_{n+1},\ 2v_n[\![$
(11)	den $\leftarrow 1 + r_n \cdot \overline{f_{n+1}}$;	$m\mathsf{M}(v_{n+1})$	$[\![0,\ v_n[\![$
(12)	Num $\leftarrow h_n \cdot r_n$;	$m^2\mathsf{M}(v_n)$	$[\![v_{n+1} - v_n,\ v_{n+1}[\![$
(13)	$\mathrm{N}_n \leftarrow$ Num/den ;	$4m^2\mathsf{M}(v_n)$	$[\![v_{n+1} - v_n,\ v_{n+1}[\![$
(14)	$B_{n+1}^{-1} \leftarrow B_n^{-1} - \mathrm{N}_n$;		$[\![0,\ v_{n+1}[\![$
(15)	return B_{n+1}^{-1}, x_{n+1}, f_{n+1}		

We emphasize again that thanks to the careful changes of precision undertaken, the precisions are automatically managed by the software, would it have zealous arithmetic implemented. It is then immediate to check that the output verifies the specifications. Moreover from the positive valuation of N_n it is clear that B_{n+1} is unimodular. Thus Iteration $n+1$ can be initiated with these outputs.

Complexity of the ideal situation. The arithmetic cost of Iteration n is within $(3m^2 + m)\mathsf{M}(v_{n+1}) + 5m^2\mathsf{M}(v_n) + L \cdot \mathsf{M}(v_{n+2} + v_{n+1})$. If we assume an exponent of convergence $\alpha > 1$, *i.e.* $v_{n+1} \approx \alpha v_n$ for "not too small" n, then the total cost to reach a precision $N \approx \alpha^{\ell+1} \approx v_{\ell+1}$ (ℓ steps, including a 0-th one) is upper-bounded by

$$(5m^2 + (3m^2 + m)\alpha^2 + L(1+\alpha)^2\alpha^2)\mathsf{M}(N/(\alpha - 1)) \qquad (32)$$

In reality. Using the same notations and inputs at Iteration n as in the ideal situation above, what changes in reality is that while v_n is known v_{n+1} and v_{n+2} are not, but are approximated by $\alpha v_n \ge$

v_{n+1} and $\alpha^2 v_n \geq v_{n+2}$ respectively, where α is fixed by the user. Precisely, B_n^{-1} and x_n are known at the correct precision, but f_n has an approximated interval $[[0,\ v_n + \alpha v_n[[$. To minimize the overhead cost it induces compared to the ideal situation, once we know v_{n+1} (Line 5) we insert some intermediate corrective steps denoted (5.1)-(5.5) thereafter, between Line (5) and Line (6); they require no arithmetic operations.

(5.1) ChangePrec($\sim B_n^{-1},\ v_{n+1}$)
(5.2) ChangePrec($\sim s_n,\ v_n + v_{n+1}$)
(5.3) Tune α if necessary using the new ratio $\frac{v_{n+1}}{v_n}$
(5.4) ChangePrec($\sim x_n,\ v_{n+1} + \alpha v_{n+1}$)
(5.5) ChangePrec($\sim f_{n+1},\ v_{n+1} + \alpha v_{n+1}$)

Most importantly, the remaining Lines (6)-(15) are not impacted since these computations involve now the known v_{n+1} (and not the unknown v_{n+2}): the intervals, and thus costs obtained are the same as in the ideal situation. On the other hand, Lines (1)-(5) are performed as such with an overhead cost. Among them, only Lines (2), (5) have a non negligible cost. At Line (2), B_n^{-1} has approximated interval $[[0,\ \alpha v_n[[$, yielding a cost of $m^2 \mathsf{M}(\alpha v_n)$. At Line (5) x_{n+1} has approximated interval $[[0,\ v_n(\alpha + \alpha^2)[[$, yielding a cost of $L\mathsf{M}(v_n(\alpha(1+\alpha)))$. Thus the overhead cost "ovh_n" at Iteration n is:

$$m^2(\mathsf{M}(\alpha v_n) - \mathsf{M}(v_{n+1})) + L(\mathsf{M}(v_n\alpha(1+\alpha)) - \mathsf{M}(v_{n+1}+v_{n+2})) \quad (33)$$

This quantity depends on the gaps $\alpha v_n - v_{n+1}$ and $\alpha^2 v_n - v_{n+2}$. These gaps increase with n, but, thanks to the tuning of Step (5.3), reasonably at a linear rate:

Assumption 6.2. *The "error gap"* $|\alpha v_n - v_{n+1}| = O(n)$.

Under this assumption it is easy to (crudely) bound $\sum_{n=0}^{\ell+1} \mathrm{ovh}_n$ of Eq. (33) by $(L + m^2)O(N\log(N))$. Being independent on α this is negligible in front of $O(L + m^2)\mathsf{M}(\frac{N}{\alpha-1})$ for $\alpha < 2$. The theorem below wraps up the considerations made above with Eq. (32):

Theorem 6.3. *If Broyden's method has Q-order of convergence α on $B(x^\star, \rho_1) \times B(f'(x^\star), \rho_2)$, then under Assumption 6.1 and 6.2, the cost of computing $x^\star + O(\pi^N)$ is in $O\left((m^2 + L)\right)\mathsf{M}\left(\frac{N}{\alpha-1}\right)$.*

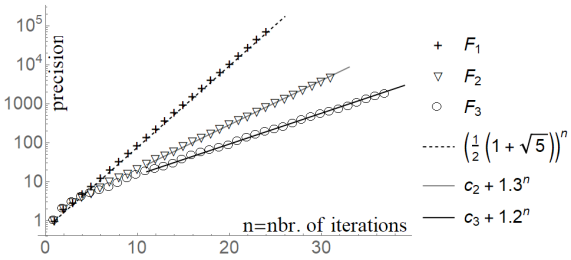
Remark 6.4. Understanding the Q-order of convergence is a major and notoriously difficult problem in the numerical analysis community. Numerical evidence shows it deteriorates with m, and is larger than $2^{1/2m}$ (Theorems 4.7-4.8). Some experiments suggest that taking $\alpha \approx 2^{1/m}$ is not unreasonable. We then get a cost in $O\left((m^2 + L)\mathsf{M}\left(\frac{N}{\alpha-1}\right)\right) \approx O\left((m^2 + L)\mathsf{M}(Nm)\right)$. For comparison, denoting $\omega < 3$ the exponent of the cost of matrix product, the standard analysis of Newton's method for rational fractions would lead to $O((m^\omega + mL)\mathsf{M}(N))$. Consequently, in this setting, for large m, there is little hope that Broyden's method can outperform Newton's when both are available. Remember though other worthwile applications in the paragraph *"Motivations"* in Introduction.

7 NUMERICAL DATA

An implementation of our ultrametric Broyden method in Magma [4] with more data is available at http://xdahan.sakura.ne.jp/broyden20.html. We report the data obtained using the three families of systems, derived from page 36 of [18]. The families are indexed by $t \in \pi O_K$:

- $F_1 = \left((x_1-1)^2+(x_2-1)^2-4-tx_1x_2-t^2x_1,\ (x_1+1)^2+(x_2+1)^2-4-tx_1\right)$ in $K[x_1, x_2]$.
- $F_2 = \left((x_1-1)^2+(x_2-1)^2+(x_3-1)^2-5-t-t^2,\ (x_1+1)^2+(x_2+1)^2+(x_3+1)^2-5-t, 2x_1^2+x_2^2+x_3^2-3-t^2\right)$ in $K[x_1, x_2, x_3]$.
- $F_3 = \left((x_1-1)^2+(x_2-1)^2+(x_3-1)^2+(x_4-1)^2-8-t-t^2,\ (x_1+1)^2+(x_2+1)^2+(x_3+1)^2+(x_4+1)^2-8-t, 2x_1^2+x_2^2+x_3^2+x_4^2-5-t^2, 2x_1x_2+x_3x_2-2x_3x_4+2x_4x_1+3-t^2\right)$ in $K[x_1, x_2, x_3, x_4]$.

Valuation of $f(x_k)$ and numerical estimation of the order of Q-convergence for $\mathbb{Q}[\![T]\!]$ are compiled in the following graphic. For $K = \mathbb{Q}_p$, and $\mathbb{F}_p[\![t]\!]$ with $p = 17$ we experienced the same behaviour.



REFERENCES

[1] E. Bach. Iterative root approximation in p-adic numerical analysis. *J. of Complexity*, 25(6):511–529, 2009.
[2] W. Baur and V. Strassen. The complexity of partial derivatives. *Theoretical computer science*, 22(3):317–330, 1983.
[3] J. Berthomieu, J. Van Der Hoeven, and G. Lecerf. Relaxed algorithms for p-adic numbers. *J. Théor. Nombres Bordeaux*, 23(3):541–577, 2011.
[4] W. Bosma, J. Cannon, and C. Playoust. The Magma algebra system. I. The user language. *J. Symbolic Computation*, 24(3-4):235–265, 1997.
[5] RP Brent and HT Kung. Fast algorithms for manipulating formal power series. *J. of the ACM*, 25(4):581–595, 1978.
[6] CG Broyden. A class of methods for solving nonlinear simultaneous equations. *Mathematics of computation*, 19(92):577–593, 1965.
[7] CG Broyden, JE Dennis Jr, and JJ Moré. On the local and superlinear convergence of quasi-Newton methods. *IMA J. Applied Mathematics*, 12(3):223–245, 1973.
[8] RL Burden and JD Faires. *Numerical analysis*. Brooks/Cole, Cengage Learning, 9th edition, 2011.
[9] X. Caruso. Computations with p-adic numbers. *cours CIRM*, 5(1):1–75, 2017.
[10] JE Dennis, Jr and JJ Moré. Quasi-Newton methods, motivation and theory. *SIAM review*, 19(1):46–89, 1977.
[11] MJ Fischer and LJ Stockmeyer. Fast on-line integer multiplication. *J. Computer and System Sciences*, 9(3):317 – 331, 1974.
[12] AS Fraenkel and Y. Yesha. Complexity of solving algebraic equations. *Information Processing Letters*, 10(4):178 – 179, 1980.
[13] DM Gay. Some convergence properties of Broyden's method. *SIAM J. Numerical Analysis*, 16(4):623–630, 1979.
[14] CT Kelley and EW Sachs. Broyden's method for approximate solution of nonlinear integral equations. *J. Integral Equations*, 9(1):25–43, 1985.
[15] CT Kelley and EW Sachs. Approximate quasi-Newton methods. *Mathematical Programming*, 48(1-3):41–70, 1990.
[16] HT Kung. On computing reciprocals of power series. *Numerische Mathematik*, 22(5):341–348, 1974.
[17] HT Kung and JF Traub. All algebraic functions can be computed fast. *J. of the ACM*, 25(2):245–260, 1978.
[18] G. Lecerf. *Une alternative aux méthodes de réécriture pour la résolution des systèmes algébriques*. PhD thesis, École polytechnique, France, 2001.
[19] JM Martínez. Practical quasi-Newton methods for solving nonlinear systems. *J. Computational and Applied Mathematics*, 124(1):97 – 121, 2000.
[20] JM Ortega and WC Rheinboldt. *Iterative solution of nonlinear equations in several variables*, volume 30. Siam, 1970.
[21] J.-P. Serre. *A course in arithmetic*. Springer GTM 7, 1973.
[22] J. Sherman and WJ Morrison. Adjustment of an inverse matrix corresponding to a change in one element of a given matrix. *The Annals of Mathematical Statistics*, 21(1):124–127, 1950.
[23] J. van der Hoeven. Relax, but don't be too lazy. *J. Symbolic Computation*, 34(6):479 – 542, 2002.
[24] J. van der Hoeven. Newton's method and FFT trading. *J. Symbolic Computation*, 45(8):857–878, 2010.
[25] J. von zur Gathen and J. Gerhard. *Modern computer algebra*. Cambridge University Press, New York, NY, USA, 2003. Second Edition.

Decidability of Membership Problems for Flat Rational Subsets of $\mathrm{GL}(2,\mathbb{Q})$ and Singular Matrices

Volker Diekert
Formale Methoden der Informatik,
Universität Stuttgart
Stuttgart, Germany
diekert@fmi.uni-stuttgart.de

Igor Potapov
Department of Computer Science,
University of Liverpool
Liverpool, United Kingfom
potapov@liverpool.ac.uk

Pavel Semukhin
Department of Computer Science,
University of Oxford
Oxford, United Kingfom
pavel.semukhin@cs.ox.ac.uk

ABSTRACT

This work relates numerical problems on matrices over the rationals to symbolic algorithms on words and finite automata. Using exact algebraic algorithms and symbolic computation, we prove new decidability results for 2×2 matrices over $\mathbb{Q}$. Namely, we introduce a notion of *flat rational sets*: if M is a monoid and $N \le M$ is its submonoid, then flat rational sets of M relative to N are finite unions of the form $L_0 g_1 L_1 \cdots g_t L_t$ where all L_is are rational subsets of N and $g_i \in M$. We give quite general sufficient conditions under which flat rational sets form an effective relative Boolean algebra. As a corollary, we obtain that the emptiness problem for Boolean combinations of flat rational subsets of $\mathrm{GL}(2,\mathbb{Q})$ over $\mathrm{GL}(2,\mathbb{Z})$ is decidable.

We also show a dichotomy for nontrivial group extension of $\mathrm{GL}(2,\mathbb{Z})$ in $\mathrm{GL}(2,\mathbb{Q})$: if G is a f.g. group such that $\mathrm{GL}(2,\mathbb{Z}) < G \le \mathrm{GL}(2,\mathbb{Q})$, then either $G \cong \mathrm{GL}(2,\mathbb{Z}) \times \mathbb{Z}^k$, for some $k \ge 1$, or G contains an extension of the Baumslag-Solitar group $\mathrm{BS}(1,q)$, with $q \ge 2$, of infinite index. It turns out that in the first case the membership problem for G is decidable but the equality problem for rational subsets of G is undecidable. In the second case, decidability of the membership problem is open for every such G. In the last section we prove new decidability results for flat rational sets that contain singular matrices. In particular, we show that the membership problem is decidable for flat rational subsets of $M(2,\mathbb{Q})$ relative to the submonoid that is generated by the matrices from $M(2,\mathbb{Z})$ with determinants $0, \pm 1$ and the central rational matrices.

CCS CONCEPTS

• **Theory of computation → Formal languages and automata theory**; • **Computing methodologies → Symbolic and algebraic algorithms**.

KEYWORDS

membership problem, rational sets, general linear group

Partial support for V. Diekert by the DFG grant DI 435/7, for I. Potapov by the EPSRC grant EP/R018472/1 and for P. Semukhin by the ERC grant AVS-ISS (648701) is greatly acknowledged.

ISSAC '20, July 20–23, 2020, Kalamata, Greece

ACM ISBN 978-1-4503-7100-1/20/07…$15.00
https://doi.org/10.1145/3373207.3404038

ACM Reference Format:
Volker Diekert, Igor Potapov, and Pavel Semukhin. 2020. Decidability of Membership Problems for Flat Rational Subsets of $\mathrm{GL}(2,\mathbb{Q})$ and Singular Matrices. In *International Symposium on Symbolic and Algebraic Computation (ISSAC '20), July 20–23, 2020, Kalamata, Greece.* ACM, New York, NY, USA, 8 pages. https://doi.org/10.1145/3373207.3404038

1 INTRODUCTION

Many problems in the analysis of matrix products are inherently difficult to solve even in dimension two, and most of such problems become undecidable in general starting from dimension three or four. One of these hard questions is the *membership problem* for matrix semigroups: Given $n \times n$ matrices $\{M, M_1, \ldots, M_m\}$, determine whether there exist an integer $k \ge 1$ and $i_1, \ldots, i_k \in \{1, \ldots, m\}$ such that $M = M_{i_1} \cdots M_{i_k}$. In other words, determine whether a matrix belongs to a finitely generated (f.g. for short) semigroup. The membership problem has been intensively studied since 1947 when A. Markov showed in [29] that this problem is undecidable for matrices in $\mathbb{Z}^{6\times 6}$. A natural and important generalization is the *membership problem in rational subsets* of a monoid. Rational sets are those which can be specified by regular expressions. A special case is the problem above: membership in the semigroup generated by the matrices $M_1, \ldots, M_m$. Another difficult question is to decide the *knapsack problem*: "$\exists x_1, \ldots, x_m \in \mathbb{N}\colon M_1^{x_1} \cdots M_m^{x_m} = M$?". Even significantly restricted cases of these problems become undecidable for high dimensional matrices over the integers [6, 26]; and very few cases are known to be decidable, see [3, 7, 12]. The decidability of the membership problem remains open even for 2×2 matrices over integers [11, 14, 21, 25, 33].

Membership in rational subsets of $\mathrm{GL}(2,\mathbb{Z})$ (the 2×2 integer matrices with determinant ± 1) is decidable. Indeed, $\mathrm{GL}(2,\mathbb{Z})$ has a free subgroup of rank 2 and of index 24 by [32]. Hence it is a f.g. virtually free group, and therefore the family of rational subsets forms an effective Boolean algebra [38, 40]. Two recent results which extended the border of decidability for the membership problem beyond $\mathrm{GL}(2,\mathbb{Z})$ were [34, 35]. The first one is in case of the semigroups of 2×2 nonsingular integer matrices, and the second one is in case of $\mathrm{GL}(2,\mathbb{Z})$ extended by integer matrices with zero determinant.

This paper pushes the decidability border even further. First of all, we consider membership problems for 2×2 matrices over the rationals whereas [34, 35] deal only with integer matrices. Since decidability of the rational membership problem is known for $\mathrm{GL}(2,\mathbb{Z})$, we focus on subgroups G of $\mathrm{GL}(2,\mathbb{Q})$ which contain $\mathrm{GL}(2,\mathbb{Z})$.

In Sec. 4 we prove a dichotomy result. In the first case of the dichotomy, G is generated by $\mathrm{GL}(2,\mathbb{Z})$ and central matrices $\left(\begin{smallmatrix} r & 0 \\ 0 & r \end{smallmatrix}\right)$.

In that case G is isomorphic to $\mathrm{GL}(2,\mathbb{Z}) \times \mathbb{Z}^k$ for $k \geq 1$. It can be derived from known results in the literature about free partially commutative monoids and groups that equality test for rational sets in G is undecidable, but the membership problem in rational subsets is still decidable. So, this is the best we can hope for if a group is sitting strictly between $\mathrm{GL}(2,\mathbb{Z})$ and $\mathrm{GL}(2,\mathbb{Q})$, in general.

If such a group G is not isomorphic to $\mathrm{GL}(2,\mathbb{Z}) \times \mathbb{Z}^k$, then our dichotomy states that it contains a Baumslag-Solitar group $\mathrm{BS}(1,q)$ for $q \geq 2$. The Baumslag-Solitar groups $\mathrm{BS}(p,q)$ are defined by two generators a and t with the defining relation $ta^pt^{-1} = a^q$. They were introduced in [4] and widely studied since then. It is fairly easy to see (much more is known) that they have no free subgroup of finite index unless $pq = 0$ [18]. As a consequence, in both cases of the dichotomy, $\mathrm{GL}(2,\mathbb{Z})$ has infinite index in G. Actually, we prove more, namely, if G contains a matrix of the form $\left(\begin{smallmatrix} r_1 & 0 \\ 0 & r_2 \end{smallmatrix}\right)$ with $|r_1| \neq |r_2|$ (which is the second case in the dichotomy), then G contains some $\mathrm{BS}(1,q)$ for $q \geq 2$ which has *infinite* index in G. It is wide open whether the membership to rational subsets of G can be decided in that second case. For example, let $p \geq 2$ be a prime, and let G' be generated by $\left(\begin{smallmatrix} 0 & -1 \\ 1 & 0 \end{smallmatrix}\right)$, $\left(\begin{smallmatrix} 1 & 1 \\ 0 & 1 \end{smallmatrix}\right)$, and $\left(\begin{smallmatrix} 1 & 0 \\ 0 & p \end{smallmatrix}\right)$. In this case $\left(\begin{smallmatrix} p & 0 \\ 0 & p^{-1} \end{smallmatrix}\right)$ also belongs to G'. Due to [5], the matrices $\left(\begin{smallmatrix} 0 & -1 \\ 1 & 0 \end{smallmatrix}\right)$, $\left(\begin{smallmatrix} 1 & 1 \\ 0 & 1 \end{smallmatrix}\right)$, and $\left(\begin{smallmatrix} p & 0 \\ 0 & p^{-1} \end{smallmatrix}\right)$ generate the group $\mathrm{SL}(2,\mathbb{Z}[1/p])$. [1] So G' contains $\mathrm{SL}(2,\mathbb{Z}[1/p])$ as a subgroup. The structure $\mathrm{SL}(2,\mathbb{Z}[1/p])$ is known [39, II.1 Cor. 2] as an amalgam of two copies of $\mathrm{SL}(2,\mathbb{Z})$ over common subgroup of finite index. It is not even known how to decide subgroup membership in such amalgams. Moreover, $\left(\begin{smallmatrix} 1 & 0 \\ 0 & p \end{smallmatrix}\right)$ acts by conjugation on $\mathrm{SL}(2,\mathbb{Z}[1/p])$, and since $\left(\begin{smallmatrix} 1 & 0 \\ 0 & p \end{smallmatrix}\right)$ generates an infinite cyclic group, we have that $G' = \mathrm{SL}(2,\mathbb{Z}[1/p]) \rtimes \mathbb{Z}$. Hence, even if subgroup membership for $\mathrm{SL}(2,\mathbb{Z}[1/p])$ was decidable, then it could still be undecidable in G'. The situation is better for the subgroup $\mathrm{UT}(2,\mathbb{Z}[1/p]) \rtimes \mathbb{Z} \cong Z[1/p] \rtimes \mathbb{Z} \cong \mathrm{BS}(1,p)$ of G' (which is generated by $\left(\begin{smallmatrix} 1 & 1 \\ 0 & 1 \end{smallmatrix}\right)$ and $\left(\begin{smallmatrix} 1 & 0 \\ 0 & p \end{smallmatrix}\right)$) because the subgroup membership is decidable in f.g. metabelian groups [36].[2]

The complicated structures of simple examples of subgroups in $\mathrm{SL}(2,\mathbb{Q})$ and $\mathrm{GL}(2,\mathbb{Q})$ provide strong reasons to believe that the membership in rational sets becomes undecidable for subgroups of $\mathrm{GL}(2,\mathbb{Q})$, in general. The dichotomy result Thm. 4.1 makes that very concrete. It led us in the direction where we came up with a new, but natural subclass of rational subsets. It is the class of *flat rational sets* $\mathrm{Frat}(M,N)$. The new class satisfies surprisingly good properties. $\mathrm{Frat}(M,N)$ is a relative notion where N is a submonoid of M. It consists of all finite unions of the form $L_0g_1L_1\cdots g_tL_t$, where $g_i \in M$ and $L_i \in \mathrm{Rat}(N)$. Of particular interest in our context is the class $\mathrm{Frat}(G,H)$ where H and G are f.g. groups, $\mathrm{Rat}(H)$ forms a Boolean algebra, and G is the commensurator[3] of H. In this case Thm. 3.3 shows that $\mathrm{Frat}(G,H)$ forms a relative Boolean algebra, i.e., it satisfies $L,K \in \mathrm{Frat}(G,H) \implies L \setminus K \in \mathrm{Frat}(G,H)$. Under some mild effectiveness assumptions this means that the emptiness of finite Boolean combinations of sets in $\mathrm{Frat}(G,H)$ can be decided. Thus, we have an abstract general condition to decide such questions for a natural subclass of all rational sets in G where the whole class $\mathrm{Rat}(G)$ need not be an effective Boolean algebra. The immediate application in the present paper concerns $\mathrm{Frat}(\mathrm{GL}(2,\mathbb{Q}),\mathrm{GL}(2,\mathbb{Z}))$, see Thm. 3.3 and Cor. 3.4. For example, $\mathrm{GL}(2,\mathbb{Z}) \times \mathbb{Z}$ appears in $\mathrm{GL}(2,\mathbb{Q})$ and $\mathrm{Rat}(\mathrm{GL}(2,\mathbb{Z}) \times \mathbb{Z})$ is not an effective Boolean algebra. Still the smaller class of flat rational sets $\mathrm{Frat}(\mathrm{GL}(2,\mathbb{Z}) \times \mathbb{Z}, \mathrm{GL}(2,\mathbb{Z}))$ is a relative Boolean algebra. In order to apply Thm. 3.3, we need $\mathrm{Rat}(H)$ to be an effective relative Boolean algebra. It happens to be an effective Boolean algebra for virtually free groups and many other groups. This class includes, for example, all f.g. abelian groups, and it is closed under free products.

The power of flat rational sets is even more apparent in the context of the membership problem for rational subsets of $\mathrm{GL}(2,\mathbb{Q})$. Let $P(2,\mathbb{Q})$ denote the monoid $\mathrm{GL}(2,\mathbb{Z}) \cup \{h \in \mathrm{GL}(2,\mathbb{Q}) \mid |\det(h)| > 1\}$; then Thm. 3.6 states that we can solve the membership problem "$g \in R$?" for all $g \in \mathrm{GL}(2,\mathbb{Q})$ and all $R \in \mathrm{Frat}(\mathrm{GL}(2,\mathbb{Q}),P(2,\mathbb{Q}))$. Thm. 3.6 generalizes the main result in [34].

Let us summarize the statements about groups G sitting between $\mathrm{GL}(2,\mathbb{Z})$ and $\mathrm{GL}(2,\mathbb{Q})$. Our current knowledge is as follows. There is some evidence that membership in rational subsets of G is decidable if and only if G doesn't contain any $\left(\begin{smallmatrix} r_1 & 0 \\ 0 & r_2 \end{smallmatrix}\right)$ where $|r_1| \neq |r_2|$. However, we can always decide the membership problem for all $L \in \mathrm{Frat}(\mathrm{GL}(2,\mathbb{Q}),P(2,\mathbb{Q}))$. Moreover, it might be that such a positive result is close to the border of decidability.

We also consider singular matrices and generalize the main result of [35] as follows. Let g be a singular matrix in $M(2,\mathbb{Q})$ and let P be the submonoid generated by $\left\{\left(\begin{smallmatrix} r & 0 \\ 0 & r \end{smallmatrix}\right) \,\middle|\, r \in \mathbb{N}\right\} \cup \mathrm{GL}(2,\mathbb{Z}) \cup \{h \in M(2,\mathbb{Z}) \mid \det(h) = 0\}$. Then we can decide the membership problem "$g \in R$?" for all $R \in \mathrm{Frat}(M(2,\mathbb{Q}),P)$.

Our paper concentrates on decidability. For the complexity of our algorithms with respect to binary encoding of matrices a trivial upper bound is exponential space. This follows, for instance, from [38]. We conjecture that membership for flat rational subsets of $\mathrm{GL}(2,\mathbb{Q})$ over $\mathrm{GL}(2,\mathbb{Z})$ is in NP and that the emptiness problem for Boolean combinations of such sets is in PSPACE.

The following facts about complexities are known: [20] shows that the *subgroup membership problem* is decidable in polynomial time for matrices from the modular group $\mathrm{PSL}(2,\mathbb{Z})$. In [8], Thm. 5.2 says that membership for rational subsets for $\mathrm{PSL}(2,\mathbb{Z})$ is in NP; and Cor. 5.2 states that the problem "$1 \in \{M_1,\ldots,M_n\}^*$?" is NP-complete for $\mathrm{SL}(2,\mathbb{Z})$.

Note that solving the membership problem for rational sets plays an important role in modern group theory as highlighted for example in [41] and used in [13].

2 PRELIMINARIES

By $M(n,R)$ we denote the ring of $n \times n$ matrices over a commutative ring R, and $\det : M(n,R) \to R$ is the determinant. By $\mathrm{GL}(n,R)$ we mean the group of invertible matrices, that is, the matrices $g \in M(n,R)$ for which $\det(g)$ is a unit in R. By $\mathrm{SL}(n,R)$ we denote the normal subgroup $\det^{-1}(1)$ of $\mathrm{GL}(n,R)$, called the *special linear group*. Explicit calculation for $\mathrm{SL}(2,\mathbb{Z})$ and for special linear groups over rings of p-adic numbers and function fields are e.g.

[1] For the notation $\mathbb{Z}[1/p]$ and some elementary calculations see Sec. 6.

[2] Decidability of membership for rational subsets in $\mathrm{BS}(1,q)$ for $q \geq 2$ was shown only very recently by Cadilhac, Chistikov, and Zetzsche in [10].

[3] The notion of *commensurator* is a standard concept in group theory which includes many more than matrix groups; the formal definition is given in Sec. 2.1.

in [39]. $\mathrm{BS}(p,q)$ denotes the Baumslag-Solitar group $\mathrm{BS}(p,q) = \langle a, t \mid ta^pt^{-1} = a^q\rangle$.

For groups (and more generally for monoids) we write $N \leq M$ if N is a submonoid of M and $N < M$ if $N \leq M$ but $N \neq M$. If M is a monoid, then $Z(M)$ denotes the center of M, that is, the submonoid of elements which commute with all elements in M. A subsemigroup I of a monoid M is an *ideal* if $MIM \subseteq I$.

2.1 Smith normal forms and commensurators

The standard application for all our results is $\mathrm{GL}(2,\mathbb{Q})$, but the results are more general and have the potential to go far beyond. Let $n \in \mathbb{N}$. It is a classical fact from linear algebra that each nonzero matrix $g \in M(n,\mathbb{Q})$ admits a *Smith normal form*. This is a factorization $g = r\,e\,s_q\,f$ such that $r \in \mathbb{Q}^*$ with $r > 0$, $e, f \in \mathrm{SL}(n,\mathbb{Z})$, and $q \in \mathbb{Z}$ where $s_q = \left(\begin{smallmatrix} 1 & 0 \\ 0 & q \end{smallmatrix}\right)$. The matrices e and f in the factorization are not unique, but both the numbers r and q are. The existence and uniqueness of r and s_q are easy to see by the corresponding statement for integer matrices. Clearly, $r^2q = \det(g)$. So, for $g \neq 0$, the sign of $\det(g)$ is determined by the sign of q. It is known that the Smith normal form can be computed in polynomial time [23].

The notion of "commensurator" is well established in group theory. Let H be a subgroup in G, then the *commensurator* of H in G is the set of all $g \in G$ such that $gHg^{-1} \cap H$ has finite index in H. This also implies that $gHg^{-1} \cap H$ has finite index in gHg^{-1}, too. If H has finite index in G, then G is always a commensurator of H because the normal subgroup $N = \bigcap \{gHg^{-1} \mid g \in G\}$ is of finite index in G if and only if G/H is finite.

Moreover, if $H \leq H'$ is of finite index and $H' \leq G' \leq G$ such that G is a commensurator of H, then G' is a commensurator of H'. The notion of a commensurator pops up naturally in our context. Indeed, let $H = \mathrm{SL}(2,\mathbb{Z})$ and write $g \in \mathrm{GL}(2,\mathbb{Q})$ in its Smith normal form $g = r\,e\,s_q\,f$. Then the index of $gHg^{-1} \cap H$ in H is the same as the index of $s_qHs_q^{-1} \cap H$ in H; and every matrix of the form $\left(\begin{smallmatrix} a & b/q \\ qc & d \end{smallmatrix}\right)$ is in $s_qHs_q^{-1}$ if $\left(\begin{smallmatrix} a & b \\ c & d \end{smallmatrix}\right) \in \mathrm{SL}(2,\mathbb{Z})$. Thus, the index of $s_qHs_q^{-1} \cap H$ in H is bounded by the size of the finite group $\mathrm{SL}(n,\mathbb{Z}/q\mathbb{Z})$. For $n = 2$ this size is in $O(q^3)$. It follows that $\mathrm{GL}(2,\mathbb{Q})$ is the commensurator of $\mathrm{SL}(2,\mathbb{Z})$, and hence of $\mathrm{GL}(2,\mathbb{Z})$. In fact, it is known that $\mathrm{GL}(n,\mathbb{Q})$ is the commensurator of $\mathrm{SL}(n,\mathbb{Z})$ for all $n \in \mathbb{N}$, e.g., see [22].

2.2 Rational and recognizable sets

The results in this section are not new. An exception is however Lem. 2.6. We follow the standard notation as in Eilenberg [16]. Let M be any monoid, then $\mathrm{Rat}(M)$ has the following inductive definition using *rational* (aka *regular*) expressions.

(1) $|L| < \infty, L \subseteq M \implies L \in \mathrm{Rat}(M)$.
(2) $L_1, L_2 \in \mathrm{Rat}(M) \implies L_1 \cup L_2, L_1 \cdot L_2, L_1^* \in \mathrm{Rat}(M)$.

For $L \subseteq M$ the set L^* denotes the submonoid of M which is generated by L. The submonoid L^* is also called the *Kleene-star* of L. Note that the definition of $\mathrm{Rat}(M)$ is intrinsic without reference to any generating set. It is convenient to define simultaneously a *basis* $B(L)$ for L (more precisely for a given rational expression): If $|L| < \infty$, then $B(L) = L$. Moreover, $B(L_1 \cup L_2) = B(L_1) \cup B(L_2)$, $B(L_1 \cdot L_2) = B(L_1) \cup B(L_2)$ if both L_1 and L_2 are nonempty, and $B(L_1 \cdot L_2) = \emptyset$ otherwise. Finally, $B(L^*) = B(L) \cup \{1\}$. Since $B(L)$ is finite, L is a subset of the f.g. submonoid $B(L)^*$. Note that $B(L) = \emptyset \iff L = \emptyset$, hence the emptiness problem is decidable for rational subsets of M if, for example, they are given by rational expressions.

Definition 2.1. Let M be a monoid.[4] The *membership problem for rational subsets* is defined as follows: given $g \in M$ and $R \in \mathrm{RAT}(M)$, decide whether $g \in R$.

Definition 2.2. Let C be a family of subsets of M. We say that C is a *relative Boolean algebra* if it is closed under finite unions and $K, L \in C$ implies $K \setminus L \in C$. It is an *effective relative Boolean algebra* if first, every $L \in C$ is given by an effective description and second, for $L, K \in C$ the union $L \cup K$ and the relative complement $K \setminus L$ are computable. If additionally, M belongs to C, then C is called an *(effective) Boolean algebra*.

By definition, a relative Boolean algebra is closed under finite unions, it follows that it is closed under finite intersection, too.

Note that $\mathrm{Rat}(\mathbb{Q})$ is a relative Boolean algebra because every finitely generated subgroup is isomorphic to $\mathbb{Z}$. It is not a Boolean algebra by Prop. 2.4 because $\mathbb{Q} \notin \mathrm{Rat}(\mathbb{Q})$ as $(\mathbb{Q}, +)$ is not f.g.

Proposition 2.3. *The class of monoids M for which $\mathrm{Rat}(M)$ is an effective Boolean algebra satisfies the following properties:*

(1) *It contains only f.g. monoids. (Trivial.)*
(2) *It contains all f.g. free monoids, f.g. free groups, and f.g. abelian monoids [9, 17, 24].*
(3) *It contains all f.g. virtually free groups [38, 40].*
(4) *It is closed under the operation of free product. [37].*

We also use the following well-known fact from [2].

Proposition 2.4. *Let G be a group. If a subgroup H is in $\mathrm{Rat}(G)$, then H is finitely generated.*

The family of *recognizable* subsets $\mathrm{Rec}(M)$ is defined as follows. We have $L \in \mathrm{Rec}(M)$ if and only if there is a homomorphism $\varphi\colon M \to N$ such that $|N| < \infty$ and $\varphi^{-1}\varphi(L) = L$.

The following assertions are well-known and easy to show [16].

(1) Theorem of McKnight [30]: M is finitely generated $\iff$ $\mathrm{Rec}(M) \subseteq \mathrm{Rat}(M)$.
(2) $L, K \in \mathrm{Rat}(M)$ doesn't imply $L \cap K \in \mathrm{Rat}(M)$, in general.
(3) $L \in \mathrm{Rec}(M), K \in \mathrm{Rat}(M) \implies L \cap K \in \mathrm{Rat}(M)$.
(4) Let H be a subgroup of a group G. Then $|G/H| < \infty \iff H \in \mathrm{Rec}(G)$.

The following (well-known) consequence is easy to show.

Corollary 2.5. *Let G be any group and $H \leq G$ be a subgroup of finite index. Then $\{L \cap H \mid L \in \mathrm{Rat}(G)\} = \{L \subseteq H \mid L \in \mathrm{Rat}(G)\}$.*

Cor. 2.5 doesn't hold if H has infinite index in G. For example, it fails for $F_2 \times \mathbb{Z} = F(a,b) \times F(c)$ which does not have the so-called Howson property: there are f.g. subgroups H, K such that $H \cap K$ is not finitely generated.

The assertion of Lem. 2.6 below is not obvious. It was proved first under the assumption that H has finite index in G, [19, 38, 40]. We show that this assumption is not necessary.[5]

[4] If M is not f.g., then we assume that all elements in M have an effective representation, like in $\mathrm{GL}(2,\mathbb{Q})$.
[5] Sénizergues has a proof of Lem. 2.6 using finite transducers, personal communication.

Lemma 2.6. *Let G be any group and $H \leq G$ be a subgroup. Then*

$$\{L \subseteq H \mid L \in \mathrm{Rat}(G)\} = \mathrm{Rat}(H).$$

Moreover, suppose (i) that G is a f.g. group with decidable word problem and (ii) that the question "$g \in H$?" is decidable for $g \in G$. Then for any NFA A with n states and labels in G that accepts $L \subseteq H$, we can effectively construct an NFA A' with n states and labels in H such that A' also accepts L.

Proof. Let $R \subseteq G$ be such that, first, $1 \in R$ and, second, each right coset $Hr \in H\backslash G$ is represented by exactly one $r \in R$.

Let $L \subseteq H$ and $L = L(A)$ for an NFA A with state set Q. Since $G = \langle H \cup R\rangle$ as a monoid and since $1 \in R$ and $1 \in H$ we may assume that all transition are labeled by elements from G having the form sa with $s \in R$ and $a \in H$. Moreover, we may assume that every state p is on some accepting path. Since there are only finitely many transitions there are finite subsets $H' \subseteq H$ and $S \subseteq R$ such that if sa with $s \in R$ labels a transition, then $s \in S$ and $a \in H'$. Moreover, $G' = \langle H' \cup S\rangle$ is a f.g. subgroup $G' \leq G$ such that $L \in \mathrm{Rat}(G')$.

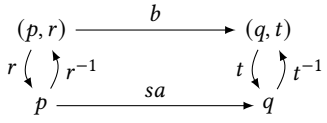
Assume we read from some initial state a word u over the alphabet $H' \cup S$ such that reading that word leads to the state p with $u \in Hr$ for $r \in R$. Then there is some $f \in G$ which leads us to a final state. Thus, $uf \in L(A) \subseteq H$, and therefore $u \in Hf^{-1}$. This means $Hf^{-1} = Hr$ and therefore r doesn't depend on u. It depends on p only: each state $p \in Q$ "knows" its value $r = r(p) \in R$. If u' is any word which we can read from the initial state to p, then $u' \in Hr(p)$. Moreover, if p is any initial or final state, then we have $r(p) = 1$.

This will show that we only need the finite subset R' of R. The set R' contains S and all $r \in R$ such that $Hf_p^{-1} = Hr$ where f_p is the label of a shortest path from a state p to a final state. Let $r = r(p) \in R'$ for $p \in Q$. We introduce exactly one new state (p, r) with transitions $p \xrightarrow{r^{-1}} (p, r)$ and $(p, r) \xrightarrow{r} p$. This does not change the language.

Now for each outgoing transition $p \xrightarrow{sa} q$ with $r = r(p)$ and $t = r(q) \in R'$ define $b \in H$ by the equation $b = rsat^{-1}$. Recall if we read u reaching p, then $ur^{-1} \in H$ and $usat^{-1} \in H$. Therefore, $ur^{-1}rsat^{-1} \in H$ and hence $b \in H$. We add a transition

$$(p, r) \xrightarrow{b} (q, t).$$

This doesn't change the language as $b = rsat^{-1}$ in G and before we added the transition there was a path $(p, r) \xrightarrow{r} p \xrightarrow{sa} q \xrightarrow{t^{-1}} (q, t)$ as can be seen in the following picture:



Now, the larger NFA still accepts L, but the crucial point is that for $u \in L(A)$ we can accept the same element in G by reading just labels from H. Indeed, consider any path $p_0 \xrightarrow{s_1a_1} p_1 \cdots \xrightarrow{s_ka_k} p_k$, where $k \geq 0$ and p_0 is an initial. We claim that the new NFA contains a path labeled by $b_1 \cdots b_k$ with $b_1, \ldots, b_k \in H$ from p_0 to $(p_k, r(p_k))$ such that $b_1 \cdots b_k = s_1a_1 \cdots s_ka_kr(p_k)^{-1}$.

This holds for $k = 0$ because $r(p_0) = 1$ and there is a transition with label 1 from p_0 to $(p_0, 1)$. Let $k \geq 1$. By induction the claim holds for $k-1$. Inspecting the figure above, where $b = b_k$, $sa = s_ka_k$, $(p, r) = (p_{k-1}, r(p_{k-1}))$ and $(q, t) = (p_k, r(p_k))$, we see that the claim holds for k since $r(p_{k-1})^{-1}b_k = s_ka_kr(p_k)^{-1}$; and so:

$$\begin{aligned} b_1 \cdots b_{k-1}b_k &= s_1a_1 \cdots s_{k-1}a_{k-1}r(p_{k-1})^{-1}b_k \\ &= s_1a_1 \cdots s_{k-1}a_{k-1}s_ka_kr(p_k)^{-1}. \end{aligned}$$

We are done, since $r(p_k) = 1$ whenever p_k is final and hence there is a transition with label 1 from $(p_k, 1)$ to p_k.

Now we can remove all original states since they are good for nothing anymore by making $(p, 1)$ initial (resp. final) if and only if p was initial (resp. final). Let us denote the new NFA by A'. Then A' has exactly the same number of states as A.

This shows the non-effective version for all groups G with subgroups H. Finally, in order to make the construction effective it is sufficient that, first, G is f.g. and has a decidable word problem and, second, that the question "$g \in H$?" is decidable for $g \in G$. □

Proposition 2.7. *Let H be a subgroup of finite index in a f.g group G. If the membership problem for rational subsets of H is decidable, then it is decidable for rational subsets of G.*

Proof. Since H is of finite index, there is a normal subgroup N of finite index in G such that $N \leq H \leq G$, [28]. Using the canonical homomorphism from G to G/N we see that H is recognizable. Hence, "$g \in H$?" is decidable. We want to decide "$g \in R$?" for some $R \in \mathrm{Rat}(G)$. Suppose $u_1, \ldots, u_k$ are all representatives of right cosets of H in G. Choose i such that $gu_i^{-1} \in H$. Then $g \in R$ if and only if $gu_i^{-1} \in Ru_i^{-1} \cap H$. Since H is recognizable, we have $Ru_i^{-1} \cap H \in \mathrm{Rat}(G)$. By Lem. 2.6, we have $Ru_i^{-1} \cap H \in \mathrm{Rat}(H)$; and hence we can decide whether $g \in R$. □

3 FLAT RATIONAL SETS

The best situation is when $\mathrm{Rat}(M)$ is an effective Boolean algebra because in this case all decision problems we are studying here are decidable. However, our focus is on matrices over the rational or integer numbers, in which case such a strong assertion is either wrong or not known to be true. Our goal is to search for weaker conditions under which it becomes possible to decide emptiness of finite Boolean combinations of rational sets or (even weaker) to decide membership in rational sets. Again, in various interesting cases the membership problem in rational subsets is either undecidable or not known to be decidable. The most prominent example is the direct product $F_2 \times F_2$ of two free groups of rank 2 in which, due to the construction of Mihailova [31], there exists a finitely generated subgroup with undecidable membership problem.

We introduce a notion of *flat rational sets* and show that the membership problem and (even stronger) the emptiness problem for Boolean combinations of flat rational sets are decidable in $\mathrm{GL}(2, \mathbb{Q})$.

Definition 3.1. Let N be a submonoid of M. We say that $L \subseteq M$ is a *flat rational subset* of M relative to N (or over N) if L is a finite union of languages of the form $L_0g_1L_1 \cdots g_tL_t$ where all $L_i \in \mathrm{Rat}(N)$ and $g_i \in M$. The family of these sets is denoted by $\mathrm{Frat}(M, N)$.

In our applications we use flat rational sets in the following setting: H is a subgroup of G, and G sits inside a monoid M, where $M \setminus G$ is an ideal (possibly empty). For example, $H = \mathrm{GL}(2, \mathbb{Z}) < G \leq \mathrm{GL}(2, \mathbb{Q})$ and $M \setminus G$ is a (possibly empty) semigroup of singular

matrices. In such a situation there is an equivalent characterization of flat rational sets in M with respect to H. Prop. 3.2 shows it can be defined as the family of rational sets when the Kleene-star is restricted to subsets which belong to the submonoid H.

Proposition 3.2. *Let H be a subgroup of G and G be a subgroup of a monoid M such that $M \setminus G$ is an ideal. Then the family $\mathrm{Frat}(M, H)$ is the smallest family $\mathcal{R}$ of subsets of M such that the following holds.*

- *$\mathcal{R}$ contains all finite subsets of M,*
- *$\mathcal{R}$ is closed under finite union and concatenation,*
- *$\mathcal{R}$ is closed under taking the Kleene-star over subsets of H which belong to $\mathcal{R}$.*

Proof. Clearly, all flat rational sets relative to H are contained in $\mathcal{R}$. To prove inclusion in the other direction, we need to show that the family of flat rational subsets of M relative to H (i) contains all finite subsets of M, (ii) is closed under finite union and concatenation, and (iii) is closed under taking the Kleene-star over subsets of H. The first two conditions are obvious. To show (iii), let L be a flat rational set relative to H such that $L \subseteq H$. Recall that L is a finite union of languages $L_0 g_1 L_1 \cdots g_t L_t$, where $\emptyset \neq L_i \in \mathrm{Rat}(H)$ and $g_i \in M$. If $g_i \in M \setminus G$ for some i, then we have $L_0 g_1 L_1 \cdots g_t L_t \setminus G \neq \emptyset$ because $M \setminus G$ is an ideal, and hence $L \not\subseteq H$.

So if $L \subseteq H$, then all $g_i \in G$ and $L \in \mathrm{Rat}(G)$. By Lem. 2.6, L is a rational subset of H, and hence $L^* \in \mathrm{Rat}(H)$. In particular, L^* is flat rational relative to H. □

Theorem 3.3. *Let H be a subgroup of a f.g. group G with decidable word problem such that the following conditions hold:*

- *$\mathrm{Rat}(H)$ is an effective relative Boolean algebra.*[6]
- *G is the commensurator of H, and moreover for a given $g \in G$ we can compute the index of H_g in H.*
- *The membership to H (that is, "$g \in H$?") is decidable.*

Then $\mathrm{Frat}(G, H)$ forms an effective relative Boolean algebra. In particular, given a finite Boolean combination B of flat rational sets of G over H, we can decide the emptiness of B.

Before proving Thm. 3.3 let us first state a consequence.

Corollary 3.4. *Let $B \subseteq \mathrm{GL}(2, \mathbb{Q})$ be a finite Boolean combination of flat rational sets of $\mathrm{GL}(2, \mathbb{Q})$ over $\mathrm{GL}(2, \mathbb{Z})$, then we can decide the emptiness of B.*

Proof. It is a well-known classical fact that $\mathrm{GL}(2, \mathbb{Z})$ is a finitely generated virtually free group, namely, it contains a free subgroup of rank 2 and index 24. Hence $\mathrm{Rat}(\mathrm{GL}(2, \mathbb{Z}))$ is an effective Boolean algebra by [40]. Let G be a f.g. subgroup of $\mathrm{GL}(2, \mathbb{Q})$ that contains B. Clearly, G has a decidable word problem. It is also well-known that $\mathrm{GL}(2, \mathbb{Q})$ is the commensurator subgroup of $\mathrm{GL}(2, \mathbb{Z})$ in $\mathrm{GL}(2, \mathbb{Q})$. Hence G is the commensurator of $\mathrm{GL}(2, \mathbb{Z})$, too. Thus all hypotheses of Thm. 3.3 are satisfied. □

A direct consequence of Cor. 3.4 is that we can decide the membership in flat rational subsets of $\mathrm{GL}(2, \mathbb{Q})$ over $\mathrm{GL}(2, \mathbb{Z})$. However in Sec. 4 we explain why we are far away from knowing how to decide the membership for all rational subsets of $\mathrm{GL}(2, \mathbb{Q})$.

For the proof of Thm. 3.3 we need the following observation.

[6] Recall that this does not imply $H \in \mathrm{Rat}(H)$: possibly H it not f.g.

Lemma 3.5. *Let $L \in \mathrm{Rat}(H)$ and $g \in G$. Recall that*

$$H_g = gHg^{-1} \cap H = \left\{ h \in H \,\middle|\, g^{-1}hg \in H \right\}.$$

Then under the assumptions of Thm. 3.3 we can compute an expression for $g^{-1}(L \cap H_g)g \in \mathrm{Rat}(H)$.

Proof. Since $gHg^{-1} \cap H$ is of finite index in H, we can compute the expression for $L' = L \cap H_g \in \mathrm{Rat}(H_g)$ over a basis $B' \subseteq H_g$ by Lem. 2.6. Now, for any g and $K \in \mathrm{Rat}(H_g)$ we have $g^{-1}K^*g = (g^{-1}Kg)^*$, $g^{-1}(L_1L_2)g = g^{-1}L_1gg^{-1}L_2g$, and $g^{-1}(L_1 \cup L_2)g = g^{-1}L_1g \cup g^{-1}L_2g$. Hence, we simply replace the basis $B' \subseteq H_g$ by $g^{-1}B'g \subseteq H$. This gives a rational expression for $g^{-1}(L \cap H_g)g$ over H. □

Proof of Thm. 3.3. Let $g \in G$ and $K \in \mathrm{Rat}(H)$. First, we claim that we can rewrite $Kg \in \mathrm{Rat}(G)$ as a finite union of languages $g'K'$ with $g' \in G$ and $K' \in \mathrm{Rat}(H)$.

Note that we can compute a set $U_g \subseteq H$ of left-representatives such that $H = \bigcup \left\{ uH_g \,\middle|\, u \in U_g \right\}$. Indeed, by assumption, the membership to H is decidable, and hence the membership to gHg^{-1} and to $H_g = gHg^{-1} \cap H$ is decidable, too. By the second assumption, we can compute the index $k = |H : H_g|$. Thus we can enumerate the elements of H until we find k elements that belong to k different left cosets of H_g. Checking if two elements belong to the same coset is decidable since the membership to H_g can be decided. Thus,

$$\begin{aligned} Kg &= \bigcup \left\{ K \cap uH_g \,\middle|\, u \in U_g \right\} g = \bigcup \left\{ ug\, g^{-1}(u^{-1}K \cap H_g)g \,\middle|\, u \in U_g \right\} \\ &= \bigcup \left\{ g'g^{-1}(gg'^{-1}K \cap H_g)g \,\middle|\, g' \in U_g g \right\}. \end{aligned}$$

Using Lem. 3.5 we obtain $g^{-1}(gg'^{-1}K \cap H_g)g = K' \in \mathrm{Rat}(H)$. This shows the claim.

Let L be a flat rational subset of G, that is, L is equal to a finite union of languages $L_0 g_1 L_1 \cdots g_t L_t$ where all $L_i \in \mathrm{Rat}(H)$. Using the claim, we can write L as a finite union of languages gK with $g \in G$ and $K \in \mathrm{Rat}(H)$. Since membership in H is decidable, we can computably enumerate a set S of all distinct representatives of the right cosets of H, and moreover for each $g \in G$ find a representative $g' \in S$ such that $g \in g'H$. Since $g = g'h$ for some $h \in H$, we can write $gK = g'(hK)$, where $hK \in \mathrm{Rat}(H)$. Therefore, every flat rational set L can be written as a union $L = \bigcup_{i=1}^n g_i K_i$, where $g_i \in S$ and $K_i \in \mathrm{Rat}(H)$. Since $gK_1 \cup gK_2 = g(K_1 \cup K_2)$, we may assume that all g_i in the expression $L = \bigcup_{i=1}^n g_i K_i$ are different.

Now let L and R be two flat rational sets. By the above argument we may assume that $L = \bigcup_{i=1}^n a_i L_i$ and $R = \bigcup_{j=1}^m b_j R_j$, where $a_i, b_j \in S$ and $L_i, R_j \in \mathrm{Rat}(H)$. Then we have $L \setminus R = \bigcup_{i=1}^n \left(a_i L_i \setminus \bigcup_{j=1}^m b_j R_j\right)$. Note that if $a_i \notin \{b_1, \ldots, b_m\}$, then $a_i L_i \setminus \bigcup_{j=1}^m b_j R_j = a_i L_i$, but if $a_i = b_j$ for some j then $a_i L_i \setminus \bigcup_{j=1}^m b_j R_j = a_i(L_i \setminus R_j)$. Since $\mathrm{Rat}(H)$ is an effective relative Boolean algebra, we can compute the rational expression for $L_i \setminus R_j$ in H. Hence we can compute the flat rational expression for $L \setminus R$. □

Below we give one more application of Thm. 3.3. Let $P(2, \mathbb{Q})$ denote the following submonoid of $\mathrm{GL}(2, \mathbb{Q})$ of matrices:

$$P(2, \mathbb{Q}) = \{ h \in \mathrm{GL}(2, \mathbb{Q}) \mid |\det(h)| > 1 \} \cup \mathrm{GL}(2, \mathbb{Z}).$$

Note that $P(2, \mathbb{Q})$ contains all nonsingular matrices from $M(2, \mathbb{Z})$. So, the next theorem is a generalization of the main result in [34].

Theorem 3.6. *For any $g \in \mathrm{GL}(2,\mathbb{Q})$ and for any flat rational subset R of $\mathrm{GL}(2,\mathbb{Q})$ relative to $P(2,\mathbb{Q})$, it is decidable whether $g \in R$.*

Proof. Writing g in Smith normal form, we obtain

$$g = c_r e s_n f = c_r e \left(\begin{smallmatrix} 1 & 0 \\ 0 & n \end{smallmatrix}\right) f,$$

where $c_r = \left(\begin{smallmatrix} r & 0 \\ 0 & r \end{smallmatrix}\right)$ is central, $e, f \in \mathrm{SL}(2,\mathbb{Z})$ and $r \in \mathbb{Q}$. Replacing R by $r^{-1}e^{-1}Rf^{-1}$, we may assume that $g = s_n$ with $0 \neq n \in \mathbb{Z}$. Moreover, by making guesses we may assume that $R = R_0 g_1 R_1 \cdots g_t R_t$ where $R_i \in \mathrm{Rat}(P(2,\mathbb{Q}))$ and each g_i is of the form $g_i = \left(\begin{smallmatrix} r & 0 \\ 0 & r \end{smallmatrix}\right)$ with $0 < r < 1$. Multiplying g and R with some appropriate natural number, we can assume that $g = \left(\begin{smallmatrix} m & 0 \\ 0 & n \end{smallmatrix}\right)$ with $m, n \in \mathbb{N} \setminus \{0\}$ and $R \in \mathrm{Rat}(P(2,\mathbb{Q}))$.

Without restriction we may assume that R is given by a trim NFA $\mathcal{A}$ with state space Q, initial states I and final states F. (Trim means that every state is on some accepting path.) Note that a path in $\mathcal{A}$ accepting g can use transitions with labels from $P(2,\mathbb{Q}) \setminus \mathrm{GL}(2,\mathbb{Z})$ at most $k = \left\lfloor \frac{\log(mn)}{\log t} \right\rfloor$ many times, where

$$t = \min\{\, |\det(h)| \;:\; |\det(h)| > 1 \text{ and } h \text{ appears as a label of a transition in } \mathcal{A} \,\}.$$

Consider a new automaton $\mathcal{B}$ with state space $Q \times \{0,\dots,k\}$, initial states $I \times \{0\}$ and final states $F \times \{0,\dots,k\}$. The transitions of $\mathcal{B}$ are defined as follows:

- for each transition $p \xrightarrow{g} q$ in $\mathcal{A}$ with $g \in \mathrm{GL}(2,\mathbb{Z})$, there is a transition $(p,i) \xrightarrow{g} (q,i)$ in $\mathcal{B}$ for every $i = 0,\dots,k$;
- for every transition $p \xrightarrow{g} q$ in $\mathcal{A}$ with $g \in P(2,\mathbb{Q}) \setminus \mathrm{GL}(2,\mathbb{Z})$, there is a transition $(p,i) \xrightarrow{g} (q,i+1)$ in $\mathcal{B}$ for every $i = 0,\dots,k-1$.

The automaton $\mathcal{B}$ defines a flat rational subset $R' \subseteq R$ over $\mathrm{GL}(2,\mathbb{Z})$ such that $g \in R' \iff g \in R$. So, using Thm. 3.3, we can decide whether $g \in R'$ and hence whether $g \in R$. □

4 DICHOTOMY IN $\mathrm{GL}(2,\mathbb{Q})$

Below we show a dichotomy result. To the best of the authors knowledge the result has not been stated elsewhere. The dichotomy shows that extending our decidability results beyond flat rational sets over $\mathrm{GL}(2,\mathbb{Z})$ seems to be quite demanding.

Theorem 4.1. *Let G be a f.g. group such that $\mathrm{GL}(2,\mathbb{Z}) < G \leq \mathrm{GL}(2,\mathbb{Q})$. Then there are two mutually exclusive cases.*

1. *G is isomorphic to $\mathrm{GL}(2,\mathbb{Z}) \times \mathbb{Z}^k$, with $k \geq 1$, and it does not contain the Baumslag-Solitar group $\mathrm{BS}(1,q)$ for any $q \geq 2$.*
2. *G contains a subgroup which is an extension of infinite index of $\mathrm{BS}(1,q)$ for some $q \geq 2$.*

Proof. Let $H = \mathrm{GL}(2,\mathbb{Z})$. There are two cases. In the first case some finite generating set for G contains only elements from H and from the center $Z(G)$. Since $\mathrm{GL}(2,\mathbb{Z}) \leq G$ we see that $Z(G) \leq \left\{ \left(\begin{smallmatrix} r & 0 \\ 0 & r \end{smallmatrix}\right) \,\middle|\, r \in \mathbb{Q} \right\}$. Moreover, since $\left(\begin{smallmatrix} -1 & 0 \\ 0 & -1 \end{smallmatrix}\right) \in H$, we may assume in the fist case that G is generated by H and f.g. subgroup $Z \leq \left\{ \left(\begin{smallmatrix} r & 0 \\ 0 & r \end{smallmatrix}\right) \,\middle|\, r \in \mathbb{Q} \wedge r > 0 \right\}$. The homomorphism $g \mapsto |\det(g)|$ embeds Z into the torsion free group $\{r \in \mathbb{Q}^* \mid r > 0\}$. Hence, Z is isomorphic to $\mathbb{Z}^k$ for some $k \geq 1$. Since $Z \cap H = \{1\}$, the canonical surjective homomorphism from $Z \times H$ onto G is an isomorphism.

In the second case we start with any generating set and we write the generators in Smith normal form $e \left(\begin{smallmatrix} r & 0 \\ 0 & rq \end{smallmatrix}\right) f$. Since $e, f \in \mathrm{GL}(2,\mathbb{Z})$ and $\mathrm{GL}(2,\mathbb{Z}) < G$, without restriction, the generators are either from $\mathrm{GL}(2,\mathbb{Z})$ or they have the form $\left(\begin{smallmatrix} r & 0 \\ 0 & rq \end{smallmatrix}\right)$ with $r > 0$ and $0 \neq q \in \mathbb{N}$. So, if we are not in the first case, there is at least one generator $s = \left(\begin{smallmatrix} r & 0 \\ 0 & rq \end{smallmatrix}\right)$ where $r > 0$ and $2 \leq q \in \mathbb{N}$.

Let BS be the subgroup of G which is generated by $\left(\begin{smallmatrix} 1 & 0 \\ 1 & 1 \end{smallmatrix}\right)$ and s and $\mathrm{BS}(1,q)$ be the Baumslag-Solitar group with generators b and t such that $tbt^{-1} = b^q$. We have $s \left(\begin{smallmatrix} 1 & 0 \\ 1 & 1 \end{smallmatrix}\right) s^{-1} = \left(\begin{smallmatrix} 1 & 0 \\ 1 & 1 \end{smallmatrix}\right)^q$. Hence, there is a surjective homomorphism $\varphi : \mathrm{BS}(1,q) \to \mathrm{BS}$ such that $\varphi(t) = s$ and $\varphi(b) = \left(\begin{smallmatrix} 1 & 0 \\ 1 & 1 \end{smallmatrix}\right)$. Let us show that φ is an isomorphism. Every element $g \in \mathrm{BS}(1,q)$ can be written in the form $t^k b^x t^n$ where k, x, n are integers. Suppose $\varphi(t^k b^x t^n) = 1$. Then $\left(\begin{smallmatrix} 1 & 0 \\ x & 1 \end{smallmatrix}\right) = \varphi(b^x) = \varphi(t^{-k-n}) = \left(\begin{smallmatrix} r & 0 \\ 0 & rq \end{smallmatrix}\right)^{-k-n}$ is a diagonal matrix. But then $g = t^m$ and $\varphi(g) = s^m = 1$ implies $m = 0$. Hence, φ is an isomorphism and BS is the group $\mathrm{BS}(1,q)$. Moreover, consider any $g \in \mathrm{BS} \cap \mathrm{SL}(2,\mathbb{Z})$. As above $g = s^k \left(\begin{smallmatrix} 1 & 0 \\ 1 & 1 \end{smallmatrix}\right)^x s^m$ with $x, k, m \in \mathbb{Z}$. Since by assumption $\det(g) = 1$ we obtain $m = -k$ and hence $g = \left(\begin{smallmatrix} 1 & 0 \\ q^k x & 1 \end{smallmatrix}\right) \in \left\langle \left(\begin{smallmatrix} 1 & 0 \\ 1 & 1 \end{smallmatrix}\right) \right\rangle$. Therefore $\mathrm{SL}(2,\mathbb{Z}) \cap \mathrm{BS}$ is the infinite cyclic group $\left\langle \left(\begin{smallmatrix} 1 & 0 \\ 1 & 1 \end{smallmatrix}\right) \right\rangle = \mathbb{Z}$, which has infinite index in $\mathrm{SL}(2,\mathbb{Z})$. It follows that G contains an extension of $\mathrm{BS}(1,q)$ of infinite index.

But this is not enough, we need to show that $\mathrm{GL}(2,\mathbb{Z}) \times \mathbb{Z}^k$ cannot contain $\mathrm{BS}(1,q)$, otherwise there is no dichotomy. Actually, we do more: there is no abelian group A such that $\mathrm{BS}(1,q)$ is a subgroup of $\mathrm{GL}(2,\mathbb{Z}) \times A$.

Assume by contradiction that it is. Then there are generators $b = (a,x), t = (s,y) \in \mathrm{GL}(2,\mathbb{Z}) \times A$ such that $tbt^{-1} = b^q$. This implies $(q-1)x = 0$. Since $q \geq 2$, the element x generates a finite subgroup in A. Since b generates an infinite cyclic group, we conclude that $a^m \neq 1$ for all $m \neq 0$. Consider the canonical projection φ of $\mathrm{GL}(2,\mathbb{Z}) \times A$ onto $\mathrm{GL}(2,\mathbb{Z})$ such that $\varphi(b) = a$ and $\varphi(t) = s$. We claim that the restriction of φ to $\langle b, t\rangle$ is injective.

Let $\varphi(g) = 1$ for $g \in \langle b, t\rangle$. As above we write $g = t^k b^z t^n$ with $z, k, n \in \mathbb{Z}$. Then we have $s^k a^z s^n = 1 \in \mathrm{GL}(2,\mathbb{Z})$; and therefore $a^z = s^{-k-n}$. Hence a^z commutes with s. Hence $a^z = s a^z s^{-1} = a^{qz}$. We conclude $a^{(q-1)z} = 1$. Since $a^m \neq 1$ for all $m \neq 0$ and $q \geq 2$ we have $z = 0$. Hence $g = t^m$ for some $m \in \mathbb{Z}$. Since $\varphi(g) = 1$, we know $s^m = 1$. Therefore, $t^m = (s^m, my)$ acts trivially on b. But in $\mathrm{BS}(1,q)$ this happens for $m = 0$, only. This tells us that φ is injective on $\langle b, t\rangle$, and the claim follows.

The above claim implies that $\mathrm{BS}(1,q)$ appears as a subgroup in $\mathrm{GL}(2,\mathbb{Z})$. However, no virtually free group can contain $\mathrm{BS}(1,q)$ by [18][7]; and $\mathrm{GL}(2,\mathbb{Z})$ is virtually free. A contradiction. □

Proposition 4.2. *Let G be isomorphic to $\mathrm{GL}(2,\mathbb{Z}) \times \mathbb{Z}^k$ with $k \geq 1$. Then, the question "$L = R$?" on input $L, R \in \mathrm{Rat}(G)$ is undecidable. However, the question "$g \in R$?" on input $g \in G$ and $R \in \mathrm{Rat}(G)$ is decidable.*

[7] Actually, [18] shows a stronger result. If a Baumslag-Solitar group $\mathrm{BS}(p,q)$ appears in a group G with $pq \neq 0$, then G is not hyperbolic. The result is stronger since all f.g. virtually free groups are hyperbolic.

Proof. The group $\mathrm{GL}(2,\mathbb{Z})$ contains a free monoid $\{a,b\}^*$ of rank 2. Thus, under the conditions above, G contains the free partially commutative monoid $M = \{a,b\}^* \times \{c\}^*$. It is known that the question "$L = R$?" on input $L, R \in \mathrm{Rat}(G)$ is undecidable for M [1].

For the decidability we use the fact that $\mathrm{GL}(2,\mathbb{Z})$ has a free subgroup F of rank two and index 24. By [27] the question "$g \in R$?" is decidable in $F \times \mathbb{Z}^k$. Since $F \times \mathbb{Z}^k$ is of finite index (actually 24) in G, the membership problem in G is decidable by Prop. 2.7. □

Remark 1. Let G be a group extension of $\mathrm{GL}(2,\mathbb{Z})$ inside $\mathrm{GL}(2,\mathbb{Q})$ which is not isomorphic to $\mathrm{GL}(2,\mathbb{Z}) \times \mathbb{Z}^k$ for $k \geq 0$. Then, by Thm. 4.1, the group G contains an infinite extension of $\mathrm{BS}(1,q)$ for $q \geq 2$. By [10] the membership in rational sets of $\mathrm{BS}(1,q)$ is decidable. However, to date it is not clear how to extend this result to infinite extensions of $\mathrm{BS}(1,q)$.

5 SINGULAR MATRICES

In this section we show that the membership problem is decidable for flat rational sets containing singular matrices. This extends the results of [35] which considers only integer matrices.

For $H \in \mathrm{GL}(2,\mathbb{Z})$ and $a \in \mathbb{Z}$ we let

$$M_{ij}(a) = \left\{ \left(\begin{smallmatrix} g_{11} & g_{12} \\ g_{21} & g_{22} \end{smallmatrix}\right) \in H \,\middle|\, g_{ij} = a \right\} \subseteq \mathrm{M}(2,\mathbb{Z}).$$

Throughout we will use Lem. 5.1; for a proof see [15, 35].

Lemma 5.1. *The sets $M_{ij}(a)$ are rational for all i, j and $a \in \mathbb{Z}$.*

Theorem 5.2. *Let P be the submonoid of $M(2,\mathbb{Q})$ which is generated by $\mathrm{GL}(2,\mathbb{Z})$, all central matrices $\left(\begin{smallmatrix} r & 0 \\ 0 & r \end{smallmatrix}\right)$ with $r \in \mathbb{N}$, and all matrices $h \in M(2,\mathbb{Z})$ with $\det(h) = 0$. If $R \subseteq M(2,\mathbb{Q})$ is flat rational over P, then "$g \in R$?" is decidable for singular matrices $g \in M(2,\mathbb{Q})$.*

Proof. Without restriction, R is given by a trim NFA $\mathcal{A}$ over a f.g. submonoid M of $\mathrm{M}(2,\mathbb{Q})$ such that transitions are labeled with elements of H or with matrices rs_q where $q \in \mathbb{N}$ or $r \geq 0$. If $g = 0$ and there is one transition labeled by 0, then we know $g \in R$. For $g \neq 0$ we cannot use any transition labeled by 0. Hence without restriction, if a transition is labeled by a rational number r, then $r > 0$. Using Smith normal form and writing rs_q as a product, in the beginning all transitions are labeled either by a matrix in $\mathrm{GL}(2,\mathbb{Z})$ or by a central matrix $\left(\begin{smallmatrix} r & 0 \\ 0 & r \end{smallmatrix}\right)$ or by $s_0 = \left(\begin{smallmatrix} 1 & 0 \\ 0 & 0 \end{smallmatrix}\right)$.

Since $\det(g) = 0$, the label s_0 must be used at least once. By writing R as a finite union $R_1 \cup R_m$ and guessing the correct j we may assume without restriction that $g \in R_j = R = L_1 s_0 L_2$ where $L_i \in \mathrm{Rat}(M)$. Note that the L_i are just rational, and not assumed to be flat rational. Throughout we use the following equation for $r \in \mathbb{Q}$ and $a, b, c, d \in \mathbb{Z}$:

$$s_0 r \begin{pmatrix} a & b \\ c & d \end{pmatrix} s_0 = s_0 \begin{pmatrix} ra & 0 \\ 0 & 0 \end{pmatrix} s_0 = s_0 ras_0 = ras_0. \tag{1}$$

Now, we perform a Benois-type (cf. [9]) of "flooding-the-NFA".

First Round. More transitions without changing the state set.

(1) For all states p, q of $\mathcal{A}$ consider the subautomaton $\mathcal{B}$ where p is the unique initial and q is the unique final state and where all transitions are labeled by $h \in H$ (all other are removed from $\mathcal{A}$). This defines a rational language $L(p,q) \in \mathrm{Rat}(H)$.
(2) Introduce for all states p, q of $\mathcal{A}$ an additional new transition labeled by $L(p,q)$.
(3) If $g = 0$ and $0 \in L(p,q)$, then accept $g \in R$. After that replace all $L(p,q)$ by $L(p,q) \setminus \{0\}$.
(4) If $1 \in L(p,q)$, where $1 = \left(\begin{smallmatrix} 1 & 0 \\ 0 & 1 \end{smallmatrix}\right)$ is the identity matrix, replace $L(p,q)$ by $L(p,q) \setminus \{1\}$ and add a new transition $p \xrightarrow{1} q$.

After that we may assume that all accepting paths of $\mathcal{A}$ are as follows:

$$p_1 \xrightarrow{L_1} q_1 \xrightarrow{r_1 s_0} p_2 \xrightarrow{L_2} \cdots \xrightarrow{r_k s_0} p_k \xrightarrow{L_k} q_k \tag{2}$$

where $r_i \in \mathbb{Q}$, $r_i > 0$, and $0, 1 \notin L_i$ for all $1 \leq i \leq k$. We may assume without restriction that the transition $p_1 \xrightarrow{L_1} q_1$ is the only transition leaving a unique initial state p_1.

It is convenient to assume that the states are divided into two sets: p-states where outgoing transitions are labeled by rational subsets of H and which lead to q-states; and q-states where outgoing transitions are labeled by rs_0 and lead to p-states. In particular, $p_i \neq q_j$ for all i, j.

Since R is flat over P, there is a constant ρ depending on R such that each accepting path as in (2) uses a transition labeled by $r = \left(\begin{smallmatrix} r & 0 \\ 0 & r \end{smallmatrix}\right)$ with $r \notin \mathbb{N}$ at most ρ times. Splitting R again into a finite union we may assume that all accepting paths have the form

$$q_0 \xrightarrow{r} p_1 \xrightarrow{L_1} q_1 \xrightarrow{r_1 s_0} p_2 \xrightarrow{L_2} \cdots \xrightarrow{r_k s_0} p_k \xrightarrow{L_k} q_k \tag{3}$$

where the $r \in \mathbb{Q}$, $r \neq 0$, $r_i \in \mathbb{N} \setminus \{0\}$, and $0, 1 \notin L_i \in \mathrm{Rat}(M)$. Here, q_0 is a new unique initial state. We choose some $z \in \mathbb{Z}$ such that $rz \in \mathbb{N}$; and we aim to decide $zg \in zR$. The NFA for zR is obtained by making the unique p_1-state initial again, to remove q_0, and to replace all outgoing transitions $q_1 \xrightarrow{r_1 s_0} p_2$ by $q_1 \xrightarrow{zr_1 s_0} p_2$. After that little excursion we are back at a situation as in (2). The difference is that all r_i are positive natural numbers. In order to have $g \in R$, we must have $g \in \mathrm{M}(2,\mathbb{Z})$. So, we can assume that, too.

Phrased differently, without restriction from the very beginning assume $g \in \mathrm{M}(2,\mathbb{Z})$, $\det(g) = 0$, and $\mathcal{A}$ accepts R such that all accepting paths are as in (2) where all $r_i \in \mathbb{N} \setminus \{0\}$.

Let $g = \left(\begin{smallmatrix} g_{11} & g_{12} \\ g_{21} & g_{22} \end{smallmatrix}\right)$. We define a *target* value $t \in \mathbb{N}$ by the greatest common divisor of the numbers in $\{g_{11}, g_{12}, g_{21}, g_{22}\}$.

We keep the following assertion as an invariant. If a transition $q \xrightarrow{rs_0}$ appears in $\mathcal{A}$, then r divides t.

Second Round. As long as possible, do the following.

- Choose a sequence of transitions $q' \xrightarrow{rs_0} p \xrightarrow{L} q \xrightarrow{r's_0} p'$ and an integer $z \in \mathbb{Z}$ such that:
  (1) $z = 0 \iff g = 0$,
  (2) the integer rzr' divides t,
  (3) we have $L \cap M_{11}(z) \neq \emptyset$,
  (4) there is no transition $q' \xrightarrow{rzr'} p'$.
- Introduce an additional transition $q' \xrightarrow{rzr'} p'$.

It is clear that the procedure terminates since for $g \neq 0$ the target t has only finitely many divisors. So, the number of integers r, z, r' such that rzr' divides t is finite for $g \neq 0$. For $g = 0$ we have $z = 0$ and 0 divides the target 0. The accepted language of $\mathcal{A}$ was not changed. But now, every accepting path for g can take short cuts. As a consequence, we may assume that all accepting paths for g have length three: $p_1 \xrightarrow{L_1} q_1 \xrightarrow{rs_0} p_2 \xrightarrow{L_2} q_2$. By guessing such a

sequence of length three, we may assume that the NFA is equal to that path with those four states and where r divides t.

We are ready to check whether $g \in L(\mathcal{A})$. Indeed, we know that each matrix $m \in L(\mathcal{A})$ can be written as $m = f_1 r s_0 f_2$ with $f_k \in L_k \in \mathrm{Rat}(H)$ for $k = 1, 2$. We can write $f_1 r s_0 = r \left(\begin{smallmatrix} a & 0 \\ b & 0 \end{smallmatrix}\right)$ and $s_0 f_2 = \left(\begin{smallmatrix} c & d \\ 0 & 0 \end{smallmatrix}\right)$ where the a, b, c, d depend on the pair (f_1, f_2). Hence, $m = r f s_0 h = r f s_0 s_0 h = r \left(\begin{smallmatrix} a & 0 \\ b & 0 \end{smallmatrix}\right) \left(\begin{smallmatrix} c & d \\ 0 & 0 \end{smallmatrix}\right) = r \left(\begin{smallmatrix} ac & ad \\ bc & bd \end{smallmatrix}\right)$. Remember that $0 \neq r \in \mathbb{Z}$. We make the final tests. We have $g \in R$ if and only if r, L_1, and L_2 allow to have the four values rac, rad, rbc, rbd to be the corresponding g_{ij}. To see this we start with eight tests "$0 \in M_{ij}(0) \cap L_k = \emptyset$?". Now, it is enough to consider entries g_{ij} where $g_{ij} \neq 0$. But then each g_{ij}/r has finitely many divisors $e \in \mathbb{Z}$, only. Thus, a few tests "$M_{ij}(e) \cap L_k = \emptyset$?" suffice to decide $g \in R$. □

Theorem 5.3. *Let P' be the submonoid of $M(2, \mathbb{Q})$ which is generated by $\mathrm{GL}(2, \mathbb{Z})$, all central matrices $\left(\begin{smallmatrix} r & 0 \\ 0 & r \end{smallmatrix}\right)$ with $r \in \mathbb{Q}$, and all matrices $h \in M(2, \mathbb{Z})$ with $\det(h) = 0$. If $R \subseteq M(2, \mathbb{Q})$ is flat rational over P', then we can decide $\left(\begin{smallmatrix} 0 & 0 \\ 0 & 0 \end{smallmatrix}\right) \in R$.*

Note that $P' = P \cdot \left\{ \left(\begin{smallmatrix} r & 0 \\ 0 & r \end{smallmatrix}\right) \,\middle|\, r \in \mathbb{Q} \right\}$ where P is from Thm. 5.2. The proof of Thm. 5.3 is straightforward, details are in [15].

6 GENERATORS OF $\mathrm{SL}(2, \mathbb{Z}[1/p])$

As usual, $\mathbb{Z}[1/p]$ denotes the ring $\{p^n r \in \mathbb{Q} \mid n, r \in \mathbb{Z}\}$. We give a simple proof for the well-known fact that $\mathrm{SL}(2, \mathbb{Z}[1/p])$ is generated by $\left(\begin{smallmatrix} 0 & -1 \\ 1 & 0 \end{smallmatrix}\right)$, $\left(\begin{smallmatrix} 1 & 1 \\ 0 & 1 \end{smallmatrix}\right)$, and $\left(\begin{smallmatrix} p & 0 \\ 0 & p^{-1} \end{smallmatrix}\right)$. We use the following notation: let $\alpha, \beta, \gamma, \delta$ denote elements in $\mathbb{Z}[1/p]$, and a, b, c, d denote elements in $\mathbb{Z}$. Starting with a matrix $\left(\begin{smallmatrix} \alpha & \beta \\ \gamma & \delta \end{smallmatrix}\right)$ we do the following:

(1) Multiply by $\left(\begin{smallmatrix} p^{-1} & 0 \\ 0 & p \end{smallmatrix}\right)$ on the left until we reach $\left(\begin{smallmatrix} \alpha & \beta \\ c & d \end{smallmatrix}\right)$.
(2) Multiply by $\left(\begin{smallmatrix} 0 & -1 \\ 1 & 0 \end{smallmatrix}\right)$, $\left(\begin{smallmatrix} 1 & \pm 1 \\ 0 & 1 \end{smallmatrix}\right)$, and $\left(\begin{smallmatrix} 1 & 0 \\ \pm 1 & 1 \end{smallmatrix}\right)$ until we reach $\left(\begin{smallmatrix} \alpha & \beta \\ 0 & d \end{smallmatrix}\right)$. This is trivial for $|c| = |d|$. In the other case we may assume $|c| > |d|$. Next, transform $\left(\begin{smallmatrix} \alpha & \beta \\ c & d \end{smallmatrix}\right)$ into a matrix of type $\left(\begin{smallmatrix} \alpha & \beta \\ c \pm d & d \end{smallmatrix}\right)$ such that $|c \pm d| < |c|$. Use induction on $|c| + |d|$.
(3) Multiply by $\left(\begin{smallmatrix} p & 0 \\ 0 & p^{-1} \end{smallmatrix}\right)$ on the left until we reach $\left(\begin{smallmatrix} \alpha & b \\ 0 & \delta \end{smallmatrix}\right)$.
(4) Now, $\alpha\delta = 1$. Hence $\alpha = p^m a$ and $\delta = p^n d$ where $\gcd(a, p) = \gcd(d, p) = 1$. Since p is a prime, $m + n = 0$ and $ad = 1$.
(5) WLOG $a = d = 1$ and $m \geq 1$ and hence, $\left(\begin{smallmatrix} \alpha & b \\ 0 & \delta \end{smallmatrix}\right) = \left(\begin{smallmatrix} p^m & b \\ 0 & p^{-m} \end{smallmatrix}\right)$.
(6) Using $\left(\begin{smallmatrix} 1 & \pm 1 \\ 0 & 1 \end{smallmatrix}\right)$ we can add or subtract the lower row $p^m |b|$ times to the upper row. Since $m \geq 1$ we obtain $\left(\begin{smallmatrix} p & 0 \\ 0 & p^{-1} \end{smallmatrix}\right)^m$.

REFERENCES

[1] IJ. J. Aalbersberg and H. J. Hoogeboom. 1989. Characterizations of the Decidability of Some Problems for Regular Trace Languages. *Math. Syst. Th.* 22 (1989), 1–19.
[2] Anatolij V. Anisimow and Franz D. Seifert. 1975. Zur algebraischen Charakteristik der durch kontext-freie Sprachen definierten Gruppen. *Elektron. Inf.-Verarbeit. Kybernetik* 11 (1975), 695–702.
[3] László Babai, Robert Beals, Jin-yi Cai, Gábor Ivanyos, and Eugene M. Luks. 1996. Multiplicative Equations over Commuting Matrices. In *Proc. 7th SODA*. 498–507.
[4] Gilbert Baumslag and Donald Solitar. 1962. Some two-generator one-relator non-Hopfian groups. *Bull. Amer. Math. Soc.* 68 (1962), 199–201.
[5] H. Behr and J. Mennicke. 1968. A presentation of the groups PSL(2, p). *Canadian Journal of Mathematics* 20 (1968), 1432–1438.
[6] P. Bell, V. Halava, T. Harju, J. Karhumäki, and I. Potapov. 2008. Matrix Equations and Hilbert's Tenth Problem. *Int. J. Algebra Comp.* 18 (2008), 1231–1241.
[7] P. Bell, I. Potapov, and P. Semukhin. 2019. On the Mortality Problem: From Multiplicative Matrix Equations to Linear Recurrence Sequences and Beyond. In *Proc. 44th MFCS (LIPIcs)*. 83:1–83:15. https://doi.org/10.4230/LIPIcs.MFCS.2019.83
[8] P. C. Bell, M. Hirvensalo, and I. Potapov. 2017. The identity problem for matrix semigroups in $\mathrm{SL}_2(\mathbb{Z})$ is NP-complete. In *Proc. SODA'17*. SIAM, 187–206.
[9] Michèle Benois. 1969. Parties rationelles du groupe libre. *C. R. Acad. Sci. Paris, Sér. A* 269 (1969), 1188–1190.
[10] Michaël Cadilhac, Dmitry Chistikov, and Georg Zetzsche. 2020. Rational subsets of Baumslag-Solitar groups. To appear: Proc. of the 47th ICALP 2020 in LIPIcs.
[11] J. Cassaigne, V. Halava, T. Harju, and F. Nicolas. 2014. Tighter Undecidability Bounds for Matrix Mortality, Zero-in-the-Corner Problems, and More. *arXiv eprints* abs/1404.0644 (2014).
[12] Émilie Charlier and Juha Honkala. 2014. The freeness problem over matrix semigroups and bounded languages. *Inf. Comp.* 237 (2014), 243–256.
[13] Laura Ciobanu and Murray Elder. 2019. Solutions Sets to Systems of Equations in Hyperbolic Groups Are EDT0L in PSPACE. In *Proc. 46th ICALP (LIPIcs, Vol. 132)*. 110:1–110:15. https://doi.org/10.4230/LIPIcs.ICALP.2019.110
[14] T. Colcombet, J. Ouaknine, P. Semukhin, and J. Worrell. 2019. On Reachability Problems for Low-Dimensional Matrix Semigroups. In *Proc. 46th ICALP (LIPIcs)*. 44:1–44:15. https://doi.org/10.4230/LIPIcs.ICALP.2019.44
[15] Volker Diekert, Igor Potapov, and Pavel Semukhin. 2019. Decidability of membership problems for flat rational subsets of GL(2,Q) and singular matrices. *arXiv eprints* abs/1910.02302 (2019).
[16] S. Eilenberg. 1974. *Automata, Languages, and Machines*. Vol. A. Academic Press.
[17] Samuel Eilenberg and Marcel-Paul Schützenberger. 1969. Rational sets in commutative monoids. *J. Algebra* 13 (1969), 173–191.
[18] S. M. Gersten. 1992. Dehn functions and l_1-norms of finite presentations. In *Algorithms and classification in combinatorial group theory (Berkeley, CA, 1989)*. Math. Sci. Res. Inst. Publ., Vol. 23. 195–224.
[19] Z. Grunschlag. 1999. *Algorithms in Geometric Group Theory*. Ph.D. Dissertation.
[20] Yuri Gurevich and Paul Schupp. 2007. Membership problem for the modular group. *SIAM J. Comput.* 37, 2 (2007), 425–459.
[21] Tero Harju. 2009. Post Correspondence Problem and Small Dimensional Matrices. In *Proc. 13th DLT (LN in Comp. Sci, Vol. 5583)*. 39–46.
[22] J. Hillman. 2007. Commensurators and deficiency. (2007). http://www.maths.usyd.edu.au/u/pubs/publist/preprints/2007/hillman-18.pdf
[23] R. Kannan and A. Bachem. 1979. Polynomial Algorithms for Computing the Smith and Hermite Normal Forms of an Integer Matrix. *SIAM J. Comput.* 8 (1979), 499–507. https://doi.org/10.1137/0208040
[24] S. Kleene. 1956. Representation of events in nerve nets and finite automata. In *Automata Studies*. Number 34 in Annals of Mathematics Studies. 3–40.
[25] S. Ko, R. Niskanen, and I. Potapov. 2018. On the Identity Problem for the Special Linear Group and the Heisenberg Group. In *Proc. 45th ICALP (LIPIcs)*. 132:1–132:15. https://doi.org/10.4230/LIPIcs.ICALP.2018.132
[26] Daniel König, Markus Lohrey, and Georg Zetzsche. 2015. Knapsack and subset sum problems in nilpotent, polycyclic, and co-context-free groups. *arXiv eprints* abs/1507.05145 (2015).
[27] Markus Lohrey and Benjamin Steinberg. 2008. The submonoid and rational subset membership problems for graph groups. *J. Algebra* 320 (2008), 728–755.
[28] Roger Lyndon and Paul Schupp. 2001. *Combinatorial Group Theory*. Springer.
[29] A. Markov. 1947. On certain insoluble problems concerning matrices. *Dok. Akad. Nauk SSSR* 57 (1947), 539–542.
[30] J. D. McKnight. 1964. Kleene quotient theorem. *Pac. J. Math.* (1964), 1343–1352.
[31] K. A. Mihailova. 1958. The occurrence problem for direct products of groups. *Dokl. Akad. Nauk SSSR* 119 (1958), 1103–1105. English translation in: Math. USSR Sbornik, 70: 241–251, 1966.
[32] Morris Newman. 1962. The structure of some subgroups of the modular group. *Illinois J. Math.* 6 (1962), 480–487.
[33] Igor Potapov. 2019. Reachability Problems in Matrix Semigroups. *Dagstuhl Reports* 9 (2019), 95–98. https://doi.org/10.4230/DagRep.9.3.83
[34] Igor Potapov and Pavel Semukhin. 2017. Decidability of the Membership Problem for 2×2 integer matrices. In *Proc. 28th SODA*. 170–186.
[35] Igor Potapov and Pavel Semukhin. 2017. Membership Problem in GL(2, $\mathbb{Z}$) Extended by Singular Matrices. In *Proc. 42nd MFCS*. 44:1–44:13.
[36] Nikolay S. Romanovskiĭ. 1974. Some algorithmic problems for solvable groups. *Algebra i Logika* 13 (1974), 26–34, 121.
[37] Jacques Sakarovitch. 1992. The "last" decision problem for rational trace languages. In *Proc. LATIN'92 (LN in Comp. Sci, Vol. 583)*, I. Simon (Ed.). 460–473.
[38] Géraud Sénizergues. 1996. On the rational subsets of the free group. *Acta Inf.* 33 (1996), 281–296.
[39] Jean-Pierre Serre. 1980. *Trees*. Springer.
[40] Pedro V. Silva. 2002. Recognizable subsets of a group: finite extensions and the abelian case. *Bulletin EATCS* 77 (2002), 195–215.
[41] P. V. Silva. 2017. An Automata-Theoretic Approach to the Study of Fixed Points of Endomorphisms. In *Algorithmic and Geometric Topics Around Free Groups and Automorphisms*, J. González-Meneses, M. Lustig, and E. Ventura (Eds.). Birkhäuser.

On the Apolar Algebra of a Product of Linear Forms

Michael DiPasquale
Colorado State University
michael.dipasquale@colostate.edu

Zachary Flores
Colorado State University
flores@math.colostate.edu

Chris Peterson
Colorado State University
peterson@math.colostate.edu

ABSTRACT

Apolarity is a important tool in commutative algebra and algebraic geometry which studies a form, f, by the action of polynomial differential operators on f. The quotient of all polynomial differential operators by those which annihilate f is called the *apolar algebra* of f. In general, the apolar algebra of a form is useful for determining its Waring decomposition, which consists of writing the form as a sum of powers of linear forms with as few summands as possible. In this article we study the apolar algebra of a product of linear forms, which generalizes the case of monomials and connects to the geometry of hyperplane arrangements. In the first part of the article we provide a bound on the Waring rank of a product of linear forms under certain genericity assumptions; for this we use the defining equations of so-called star configurations due to Geramita, Harbourne, and Migliore. In the second part of the article we use the computer algebra system Bertini, which operates by homotopy continuation methods, to solve certain rank equations for catalecticant matrices. Our computations suggest that, up to a change of variables, there are exactly six homogeneous polynomials of degree six in three variables which factor completely as a product of linear forms defining an irreducible multi-arrangement and whose apolar algebras have dimension six in degree three. As a consequence of these calculations, we find six cases of such forms with cactus rank six, five of which also have Waring rank six. Among these are products defining subarrangements of the braid and Hessian arrangements.

CCS CONCEPTS

• **Computing methodologies** → *Special-purpose algebraic systems*; **Hybrid symbolic-numeric methods**; • **Applied computing** → **Mathematics and statistics**; • **Mathematics of computing** → **Solvers**.

KEYWORDS

Apolar algebras, Waring rank, hyperplane arrangements, tensor decomposition, numerical algebraic geometry

ACM Reference Format:
Michael DiPasquale, Zachary Flores, and Chris Peterson. 2020. On the Apolar Algebra of a Product of Linear Forms. In *International Symposium on Symbolic and Algebraic Computation (ISSAC '20), July 20–23, 2020, Kalamata, Greece.* ACM, New York, NY, USA, 8 pages. https://doi.org/10.1145/3373207.3404014

ISSAC '20, July 20–23, 2020, Kalamata, Greece

ACM ISBN 978-1-4503-7100-1/20/07…$15.00
https://doi.org/10.1145/3373207.3404014

1 INTRODUCTION

Given a homogeneous polynomial f of degree d, the *apolar algebra* R_f is the ring of polynomial differential operators modulo those which annihilate f. This algebra has been studied for a variety of reasons; in particular the apolar algebra of a form of degree d is always an Artinian Gorenstein algebra with socle degree d and every Artinian Gorenstein algebra with socle degree d can be represented as the apolar algebra of a form of degree d. This explicit correspondence, via the apolar algebra, between forms of degree d and Artinian Gorenstein algebras with socle degree d is well exposited by Iarrabino and Kanev in [10]. The apolar algebra of a homogeneous polynomial f of degree d is also key to studying the *Waring rank* of f – this is the smallest integer r for which there exist linear forms $\ell_1, \ldots, \ell_r$ so that $f = \ell_1^d + \cdots + \ell_r^d$ (we call such a representation a *Waring decomposition*). The Waring rank often depends on the field chosen – in this note we will work over an algebraically closed field. Note that homogeneous polynomials of degree d correspond to supersymmetric d-dimensional tensors and that the d^{th} power of a linear form corresponds to a rank 1 supersymmetric d-dimensional tensor. Through this correspondence, Waring rank connects to (supersymmetric) tensor rank and Waring decomposition to (supersymmetric) tensor decomposition.

In this note we study the apolar algebra of a form f of degree d which can be written as a product of d, not necessarily distinct, linear forms. Such forms correspond geometrically to hyperplane arrangements (in the case of distinct linear forms) and hyperplane multi-arrangements (in the case of non-distinct linear forms). To simplify exposition, we conflate a multi-arrangement with its defining equation. That is, if we refer to the Waring rank of a multi-arrangement, we mean the Waring rank of its defining equation. Our inspiration for studying this problem stems largely from the thesis of Max Wakefield [16], where several questions are posed about apolar algebras of multi-arrangements. In particular, we study when the apolar algebra of a multi-arrangement is a complete intersection.

If the apolar algebra of a form is a complete intersection, it is often easier to compute its Waring rank. Two important classes of examples (all multi-arrangements) serve to illustrate this point. The first is the case of a monomial, whose apolar algebra is generated by powers of variables. The Waring rank of monomials over the field of complex numbers is completely determined in [4]. The second class is when f is the fundamental skew invariant of a complex reflection group W, which is the product of the linear forms defining the pseudo-reflections of W. In this case the apolar algebra R_f is isomorphic to the ring of covariants of W [11, Chapter 26], which is the quotient of the polynomial ring by the ideal generated by invariants of W. This is a complete intersection since the ring of

invariants is itself a polynomial ring by the celebrated Chevalley-Shephard-Todd theorem. In [14], Teitler and Woo determine the Waring rank of (and a Waring decomposition of) the fundamental skew invariant of a complex reflection arrangement under some mild conditions.

Following a section providing preliminary background material, we briefly discuss *reducible* arrangements, which are arrangements that can be written as a product of lower dimensional arrangements. In Section 4 we make use of the defining equations of *star configurations* determined by Geramita, Harbourne, and Migliore [7] to give a lower bound on the initial degree of the apolar algebra of a generic arrangement (Proposition 4.10). We give two corollaries to Proposition 4.10 – the first is a lower bound on the size of a generic arrangement whose apolar ideal is a complete intersection and the second is a lower bound on the Waring rank of a generic arrangement. Section 5 gives a case study of six lines in $\mathbb{P}^2$. In particular, we use the numerical computer algebra system Bertini [3] to compute what we suspect is a complete list of irreducible multi-arrangements consisting of six lines (counting multiplicity) and annihilated by at least three cubics. We record this list in Conjecture/Theorem* 5.1 (the asterix indicates this is a computational result which can, in theory, be turned into a theorem by numerical certification). This leads to what we expect is a complete list of irreducible multi-arrangements consisting of six lines that have cactus rank equal to six (all but one of these also have Waring rank equal to six). Macaulay2 [8], Sage [15], and Bertini scripts we used to find this list and check the resulting Waring ranks can be found under the Research tab at https://midipasq.github.io/. The final section of the paper provides closing comments and gives suggestions for further research.

2 PRELIMINARIES

Let $\mathbb{K}$ be an algebraically closed field of characteristic zero and put $R = \mathbb{K}[X_0, \ldots, X_n]$. Let $S = \mathbb{K}[x_0, \ldots, x_n]$ be the R-module defined by R acting on S via partial differentiation. That is, if $f \in S$ and $\Phi \in R$,

$$\Phi \circ f = \Phi\left(\frac{\partial}{\partial x_0}, \ldots, \frac{\partial}{\partial x_n}\right) f.$$

This is known as the *apolar* action of R on S. The expository article of Geramita [6] is an excellent introduction to applications of apolarity, the book of Iarrabino and Kanev [10] can be used to go into more detail, and the article of De Paris [5] gives a recent summary of apolarity and tensor rank.

Given a form $f \in S$, the *apolar ideal* of f is

$$\mathrm{Ann}_R(f) = \{\Phi \in R : \Phi \circ f = 0\}.$$

We write $R_f = R/\mathrm{Ann}_R(f)$; this is the *apolar algebra* of f. The apolar algebra R_f is a graded Artinian Gorenstein algebra, and every graded Artinian Gorenstein algebra arises in this way [10, Lemma 2.12].

Now suppose $f \in S_d$ (where S_d denotes the degree d forms in S). A *Waring decomposition* of f is a decomposition $f = c_1\ell_1^d + \cdots + c_k\ell_k^d$, where $\ell_1, \ldots, \ell_k$ are linear forms and $c_1, \ldots, c_k \in \mathbb{K}$ (we do not strictly need $c_1, \ldots, c_k$ since $\mathbb{K}$ is algebraically closed, but it will be useful for us to consider these). The smallest number of linear forms needed in a Waring decomposition of f is the *Waring rank* of f. The following lemma relates the apolarity action and Waring decompositions. See [10, Lemma 1.15] for a proof. In what follows, we say a linear form $\ell = \sum_{i=0}^n a_i x_i \in \mathbb{K}[x_0, \ldots, x_n]$ is *dual* to the point $P = [a_0 : \cdots : a_n] \in \mathbb{P}^n_{\mathbb{K}}$. Any non-zero constant multiple of ℓ is of course dual to the same point P.

Lemma 2.1 (Apolarity Lemma). *Let $f \in S = \mathbb{K}[x_0, \ldots, x_n]$ be a form of degree d, $X = \{P_1, \ldots, P_k\} \subset \mathbb{P}^n_{\mathbb{K}}$ a set of points, and $I_X \subset R$ its corresponding ideal. Write $\ell_1, \ldots, \ell_k$ for linear forms in S dual to the points $P_1, \ldots, P_k$. Then $f = c_1\ell_1^d + \ldots + c_k\ell_k^d$ for some constants $c_1, \ldots, c_k$ if and only if $I_X \subset Ann_R(f)$.*

From the apolarity lemma we see that the Waring rank of a form is the same as the minimum degree of a zero-dimensional radical ideal contained in its apolar ideal. A related notion is the *cactus rank* of a form; this is the minimum degree of a zero-dimensional saturated ideal contained in its apolar ideal (we will see this notion in Section 5).

We will focus on forms $f \in S = \mathbb{K}[x_0, \ldots, x_n]$ which decompose as a product of (not necessarily distinct) linear forms as $f = \ell_1^{m_1} \cdots \ell_k^{m_k}$. If $g \in S$, write $V(g)$ for the set of points in $\mathbb{K}^{n+1}$ at which g vanishes. A natural geometric object to attach to the product $f = \ell_1^{m_1} \cdots \ell_k^{m_k}$ is the *multi-arrangement* $(\mathcal{A}, \mathbf{m})$ where $\mathcal{A} = \cup_{i=1}^k V(\ell_i)$ is the union of the hyperplanes $V(\ell_i) \subset \mathbb{K}^{n+1}$ and $\mathbf{m}$ is a function which assigns to each hyperplane $H \in \mathcal{A}$ the integer $\mathbf{m}(H)$, where $\mathbf{m}(H)$ is the power to which the corresponding linear form appears in f. We put $|\mathbf{m}| = \sum_H \mathbf{m}(H)$, which is the degree of the polynomial f. If $\mathbf{m}(H) = 1$ for all $H \in \mathcal{A}$ we will say $(\mathcal{A}, \mathbf{m})$ is a *simple* arrangement and write $\mathcal{A}$ instead of $(\mathcal{A}, \mathbf{m})$. Given a multi-arrangement $(\mathcal{A}, \mathbf{m})$ we define $Q(A, \mathbf{m}) := \prod_{H \in \mathcal{A}} \alpha_H^{\mathbf{m}(H)}$, where α_H is a choice of linear form vanishing on H. If $\mathcal{A}$ is simple then we write $Q(\mathcal{A})$ for the product $\prod_{H \in \mathcal{A}} \alpha_H$. We call $Q(\mathcal{A}, \mathbf{m})$ and $Q(\mathcal{A})$ the *defining polynomial* of the multi-arrangement and arrangement, respectively. Moreover we write $|\mathcal{A}|$ for the number of hyperplanes in $\mathcal{A}$, so that if $f = Q(\mathcal{A}, \mathbf{m})$, then $|\mathcal{A}|$ is the number of distinct linear factors of f. For simplicity, throughout this note we will conflate a multi-arrangement or arrangement with its defining polynomial. For instance, by "the apolar algebra of an arrangement" we will mean the apolar algebra of its defining equation.

If $\mathcal{A}_1 = \cup_{i=1}^s G_i \subset V \cong \mathbb{K}^n$ and $\mathcal{A}_2 = \cup_{j=1}^t H_j \subset W \cong \mathbb{K}^m$ are two simple arrangements, then the *product* of $\mathcal{A}_1$ and $\mathcal{A}_2$ is defined by

$$\mathcal{A}_1 \times \mathcal{A}_2 = \left(\cup_{i=1}^s G_i \times W\right) \cup \left(V \times \cup_{j=1}^t H_j\right) \subset V \times W$$

If $(\mathcal{A}_1, \mathbf{m}_1)$ and $(\mathcal{A}_2, \mathbf{m}_2)$ are multi-arrangements, the product multi-arrangement $(\mathcal{A}_1 \times \mathcal{A}_2, \mathbf{m})$ satisfies $\mathbf{m}(H \times W) = \mathbf{m}(H)$ if $H \in \mathcal{A}_1$ and $\mathbf{m}(V \times G) = \mathbf{m}(G)$ if $G \in \mathcal{A}_2$. Following [12], we will say that a simple arrangement $\mathcal{A}$ is *reducible* if, after a change of coordinates, $\mathcal{A} = \mathcal{A}_1 \times \mathcal{A}_2$ for some simple arrangements $\mathcal{A}_1$ and $\mathcal{A}_2$. Otherwise we say that $\mathcal{A}$ is *irreducible*.

Suppose $\mathcal{A} \subset \mathbb{K}^n$ is a reducible arrangement and $Q(\mathcal{A})$ is its defining polynomial. Then there is a change of variables so that $\mathcal{A} = \mathcal{A}_1 \times \mathcal{A}_2$, where $\mathcal{A}_1 \subset \mathbb{K}^s$ and $\mathcal{A}_2 \subset \mathbb{K}^t$ for some positive integers s, t satisfying $s + t = n$. Put $S_1 = \mathbb{K}[x_1, \ldots, x_s]$ and $S_2 = \mathbb{K}[y_1, \ldots, y_t]$. Then, under this change of variables, $Q(\mathcal{A}) = Q(\mathcal{A}_1)Q(\mathcal{A}_2)$. Algebraically, the defining polynomials of reducible

arrangements are those which, after an appropriate change of variables, split as a product of two defining polynomials in disjoint sets of variables.

In this note we only consider hyperplane arrangements all of whose hyperplanes pass through the origin (these are called *central* arrangements). Hence we will freely pass between a central arrangement in $\mathbb{K}^{n+1}$ and its natural quotient in $\mathbb{P}^n$ – this does not affect the algebra.

3 PRODUCTS OF ONE AND TWO DIMENSIONAL ARRANGEMENTS

In this section we observe that if $(\mathcal{A}, \mathbf{m})$ is reducible, so $(\mathcal{A}, \mathbf{m}) = (\mathcal{A}_1, \mathbf{m}_1) \times (\mathcal{A}_2, \mathbf{m}_2)$ after a change of variables, then $R_f \cong R_{f_1} \otimes_{\mathbb{K}} R_{f_2}$, where $f = Q(\mathcal{A}, \mathbf{m})$, $f_1 = Q(\mathcal{A}_1, \mathbf{m}_1)$, and $f_2 = Q(\mathcal{A}_2, \mathbf{m}_2)$. Our observation hinges on the following proposition. We suspect this is well-known but we include a proof since we were not able to find one in the literature.

Proposition 3.1. *Suppose s and t are positive integers, $f \in S_1 = \mathbb{K}[x_1, \dots, x_s]$ and $g \in S_2 = \mathbb{K}[y_1, \dots, y_t]$. Put $S = S_1 \otimes_{\mathbb{K}} S_2$. We write R_1, R_2, and R for the polynomial rings dual to S_1, S_2, and S. Then*

(1) *$R_{fg} \cong (R_1)_f \otimes_{\mathbb{K}} (R_2)_g$ and*
(2) *$Ann_R(fg) = Ann_{R_1}(f) \otimes_{\mathbb{K}} R_2 + Ann_{R_2}(g) \otimes_{\mathbb{K}} R_1$*

Remark 3.2. *The tensor product ring $S = S_1 \otimes_{\mathbb{K}} S_2$ is isomorphic as a ring to the polynomial ring $\mathbb{K}[x_1, \dots, x_s, y_1, \dots, y_t]$ via the map $x_i \otimes y_j \to x_i y_j$ (extended linearly). This is because the polynomials are in different sets of variables. Thus there is no harm in regarding tensors in S and R as multiplication in their corresponding polynomial rings; we do this in the proof of Proposition 3.1.*

Proof. Since $\mathrm{Ann}_{R_1}(f)R_2 + \mathrm{Ann}_{R_2}(g)R_1$ is the kernel of the natural map from R to $(R_1)_f \otimes_{\mathbb{K}} (R_2)_g$, it is clear that (1) and (2) are equivalent. We prove (2).

Suppose $\Phi = \sum_{\alpha,\beta} c_{\alpha,\beta} X^\alpha Y^\beta \in R$, where $\alpha = (\alpha_0, \cdots, \alpha_s), \beta = (\beta_0, \dots, \beta_t), X^\alpha = X_0^{\alpha_0} \cdots X_s^{\alpha_s}$, $Y^\beta = Y_0^{\beta_0} \cdots Y_t^{\beta_t}$, and $c_{\alpha,\beta} \in \mathbb{K}$. Then

$$\Phi \circ (fg) = \sum_{\alpha,\beta} c_{\alpha,\beta} \frac{\partial f}{\partial x^\alpha} \frac{\partial g}{\partial y^\beta}.$$

Similarly, if $\varphi_1 \in R_1$ and $\varphi_2 \in R_2$, then $\varphi_1\varphi_2 \circ fg = (\varphi_1 \circ f)(\varphi_2 \circ g)$. From this observation it is clear that $\mathrm{Ann}_{R_1}(f)R_2 + \mathrm{Ann}_{R_2}(g)R_1 \subseteq \mathrm{Ann}_R(fg)$.

We prove that $\mathrm{Ann}_R(fg) \subseteq \mathrm{Ann}_{R_1}(f)R_2 + \mathrm{Ann}_{R_2}(g)R_1$. For this we consider several maps: $\alpha_f : R_1 \to S_1$ given by $\varphi \to \varphi \circ f$, $\alpha_g : R_2 \to S_2$ by $\varphi \to \varphi \circ g$, the tensor product maps $\alpha'_f := \alpha_f \otimes_{\mathbb{K}} \mathrm{id}_{R_2} : R_1 \otimes_{\mathbb{K}} R_2 \to S_1 \otimes_{\mathbb{K}} R_2$ and $\alpha'_g := \mathrm{id}_{S_1} \otimes_{\mathbb{K}} \alpha_g : S_1 \otimes_{\mathbb{K}} R_2 \to S_1 \otimes_{\mathbb{K}} S_2$. By the above observations, $\mathrm{Ann}_R(fg) = \ker(\alpha'_g \circ \alpha'_f)$.

Suppose $\Phi = \sum_{\alpha,\beta} c_{\alpha,\beta} X^\alpha Y^\beta \in \mathrm{Ann}_R(fg)$. Then

$$\Phi \circ fg = \sum_{\alpha,\beta} c_{\alpha,\beta} \frac{\partial f}{\partial x^\alpha} \frac{\partial g}{\partial y^\beta} = 0. \tag{1}$$

Suppose the monomial x^γ appears in $\frac{\partial f}{\partial x^\alpha}$ with coefficient $d_{\gamma,\alpha} \in \mathbb{K}$. Equating coefficients of x^γ in Equation (1) yields

$$x^\gamma \sum_{\alpha,\beta} d_{\gamma,\alpha} c_{\alpha,\beta} \frac{\partial g}{\partial y^\beta} = 0.$$

It follows that $\sum_{\alpha,\beta} d_{\gamma,\alpha} c_{\alpha,\beta} Y^\beta \in \ker(\alpha'_g) = \mathrm{Ann}_{R_2}(g)$. Thus

$$\alpha'_f(\Phi) = \sum_{\alpha,\beta} c_{\alpha,\beta} \frac{\partial f}{\partial x^\alpha} Y^\beta \in \mathrm{Ann}_{R_2}(g)\alpha_f(R_1).$$

Notice that

$$\alpha'_f(\mathrm{Ann}_{R_1}(f)R_2 + \mathrm{Ann}_{R_2}(g)R_1) = \mathrm{Ann}_{R_2}(g)\alpha_f(R_1).$$

Since $\alpha'_f(\mathrm{Ann}_R(fg)) \subseteq \mathrm{Ann}_{R_2}(g)\alpha_f(R_1)$ and $\ker(\alpha'_f) = \mathrm{Ann}_{R_1}(f)R_2$, we have $\mathrm{Ann}_R(fg) \subseteq \mathrm{Ann}_{R_1}(f)R_2 + \mathrm{Ann}_{R_2}(g)R_1$, as desired. □

Corollary 3.3. *Suppose $S \cong S_1 \otimes_{\mathbb{K}} \cdots \otimes_{\mathbb{K}} S_k$, where S_i is a polynomial ring in one or two variables for $i = 1, \dots, k$. If a form $f \in S$ factors as $f = f_1 \cdots f_k$ where $f_i \in S_i$ for $i = 1, \dots, k$, then $Ann_R(f)$ is a complete intersection.*

Proof. It is well known that the apolar algebra of a homogeneous polynomial in one or two variables is a complete intersection (since Gorenstein coincides with complete intersection in one and two variables). The corollary follows directly from this fact and Proposition 3.1. □

Remark 3.4. *Over an algebraically closed field it is clear that the factors $f_1, \dots, f_k$ in Corollary 3.3 are in fact products of linear forms.*

Remark 3.5. *Corollary 3.3 shows that the apolar algebra of a multi-arrangement which is a product of one and two dimensional arrangements is a complete intersection. One may ask the reverse question: if the apolar algebra of $Q(\mathcal{A}, \mathbf{m})$ is a complete intersection for every choice of multiplicity $\mathbf{m}$, is $\mathcal{A}$ necessarily a product of one and two dimensional arrangements? A similar question has an affirmative answer: in [1] it is proved that if the module of multi-derivations $D(\mathcal{A}, \mathbf{m})$ is free for every multiplicity $\mathbf{m}$, then $\mathcal{A}$ is indeed a product of one and two dimensional arrangements.*

4 GENERIC ARRANGEMENTS

In this section we derive a lower bound on the initial degree of the apolar ideal of a *generic* arrangement $\mathcal{A} \subset \mathbb{P}^n$ with at least $n+1$ hyperplanes (Proposition 4.10). All arrangements in this section are simple arrangements.

Definition 4.1. An arrangement in $\mathbb{P}^n$ is generic if the intersection of any k of its hyperplanes has codimension $\min\{k, n+1\}$.

In preparation we give several lemmas and definitions. Given a form $G \in R$, the *gradient of G* is the vector $\nabla G := \left(\frac{\partial G}{\partial X_0}, \dots, \frac{\partial G}{\partial X_n}\right)$.

Lemma 4.2. *Suppose $g \in S$ is a homogeneous polynomial and write $f = \ell g$ for some linear form ℓ. Let $F \in R$ be homogeneous of degree $d \geq 1$. Then, if we abuse notation and write ℓ for the corresponding linear form in R, we have*

$$F \circ f = (\nabla F \cdot \nabla \ell) \circ g + \ell\,(F \circ g).$$

(Here $\nabla F \cdot \nabla \ell$ denotes the dot product.) In particular, if $f = \ell_1 \ell_2 \cdots \ell_t$ is a product of $t \geq n$ linear forms, n of which are linearly independent, then there is an $\ell \in \{\ell_1, \dots, \ell_t\}$ such that $\nabla F \cdot \nabla \ell$ is nonzero.

Proof. Write $\ell = a_0x_0 + \cdots + a_nx_n$. First, let F be a monomial of degree d, say $F = X_{i_1}^{d_1} \cdots X_{i_t}^{d_t}$, where $d_1, \ldots, d_t$ are positive. Then it is easy to see that $F \circ f$ is given by

$$\left(\sum_{j=1}^{t} d_j a_j X_{i_1}^{d_1} \cdots X_{i_j}^{d_j-1} \cdots X_{i_t}^{d_t}\right) \circ g + \ell(F \circ g) = (\nabla F \cdot \nabla \ell) \circ g + \ell(F \circ g) \quad (\star)$$

By linearity of the gradient, $(\star)$ holds for arbitrary polynomials F. The rest is clear. □

Definition 4.3. If f is a form, the *kth order Jacobian* of f is the ideal generated by all partials of f of order k and is denoted by $J^k(f)$.

Remark 4.4. *The Jacobian of f is $J^1(f)$; geometrically, $V(J^1(f))$ is the singular locus of f. Analogously, $V(J^k(f))$ is the set of singular points with multiplicity at least $k+1$.*

Remark 4.5. *Since we assume f is homogeneous, the Euler identity $\sum x_i \frac{dg}{dx_i} = \deg(g) \cdot g$ applied repeatedly to f and its partials yields the containments $(f) \subset J^1(f) \subset J^2(f) \subset \cdots \subset J^k(f)$. Geometrically, this yields a nested sequence of subvarieties of the hypersurface $V(f)$ ordered according to the severity of the singularities.*

Remark 4.6. *If f is a form of degree d, the degree k component of the apolar algebra $(R_f)_k$, is isomorphic (as a vector space over $\mathbb{K}$) to $J^{d-k}(f)_k$ via apolarity. Hence $Ann_R(f)_k = 0$ if and only if $J^{d-k}(f)$ is the kth power of the maximal ideal.*

According to Remark 4.4, if f is a product of linear forms, then $V(J^k(f))$ is exactly those points which lie at the intersection of at least $k+1$ of the hyperplanes defined by the linear forms whose product is f. Now we arrive at the crucial point: if $f = Q(\mathcal{A})$ for a *generic* arrangement, $V(J^k(f))$ is precisely the union of all codimension $k+1$ intersections of hyperplanes from $\mathcal{A}$. Thus $V(J^k(f))$ is a *star configuration* [7]; a star configuration is by definition the union of all codimension c intersections of a generic arrangement (in [7, Definition 2.1] the property of *meeting properly* is exactly what we mean by a *generic* arrangement). In [7] it is shown that the ideal of codimension c intersections of an arrangement of $|\mathcal{A}|$ hyperplanes is generated by all distinct products of $|\mathcal{A}| - c + 1$ of the linear forms defining $\mathcal{A}$.

Lemma 4.7. *Suppose f decomposes non-trivially as a product $f = gh$; write $I = Ann_R(h)$ and $I' = Ann_R(f) = Ann_R(gh)$. If $D \in I'_k \setminus I_k$, then $g \in J^{k-1}(h) : (D \circ h)$.*

Proof. Repeatedly using the product rule yields that $D \circ gh = g(D \circ h) + T$, where $T \in J^{k-1}(h)$. Since $D \circ gh = 0$, this gives the result. □

Corollary 4.8. *Suppose f is a product of at least $n+2$ distinct linear forms defining a generic arrangement $\mathcal{A}$ in $\mathbb{P}^n$. Factor f as a product $f = gh$ so that $\deg(h) \geq n+1$. Write $I = Ann_R(h)$ and $I' = Ann_R(f) = Ann_R(gh)$. If $I_k = 0$ for any $k \leq n$ then $I'_k = 0$.*

Proof. Suppose to the contrary that $D \in I'_k$ and $D \neq 0$. By Lemma 4.7, $g \in J^{k-1}(h) : (D \circ h)$. Write $h = \ell_1\ell_2 \cdots \ell_t$, where $t \geq n+1$; then $V(J^{k-1}(h))$ is the union of linear spaces which are the intersections of at least k of the hyperplanes $V(\ell_1), \cdots, V(\ell_t)$. This is nonempty since $k \leq n < t$. As $\mathcal{A}$ is a generic arrangement, none of the factors of g vanish along any component of $V(J^{k-1}(h))$; in other words g is not in any prime ideal that comprises the intersection that is the radical of $J^{k-1}(h)$. This means that $g \in J^{k-1}(h) : (D \circ h)$ only if $D \circ h$ is in every minimal prime of $J^{k-1}(h)$. In other words, $D \circ h$ is in the radical of $J^{k-1}(h)$. Let $K = \sqrt{J^{k-1}(h)}$; this is the ideal of the union of linear spaces which are the intersections of k of the hyperplanes $V(\ell_1), \cdots, V(\ell_t)$. As previously noted, this is a star configuration, and by [7, Proposition 2.9], K is generated by all possible products of $t-k+1$ of the linear forms $\ell_1, \ldots, \ell_t$. On the other hand $D \circ h$ has degree $t-k$, so $D \circ h \notin K$. With this contradiction, we must have $I'_k = 0$. □

Remark 4.9. *Consider the A_3 arrangement in $\mathbb{P}^2$, defined by $f = xyz(x-y)(x-z)(y-z)$. Write $f = gh$ with $g = y-z$ and $h = xyz(x-y)(x-z)$. Set $I' = Ann_R(f)$ and $I = Ann_R(h)$. Then $I_2 = 0$ but $I'_2 \neq 0$. Thus the hypothesis that $\mathcal{A}$ is generic in Corollary 4.8 is necessary.*

Now we give the main result of this section – a bound on the initial degree of the apolar ideal of a generic arrangement. For an ideal $I \subset R$ we will denote by $\alpha(I)$ its initial degree, that is, the smallest degree d for which $I_d \neq 0$.

Proposition 4.10. *Suppose $\mathcal{A}$ is a generic arrangement of at least $n+1$ hyperplanes in $\mathbb{P}^n$ and $f = Q(\mathcal{A})$. Then $\alpha(Ann_R(f)) \geq \min\{|\mathcal{A}| - n + 1, n+1\}$.*

Proof. We first prove by induction on $|\mathcal{A}|$ that if $n+1 \leq |\mathcal{A}| \leq 2n$, then $\alpha(\mathrm{Ann}_R(f)) \geq |\mathcal{A}| - n + 1$. If $|\mathcal{A}| = n+1$ then without loss of generality, $f = x_0x_1 \cdots x_n$ and $\mathrm{Ann}_R(f) = (x_0^2, \ldots, x_n^2)$, so $\alpha(\mathrm{Ann}_R(f)) = 2 = |\mathcal{A}| - n + 1$.

Suppose now that $n+1 < |\mathcal{A}| \leq 2n$, and additionally suppose for a contradiction that there is some $D \in \mathrm{Ann}_R(f)_{|\mathcal{A}|-n}$. Since $\mathcal{A}$ is defined by more than n linearly independent linear forms, by Lemma 4.2 there is some $\ell \in \mathcal{A}$ so that $\nabla\ell \cdot \nabla D \neq 0$. Writing $f = g\ell$, with $\deg(g) = n$, and using Lemma 4.2 again, we have

$$0 = D \circ f = (\nabla\ell \cdot \nabla D) \circ g + \ell(D \circ g).$$

Suppose $D \circ g = 0$, so that $(\nabla\ell \cdot \nabla D) \circ g = 0$. Now $\deg(\nabla\ell \cdot \nabla D) = |\mathcal{A}| - n - 1$, and by induction $\alpha(\mathrm{Ann}_R(g)) \geq |\mathcal{A}| - 1 - n + 1 = |\mathcal{A}| - n$. With this contradiction, $D \circ g \neq 0$.

With the above, $\ell(D \circ g)) = -(\nabla\ell \cdot \nabla D) \circ g$, so $\ell(D \circ g) \in J^{|\mathcal{A}|-n-1}(g)$. Write $K = \sqrt{J^{|\mathcal{A}|-n-1}(g)}$, so that K is the ideal defining all possible intersections of $|\mathcal{A}| - n$ hyperplanes of g; by [7], $\alpha(K) = (|\mathcal{A}| - 1) - (|\mathcal{A}| - n) + 1 = n$. Since $\deg(D \circ g) = (|\mathcal{A}| - 1) - (|\mathcal{A}| - n) = n - 1$, $D \circ g \notin K$. Since K is radical, ℓ must be in at least one minimal prime of K. This would imply that $V(\ell)$ passes through a codimension $|\mathcal{A}| - n$ intersection of $\mathcal{A}$. As $|\mathcal{A}| \leq 2n$, K is not the homogeneous maximal ideal, so that this contradicts that $\mathcal{A}$ is a generic arrangement. Hence no such D can exist, and it follows that $\alpha(\mathrm{Ann}_R(f)) \geq |\mathcal{A}| - n + 1$.

If $|\mathcal{A}| \geq 2n$ we prove by induction on $|\mathcal{A}|$ that $\alpha(\mathrm{Ann}_R(f)) \geq n+1$. The base case $|\mathcal{A}| = 2n$ has already been shown. If $|\mathcal{A}| > 2n$ then the result follows from Corollary 4.8. □

Remark 4.11. *We learned in a personal communication from Zach Teitler that he has obtained, with several collaborators, results overlapping with Proposition 4.10. These results have not yet appeared in print.*

Corollary 4.12. *If $\mathcal{A}$ is a generic arrangement of at least $n+2$ hyperplanes in $\mathbb{P}^n$ whose apolar ideal is a complete intersection, then $|\mathcal{A}| \geq n(n+1)$.*

Proof. Put $f = Q(\mathcal{A})$. If $\mathrm{Ann}_R(f)$ is a complete intersection generated in degrees $d_0 \leq \ldots \leq d_n$, then $(d_0-1)+(d_1-1)+\cdots+(d_n-1) = |\mathcal{A}|$, so $d_0+\cdots+d_n = |\mathcal{A}|+n+1$. With this notation, $\alpha(\mathrm{Ann}_R(f)) = d_0$, and this gives $d_0 \leq (|\mathcal{A}|+n+1)/(n+1)$.

It is straightforward to check that if $n+1 < |\mathcal{A}| \leq 2n$ then the lower bound for $\alpha(\mathrm{Ann}_R(f))$ from Proposition 4.10 is strictly larger than $(|\mathcal{A}|+n+1)/(n+1)$, so $\mathrm{Ann}_R(f)$ cannot be a complete intersection.

If $|\mathcal{A}| > 2n$ then we obtain from Proposition 4.10 that $n+1 \leq (|\mathcal{A}|+n+1)/(n+1)$ or equivalently $n(n+1) \leq |\mathcal{A}|$, proving the corollary. □

Corollary 4.13. *The Waring rank of a generic arrangement $\mathcal{A} \subset \mathbb{P}^n$ with at least $n+1$ hyperplanes is at least* $\min\{\binom{|\mathcal{A}|}{n}, \binom{2n}{n}\}$.

Proof. Put $f = Q(\mathcal{A})$. By Proposition 4.10, $\alpha(\mathrm{Ann}_R(f)) \geq \min\{|\mathcal{A}|-n+1, n+1\}$. Suppose $f = \sum_{i=1}^k \ell_i^{|\mathcal{A}|}$, and let $X = \{P_i\}_{i=1}^k$ be the dual points in $\mathbb{P}^n$ found by stripping off the coordinates of the linear forms ℓ_i. By Lemma 2.1, $I_X \subset \mathrm{Ann}_R(f)$. For this to happen, X must impose independent conditions on forms of degree $d = \alpha(\mathrm{Ann}_R(f)) - 1$. In other words, X must consist of at least as many points as the dimension of the vector space S_d, where $S = k[x_0, \ldots, x_n]$. Since $\dim S_d = \binom{n+d}{n}$, this gives the result. □

Remark 4.14. *If $\mathcal{A}$ is a generic arrangement of $k \leq n+1$ hyperplanes in $\mathbb{P}^n$, then up to a change of variables $Q(\mathcal{A}) = x_0 \cdots x_{k-1}$. Put $f = x_0 \cdots x_{k-1}$. It is straightforward that $Ann_R(f)$ is the complete intersection $\langle X_0^2, \ldots, X_{k-1}^2 \rangle$, and it is known that $Q(\mathcal{A})$ has Waring rank 2^{k-1} and explicit Waring decomposition of the form*

$$\sum_{i_1,\ldots,i_{k-1}} \gamma_{i_0,\ldots,i_{k-1}} (x_0 + (-1)^{i_1} x_1 + \cdots + (-1)^{i_{k-1}} x_{k-1})^k,$$

where the sum runs over all possibilities of $i_h \in \{0,1\}$ for $h = 1, \ldots, k-1$ and $\gamma_{i_0,\ldots,i_{k-1}}$ are constants. See [4, Proposition 4.5].

Remark 4.15. *As Corollary 4.13 does not account for the degree of $Q(\mathcal{A})$, we suspect that Corollary 4.13 is not optimal for $|\mathcal{A}| > 2n$. However we will see in Section 5 that, even if $\mathcal{A}$ is generic, $Q(\mathcal{A})$ can be annihilated by many forms of unexpectedly low degree.*

5 SIX LINES IN $\mathbb{P}^2$

In this section we give a computational case study of irreducible multi-arrangements in $\mathbb{P}^2(\mathbb{C})$ with six lines, counting multiplicity. Our motivation for this case study comes from [16, Example III.3.2], where Wakefield observes that the determinant of the catalecticant matrix (defined below) is not enough to show that the apolar algebra of a generic arrangement of six lines in $\mathbb{P}^2(\mathbb{C})$ is not a complete intersection. As a consequence of our case study, we can say with reasonable certainty that there are indeed no generic arrangements of six lines in $\mathbb{P}^2$ whose apolar algebra is a complete intersection. Another motivation for this case study is that, according to Corollary 4.12, a generic line arrangement must have at least six lines in order for its apolar algebra to have the possibility of being a complete intersection.

By Proposition 4.10, a generic arrangement $\mathcal{A}$ of six lines cannot be annihilated by any quadrics. It follows that if the apolar ideal of $\mathcal{A}$ is a complete intersection then it must be generated by a regular sequence of three cubics. In the process of looking for generic arrangements with this property, computations in the computer algebra systems Bertini and Macaulay2 led us to the following (computational) result. We have no theoretical justification for this and have not used software such as alphaCertified for Bertini to give a theoretical guarantee that the computations are correct – hence we will denote it as a Conjecture/Theorem*. The asterix emphasizes that this Conjecture/Theorem* can presumably be turned into a theorem by numerical certification.

Conjecture/Theorem* 5.1. *Suppose that $(\mathcal{A}, \mathbf{m})$ is an irreducible multi-arrangement in $\mathbb{P}^2(\mathbb{C})$ and $|\mathbf{m}| = 6$. Put $f = Q(\mathcal{A}, \mathbf{m})$. Suppose that f satisfies either:*

(1) $\dim Ann_R(f)_3 \geq 3$
(2) *f has cactus rank at most 7*

Then, up to a change in coordinates, f is one of the following six polynomials:

- $f_1 = xyz(x+y+z)(x+\alpha y+\bar{\alpha}z)(x+\bar{\alpha}y+\alpha z)$
- $f_2 = xyz(x+y+z)(x+\eta y+\omega z)(x+\bar{\eta}y+\bar{\omega}z)$
- $f_3 = xyz(x+y+z)(x+\bar{\omega}y+\omega z)(x+\bar{\eta}y+\eta z)$
- $f_4 = xyz(x+y+z)(x+\omega y+\bar{\omega}z)(x+\eta y+\bar{\eta}z)$
- $f_5 = xyz(x+y+z)(x+y)(y+z)$
- $f_6 = x^3yz(x+y+z)$,

where $\alpha = \exp(\frac{2\pi i}{3}), \omega = \exp(\frac{\pi i}{3}), \eta = \frac{1}{\sqrt{3}}\exp(\frac{\pi i}{6})$, and the bar denotes complex conjugation.

In fact, $\dim Ann_R(f_i)_3 = 4$ and the cactus rank of f_i is 6 for $1 \leq i \leq 6$. If instead we require that f has Waring rank six, then f must be one of f_1, f_2, f_3, f_4, or f_5.

Before we discuss the simplifications and further computations leading to this result, we make some remarks about the polynomials listed in Conjecture/Theorem* 5.1.

- The forms f_1, f_2, f_3, and f_4 each define generic arrangements.
- After changing coordinates, f_5 is the defining polynomial of the A_3 braid arrangement.
- The product f_1 is exactly half of the well-known Hessian arrangement (see [12, Example 6.30]).
- For each of $i = 1, \ldots, 6$, $\mathrm{Ann}_R(f_i)$ has four cubics (these are all listed in Table 1). In particular, Proposition 4.10 is tight for these.
- For each of $i = 1, \ldots, 6$, the ideal J_i generated by the elements of degree at most 3 in $\mathrm{Ann}_R(f_i)$ is the ideal of a zero-dimensional scheme of degree six in $\mathbb{P}^2$. Except for $i = 6$, the ideal J_i is the ideal of six reduced points in $\mathbb{P}^2$. These points are listed in Table 2. Via the apolarity lemma (Lemma 2.1), the ideals J_i $(1 \leq i \leq 5)$ yield an explicit Waring decomposition for f_i which is also listed in Table 2. In Table 2, the point p_i is dual to the form ℓ_i.

$\alpha = \exp(\frac{2\pi i}{3}), \omega = \exp(\frac{\pi i}{3}), \eta = \frac{1}{\sqrt{3}}\exp(\frac{\pi i}{6})$

	Annihilating cubics of f_i
f_1	$X^3 - Y^3, X^3 - Z^3, XY^2 + YZ^2 + ZX^2,$ $X^2Y + Y^2Z + Z^2X$
f_2	$X^2Z - XZ^2, 3Y^2Z - 3YZ^2 + Z^3, X^3 - 3X^2Y + 3XY^2, X^2Y - 3XY^2 + 3Y^3 + 2XYZ - XZ^2 - 2YZ^2 + Z^3$
f_3	$-6\eta XY^2 + 6\eta Y^3 + 6\eta XZ^2 + 3\eta YZ^2 - 6\eta Z^3 + X^3 + 2XY^2 - 3Y^3 + 2XYZ - 4XZ^2 - 3YZ^2 + 3Z^3, -3\eta XY^2 + 3\eta Y^3 + X^2Y - Y^3, 3\eta XZ^2 - 3\eta Z^3 + X^2Z - 3XZ^2 + 2Z^3, 3\eta YZ^2 + Y^2Z - 2YZ^2$
f_4	$-6\bar{\eta} XY^2 + 6\bar{\eta} Y^3 + 6\bar{\eta} XZ^2 + 3\bar{\eta} YZ^2 - 6\bar{\eta} Z^3 + X^3 + 2XY^2 - 3Y^3 + 2XYZ - 4XZ^2 - 3YZ^2 + 3Z^3, -3\bar{\eta} XY^2 + 3\bar{\eta} Y^3 + X^2Y - Y^3, 3\bar{\eta} XZ^2 - 3\bar{\eta} Z^3 + X^2Z - 3XZ^2 + 2Z^3, 3\bar{\eta} YZ^2 + Y^2Z - 2YZ^2$
f_5	$X^2 - XY + Y^2 - YZ + Z^2, Y^3 - 2Y^2Z + 2YZ^2, Z^4$ (generators for ideal)
f_6	$Z^3, Y^2Z - YZ^2, Y^3, XY^2 - XYZ + XZ^2 + 2YZ^2$

Table 1: Annihilating cubics of forms f_i in Conjecture/Theorem* 5.1

- It is known that the Waring (and cactus) rank of a form f is at least as large as $\dim(R_f)_k$ for any k; since $\dim(R_{f_i})_k$ is maximal when $k = 3$ and $\dim(R_{f_i})_3 = 6$ for each of $i = 1, \ldots, 6$, the minimum value the Waring (respectively, cactus) rank can be is 6. Thus the Waring rank of $f_1, \ldots, f_5$ is six. For $f_1, \ldots, f_4$, this is the lower bound predicted by Corollary 4.13.

Now we explain the computations that led us to Conjecture/ Theorem* 5.1. We first reduce the number of variables needed.

Lemma 5.2. *If $\mathcal{A}$ is an irreducible arrangement in $\mathbb{P}^2$ then we can change variables so that $f = Q(\mathcal{A})$ has the form $f = xyz(x + y + z)\ell_1\ell_2\cdots\ell_t$, where $\ell_1, \ldots, \ell_t$ are linear forms.*

Proof. If $\mathcal{A}$ is irreducible then $f = Q(\mathcal{A})$ must have three factors which are linearly independent (otherwise $\mathcal{A}$ will decompose as a product of a one or two dimensional arrangement with the 'empty' arrangement). Furthermore f must have at least four factors since otherwise it will decompose as a product of three one-dimensional arrangements.

Changing variables, we may assume that f has the form $f = xyz\ell_0\cdots\ell_t$ ($t \geq 0$). We claim that f has a collection of four factors no three of which are linearly dependent. Suppose for a contradiction that every collection of four factors of f has a subset of three factors which are linearly dependent. Applying this supposition to the collection $\{x, y, z, \ell_i\}$ yields that one of the subsets $\{x, y, \ell_i\}, \{x, z, \ell_i\}$, or $\{y, z, \ell_i\}$ is linearly dependent. Hence ℓ_i must be a linear form in only two variables for $i = 0, \ldots, t$. If each ℓ_i ($i = 0, \ldots, t$) is a function of the *same* two variables, the arrangement clearly decomposes as a product. Hence we may assume without loss that $\ell_1 = x + \alpha y$ and $\ell_2 = x + \beta z$, where $\alpha, \beta \neq 0$. But then $y, z, x + \alpha y, x + \beta z$ forms a collection of four factors of f no three of which are linearly independent, proving the claim.

Since f has a collection of four factors no three of which are linearly independent, we can change variables to make three of these factors x, y, and z. The fourth factor must involve all three variables, hence we can apply scaling in the x, y and z directions to

$\alpha = \exp(\frac{2\pi i}{3}), \beta = 1 + i, \omega = \exp(\frac{\pi i}{3}), \eta = \frac{1}{\sqrt{3}}\exp(\frac{\pi i}{6})$

Form	Dual Points	Waring Decomposition
f_1	$p_1 = [\alpha : 1 : 1]$ $p_2 = [\bar{\alpha} : 1 : 1]$ $p_3 = [1 : \alpha : 1]$ $p_4 = [1 : \bar{\alpha} : 1]$ $p_5 = [1 : 1 : \alpha]$ $p_6 = [1 : 1 : \bar{\alpha}]$	$\frac{2\alpha+1}{270}(-\ell_1^6 + \ell_2^6 - \ell_3^6 + \ell_4^6 - \ell_5^6 + \ell_6^6)$
f_2	$p_1 = [1 : \eta : 1]$ $p_2 = [1 : \bar{\eta} : 1]$ $p_3 = [0 : \eta : 1]$ $p_4 = [0 : \bar{\eta} : 1]$ $p_5 = [1 : \eta : 0]$ $p_6 = [1 : \bar{\eta} : 0]$	$\frac{2\eta-1}{10}(-\ell_1^6 + \ell_2^6 + \ell_3^6 - \ell_4^6 + \ell_5^6 - \ell_6^6)$
f_3	$p_1 = [\omega : 1 : \omega]$ $p_2 = [1 : 1 : \omega]$ $p_3 = [\omega : 1 : 0]$ $p_4 = [1 : 1 : 0]$ $p_5 = [1 : 0 : \omega]$ $p_6 = [1 : 0 : 1]$	$\frac{2\omega-1}{90}(\ell_1^6 - \ell_2^6 - \ell_3^6 + \ell_4^6 + \ell_5^6 - \ell_6^6)$
f_4	$p_1 = [\bar{\omega} : 1 : \bar{\omega}]$ $p_2 = [1 : 1 : \bar{\omega}]$ $p_3 = [\bar{\omega} : 1 : 0]$ $p_4 = [1 : 1 : 0]$ $p_5 = [1 : 0 : \bar{\omega}]$ $p_6 = [1 : 0 : 1]$	$\frac{2\bar{\omega}-1}{90}(\ell_1^6 - \ell_2^6 - \ell_3^6 + \ell_4^6 + \ell_5^6 - \ell_6^6)$
f_5	$p_1 = [\beta : 2 : \bar{\beta}]$ $p_2 = [\bar{\beta} : 2 : \beta]$ $p_3 = [\beta : 2 : \beta]$ $p_4 = [\bar{\beta} : 2 : \bar{\beta}]$ $p_5 = [1 : 0 : i]$ $p_6 = [1 : 0 : \bar{i}]$	$\frac{\ell_1^6 + \ell_2^6 - \ell_3^6 - \ell_4^6 - 8i\ell_5^6 - 8i\ell_6^6}{1920}$

Table 2: Waring decompositions of the forms f_i in Conjecture/Theorem* 5.1. The points p_i give the coefficients of the linear forms ℓ_i.

normalize the coefficients of the fourth factor to one. Thus f can be written in the form $f = xyz(x + y + z)\ell_1 \cdots \ell_t$. □

Corollary 5.3. *If $(\mathcal{A}, \mathbf{m})$ is an irreducible multi- arrangement in $\mathbb{P}^2$ with six lines, then there is a change of variables so that $Q(\mathcal{A}, \mathbf{m}) = xyz(x + y + z)\ell_1\ell_2$, with ℓ_1 and ℓ_2 linear forms.*

Definition 5.4. Let $f \in S$ be a form of degree d and $0 \leq t \leq d$ an integer. The map $\mathrm{Cat}_f(t) : R_t \to S_{d-t}$ defined by $\Phi \to \Phi \circ f$ is the *catelecticant map*. Choosing the usual basis of monomials for R_t and S_{d-t}, we obtain the corresponding *catalecticant matrix*. Abusing notation, we will refer to this matrix also as $\mathrm{Cat}_f(t)$. The rows of $\mathrm{Cat}_f(t)$ correspond to monomials in the basis of S_{d-t}, and the columns of $\mathrm{Cat}_f(t)$ correspond to monomials in the basis of R_t. Suppose X^α is a monomial in R_d and x^β is a monomial in S_{d-t}. The entry of $\mathrm{Cat}_f(t)$ in the row corresponding to X^α and column corresponding to x^β is the coefficient of the monomial x^β in $\frac{\partial f}{\partial x^\alpha}$. It is straightforward to see that $\ker(\mathrm{Cat}_f(t))$ is $\mathrm{Ann}_R(f)_t$.

We return now to the computation at hand. By Corollary 5.3 we make a change of variables so that $f = xyz(x + y + z)\ell_1\ell_2$. Introducing symbolic constants a, b, c, d, e, and f we can write $f = xyz(x + y + z)(ax + by + cz)(dx + ey + fz)$. Now consider the condition in Conjecture/Theorem* 5.1 that $\dim \mathrm{Ann}_R(f)_3 \geq 3$. Here $R = \mathbb{K}[X, Y, Z]$ and $S = \mathbb{K}[x, y, z]$. Using Definition 5.4, we see that $\mathrm{Ann}_R(f)_3 = \ker \mathrm{Cat}_f(3) : R_3 \to S_3$. Evidently $\mathrm{Cat}_f(3)$ is a ten by ten matrix with entries of bi-degree $(1, 1)$ in the variables a, b, c and d, e, f; this matrix is shown in [16, Example III.3.2]. To say $\dim \mathrm{Ann}_R(f)_3 \geq 3$ is equivalent to imposing that $\mathrm{rank}(\mathrm{Cat}_f(3)) \leq 7$. Thus the forms from Conjecture/Theorem* 5.1 can be found as the zero locus of the seven by seven minors of this matrix. As one may imagine, this approach is computationally infeasible.

To impose the rank condition we use an idea from [2] which reduces computation by introducing many auxiliary variables. Explicitly, we introduce a ten by three matrix $\mathcal{Z}$ whose first three rows form a three by three identity matrix and whose remaining entries are filled with new variables:

$$\mathcal{Z} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ A & B & C \\ D & E & F \\ G & H & V \\ J & K & L \\ M & N & O \\ P & Q & R \\ S & T & U \end{bmatrix}.$$

We then impose the condition $\mathrm{Cat}_f(3)\mathcal{Z} = 0$; this guarantees that $\mathrm{Cat}_f(3)$ will have rank at most 7. This yields a system of 30 equations of total degree three in the 27 variables $a, b, c, d, e, f, A, \ldots, V$ (we replace the variable I with V since this is reserved for the imaginary unit in Bertini). We denote this system by $\mathcal{F}$. Since we only look for solutions up to constant multiple in the variable groups a, b, c and d, e, f, we seek solutions in the 25 dimensional space $\mathbb{P}^2 \times \mathbb{P}^2 \times \mathbb{C}^{21}$. In Bertini we can specify this by using the option for homogeneous variable groups. However we still must square the system, which we do by taking 25 random linear combinations of the 30 equations resulting from $\mathrm{Cat}_f(3)B = 0$. This squared system, which we denote by $\mathcal{F}_\square$, consists of 25 equations which are each homogeneous of degree 1 in the variables a, b, c and d, e, f (respectively), and of total degree 1 in the variables $A, \ldots, V$ can now be solved by Bertini. Homogenizing with respect to the variables $A, \ldots, V$, we can regard this as a system of 25 equations of type $(1, 1, 1)$ in $\mathbb{P}^2 \times \mathbb{P}^2 \times \mathbb{P}^{21}$.

The multi-homogeneous Bezout number for a system of 25 equations of type $(1, 1, 1)$ in $\mathbb{P}^2 \times \mathbb{P}^2 \times \mathbb{P}^{21}$ is the multinomial coefficient $\binom{25}{2,2,21} = 75,900$. Thus Bertini starts by tracking $75,900$ paths from a generic multi-homogeneous start system. We ran this computation twice on a local cluster. Using eight nodes with 24 processors on four of the nodes and 20 processors on the other four nodes, this computation took about three hours. Bertini tracked the $75,900$ paths to $29,079$ solutions on the first run and $29,027$ solutions on the second run. Most of the ≈ 29000 approximate solutions of the system $\mathcal{F}_\square$ are not actually solutions of the system $\mathcal{F}$. Selecting those solutions to $\mathcal{F}_\square$ which are approximate solutions to $\mathcal{F}$ within a tolerance of 10^{-6} yields a list of about 1500 approximate solutions to $\mathcal{F}$ (this number might vary depending on the choice of squared system $\mathcal{F}_\square$). This step – selecting a tolerance of 10^{-6} – is the most arbitrary step and the most likely step in which some solutions could have been lost.

We project the list of ≈ 1500 (approximate) solutions of $\mathcal{F}$ to the coordinates corresponding to a, b, c, d, e, f. Now we consider the product $Q(a, b, c, d, e, f) = (ax + by + cz)(dx + ey + fz)$ for each of these; we are only concerned with distinct products $Q(a, b, c, d, e, f)$ up to constant multiple. Taking only those solutions for which $Q(a, b, c, d, e, f)$ is distinct, up to constant multiple, we arrive at a list of 16 or 17 solutions (for the two choices of $\mathcal{F}_\square$ that we made). Each of these was equivalent to one of the polynomials listed in Conjecture/Theorem* 5.1 via a permutation of x, y, z (such a permutation fixes the first four factors) or, in the case of f_6, via the change of variables ϕ defined by $x \to x + y + z, y \to -y, z \to -z$. In fact, it is straightforward to check that the list $f_1, \ldots, f_6$ yields a list of twenty polynomials under permutations of x, y, and z, along with the change of variables ϕ for f_6 (which gives $xyz(x + y + z)^3$). In both of the computations that we ran, we ended up with a subset of these 20 polynomials; thus allowing us to simplify the final list to the six polynomials in Conjecture/Theorem* 5.1.

The scripts in Macaulay2, Bertini, and Sage which we used to find the forms in Conjecture/Theorem* 5.1 and verify their properties may be found under the Research tab at https://midipasq.github.io/. Python scripts for post-processing the data from Bertini are also available upon request.

6 CONCLUSIONS AND FURTHER QUESTIONS

There are two main results of this paper. The first is a bound on the initial degree of the apolar ideal of a generic arrangement, attained using defining equations of star configurations from [7]. From this we obtained a necessary condition on the size of a generic arrangement with a complete intersection apolar algebra, as well as a lower bound on the Waring rank of a generic arrangement. A subsequent question raised by Wakefield [16] remains wide open – is the apolar algebra of a generic arrangement ever a complete intersection? To this we add two additional questions concerning

the optimality of Proposition 4.10 and Corollary 4.13. First, are there arbitrarily large generic arrangements in $\mathbb{P}^n$ whose apolar ideals have initial degree $n+1$? Second, are there arbitrarily large generic arrangements in $\mathbb{P}^n$ whose Waring rank is $\binom{2n}{n}$?

The second main result of this paper is the use of apolar algebras and numerical algebraic geometry to determine the irreducible multi-arrangements with six lines in $\mathbb{P}^2$ with minimal Waring rank. We determined that, up to a change of coordinates, there are six irreducible multi-arrangements that have cactus rank equal to six, five of which also have Waring rank equal to six. These results are summarized in Conjecture/Theorem* 5.1. The * indicates that this is a "numerically established theorem" and thus falls short of being a rigorously proved theorem. While one can check that each of these forms has the claimed Waring decomposition, one can't be certain that there do not exist further examples without further work. Thus, an obvious extension of this paper, that needs to be carried out, would be to either provide an alternate approach to establish that these are the only such forms that have this property or else utilize software such as alphaCertified for Bertini to give a theoretical guarantee that the computations are correct. At present, the way we have chosen to make the computations is too expensive to carry out using alphaCertified for Bertini on the system that we used.

The general problem of determining the degree d irreducible multi-arrangements in $\mathbb{P}^n$ that have minimal Waring rank (and minimal cactus rank) is currently out of reach but we leave it as a suggestion for a further path of research. It is worth noting that each of the extremal examples we found has interesting combinatorial properties. In particular, after a change of coordinates, one is the defining ideal of the A_3 braid arrangement. Another is half of the Hessian arrangement. We do not know if there is a deeper connection to reflection arrangements in the examples we have found, but perhaps there is a clue in the structure of these examples that can help one search for higher degree extremal examples. Another promising avenue is to look for extremal behavior among subarrangements of reflection arrangements or the simplicial line arrangements catalogued by Grunbaum [9]; such arrangements have recently led to interesting examples for the *containment problem* between regular and symbolic powers [13]. For now, we leave this as an open problem for the interested reader.

ACKNOWLEDGMENTS

We thank Tanner Strunk for running our Bertini script on a local CSU computer cluster. We would also like to thank Max Wakefield for pointing out to us that f_1 in Table 1 is half of the Hessian arrangement. We thank Zach Teitler for comments on the first draft, and for informing us of work he has done on this problem. The third author was partially supported by NSF 1712788 and NSF 1830676.

REFERENCES

[1] Takuro Abe, Hiroaki Terao, and Masahiko Yoshinaga. 2009. Totally free arrangements of hyperplanes. *Proc. Amer. Math. Soc.* 137, 4 (2009), 1405–1410. https://doi.org/10.1090/S0002-9939-08-09755-4

[2] Daniel J Bates, Jonathan D Hauenstein, Chris Peterson, and Andrew J Sommese. 2009. Numerical decomposition of the rank-deficiency set of a matrix of multivariate polynomials. In *Approximate Commutative Algebra*. Springer, 55–77.

[3] Daniel J. Bates, Jonathan D. Hauenstein, Andrew J. Sommese, and Charles W. Wampler. [n. d.]. Bertini: Software for Numerical Algebraic Geometry. Available at bertini.nd.edu with permanent doi: dx.doi.org/10.7274/R0H41PB5. ([n. d.]).

[4] Enrico Carlini, Maria Virginia Catalisano, and Anthony V Geramita. 2012. The solution to the Waring problem for monomials and the sum of coprime monomials. *Journal of algebra* 370 (2012), 5–14.

[5] Alessandro De Paris. 2018. Seeking for the Maximum Symmetric Rank. *Mathematics* 6, 11 (2018), 247.

[6] Anthony V. Geramita. 1996. Inverse systems of fat points: Waring's problem, secant varieties of Veronese varieties and parameter spaces for Gorenstein ideals. In *The Curves Seminar at Queen's, Vol. X (Kingston, ON, 1995)*. Queen's Papers in Pure and Appl. Math., Vol. 102. Queen's Univ., Kingston, ON, 2–114.

[7] A. V. Geramita, B. Harbourne, and J. Migliore. 2013. Star configurations in $\mathbb{P}^n$. *J. Algebra* 376 (2013), 279–299. https://doi.org/10.1016/j.jalgebra.2012.11.034

[8] Daniel R. Grayson and Michael E. Stillman. [n. d.]. Macaulay2, a software system for research in algebraic geometry. Available at http://www.math.uiuc.edu/Macaulay2/. ([n. d.]).

[9] Branko Grünbaum. 2009. A catalogue of simplicial arrangements in the real projective plane. *Ars Math. Contemp.* 2, 1 (2009), 1–25. https://doi.org/10.26493/1855-3974.88.e12

[10] Anthony Iarrobino and Vassil Kanev. 1999. *Power sums, Gorenstein algebras, and determinantal loci*. Lecture Notes in Mathematics, Vol. 1721. Springer-Verlag, Berlin. xxxii+345 pages. https://doi.org/10.1007/BFb0093426 Appendix C by Iarrobino and Steven L. Kleiman.

[11] Richard Kane. 2001. *Reflection groups and invariant theory*. CMS Books in Mathematics/Ouvrages de Mathématiques de la SMC, Vol. 5. Springer-Verlag, New York. x+379 pages. https://doi.org/10.1007/978-1-4757-3542-0

[12] Peter Orlik and Hiroaki Terao. 1992. *Arrangements of hyperplanes*. Grundlehren der Mathematischen Wissenschaften [Fundamental Principles of Mathematical Sciences], Vol. 300. Springer-Verlag, Berlin. xviii+325 pages. https://doi.org/10.1007/978-3-662-02772-1

[13] Justyna Szpond and Grzegorz Malara. 2017. The containment problem and a rational simplicial arrangement. *Electron. Res. Announc. Math. Sci.* 24 (2017), 123–128.

[14] Zach Teitler and Alexander Woo. 2015. Power sum decompositions of defining equations of reflection arrangements. *J. Algebraic Combin.* 41, 2 (2015), 365–383. https://doi.org/10.1007/s10801-014-0539-0

[15] The Sage Developers. 2017. *SageMath, the Sage Mathematics Software System (Version 8.1)*. `https://www.sagemath.org`.

[16] Max Wakefield. 2006. *On the derivation module and apolar algebra of an arrangement of hyperplanes*. ProQuest LLC, Ann Arbor, MI. 85 pages. http://gateway.proquest.com/openurl?url_ver=Z39.88-2004&rft_val_fmt=info:ofi/fmt:kev:mtx:dissertation&res_dat=xri:pqdiss&rft_dat=xri:pqdiss:3224129 Thesis (Ph.D.)–University of Oregon.

Global Optimization via the Dual SONC Cone and Linear Programming

Mareike Dressler
University of California, San Diego, Department of Mathematics
La Jolla, CA 92093-0112, USA
mdressler@ucsd.edu

Janin Heuer
Technische Universität Braunschweig, Institut für Analysis und Algebra, AG Algebra
38106 Braunschweig, Germany
janin.heuer@tu-braunschweig.de

Helen Naumann
Goethe Universität, FB 12- Institut für Mathematik
D-60054, Frankfurt a.M., Germany
naumann@math.uni-frankfurt.de

Timo de Wolff
Technische Universität Braunschweig, Institut für Analysis und Algebra, AG Algebra
38106 Braunschweig, Germany
t.de-wolff@tu-braunschweig.de

ABSTRACT

Using the dual cone of sums of nonnegative circuits (SONC), we provide a relaxation of the global optimization problem to minimize an exponential sum and, as a special case, a multivariate real polynomial. Our approach builds on two key observations. First, that the dual SONC cone is contained in the primal one. Hence, containment in this cone is a certificate of nonnegativity. Second, we show that membership in the dual cone can be verified by a linear program. We implement the algorithm and present initial experimental results comparing our method to existing approaches.

CCS CONCEPTS

• **Mathematics of computing → Nonconvex optimization**; *Semidefinite programming*; *Mathematical software performance*; *Linear programming*.

KEYWORDS

circuit polynomial, dual cone, linear programming, nonconvex global optimization, SONC

ACM Reference Format:
Mareike Dressler, Janin Heuer, Helen Naumann, and Timo de Wolff. 2020. Global Optimization via the Dual SONC Cone and Linear Programming. In *International Symposium on Symbolic and Algebraic Computation (ISSAC '20), July 20–23, 2020, Kalamata, Greece.* ACM, New York, NY, USA, 8 pages. https://doi.org/10.1145/3373207.3404043

ISSAC '20, July 20–23, 2020, Kalamata, Greece

ACM ISBN 978-1-4503-7100-1/20/07…$15.00
https://doi.org/10.1145/3373207.3404043

1 INTRODUCTION

Let $A \subseteq \mathbb{R}^n$ be a finite set and let $\mathbb{R}^A$ denote the space of all *(sparse) exponential sums* supported on A. These are of the form

$$f = \sum_{\boldsymbol{\alpha} \in A} c_{\boldsymbol{\alpha}} e^{\langle \mathbf{x}, \boldsymbol{\alpha} \rangle} \in \mathbb{R}^A, \quad c_{\boldsymbol{\alpha}} \in \mathbb{R} \text{ for all } \boldsymbol{\alpha} \in A. \tag{1}$$

We consider the following global optimization problem

$$\inf_{\mathbf{x} \in \mathbb{R}^n} f(\mathbf{x}), \tag{2}$$

which is the unconstrained version of a *signomial optimization problem*. Signomial programs are a rich class of nonconvex optimization problems with a broad range of applications; see e.g., [5, 11] for an overview.

If $A \subseteq \mathbb{N}^n$, then $\mathbb{R}^A$ coincides with the space of real polynomials on the positive orthant supported on A. Thus, (2) also represents all unconstrained *polynomial optimization problems* on $\mathbb{R}^n_{>0}$; see e.g. [4, 20, 21] for an overview about polynomial optimization problems and their applications.

Under the assumption that (2) has a finite solution, minimizing $f \in \mathbb{R}^A$ is equivalent to adding a minimal constant γ such that $f + \gamma \geq 0$. Hence, we consider the (convex, closed) *sparse nonnegativity cone* in $\mathbb{R}^A$, which is defined as

$$\mathcal{P}^+_A = \{f \in \mathbb{R}^A : f(\boldsymbol{x}) \geq 0 \text{ for all } \mathbf{x} \in \mathbb{R}^n\}. \tag{3}$$

It is well-known that deciding nonnegativity is NP-hard even in the polynomial case; see e.g., [22]. Thus, a common way to attack (2), is to search for *certificates of nonnegativity*. These conditions, which imply nonnegativity, are easier to test than nonnegativity itself, and are satisfied for a vast subset of $\mathcal{P}^+_A$. In the polynomial case, a well-known example of a certificate of nonnegativity are *sums of squares (SOS)*, which can be tested via *semidefinite programming* [19, 26]. Unfortunately, SOS decompositions do not preserve the sparsity of A.

Another certificate of nonnegativity is a decomposition of f into *sums of nonnegative circuit functions (SONC)*, which were introduced by Iliman and the last author for polynomials [16] generalizing work by Reznick [27]. Recently, the SONC approach was generalized and reinterpreted by Forsgård and the last author [14]. A *circuit function* is a function, which is supported on a minimally affine dependent set; see Definition 2.1. For these kind of functions nonnegativity

can effectively be decided by solving a system of linear equations; see Theorem 2.2.

SONCs form a closed convex cone $\mathcal{S}_A^+ \subseteq \mathcal{P}_A^+$. This cone and the functions therein respectively were investigated independently by other authors using a separate terminology. The perspective of considering $\mathcal{S}_A^+$ as a subclass of nonnegative signomials was originally introduced by Chandrasekaran and Shah [6] under the name *SAGE*, which was later generalized by Chandrasekaran, Murray, and Wierman [24, 25]. Furthermore, the notion of SONC was re-interpreted by Katthän, Theobald and the third author [18] under the name *$\mathcal{S}$-cone*. We discuss the relation of these different approaches to each other in Section 2.

The key idea of this article is to relax the problem (2) via optimizing over the *dual SONC cone* $\check{\mathcal{S}}^+_{A^+,A^-}$; see Definition 3.1 for a rigorous definition. Our approach is motivated by the recent works [10], [24], and [18], and builds on two key observations, which are the main theoretical contributions:

(1) The dual SONC cone is contained in the primal one; see Proposition 3.6.
(2) Optimizing over the dual cone can be carried out by linear programming; see Proposition 4.1.

We emphasize that neither the primal nor the dual SONC cone is polyhedral; see in this context also the results in [14]. The approach works as follows: First, we investigate a lifted version of the dual cone involving additional linear auxiliary variables (Theorem 3.2 (3)). Second, we show that the coefficients of a given exponential sum can be interpreted as variables of the dual cone; see (8). Third, we observe that fixing these coefficient variables yields an optimization problem only involving the linear auxiliary variables; see (4.1)

Based on our two key observations stated above, we present in Section 4 two linear programs (LP_{A^+}) and (LP_{A^-}) solving a relaxation of (2). We implemented the proposed algorithm and provide a collection of examples showing that (LP_{A^+}) and (LP_{A^-}) work in practice. Using the software POEM [30], we compare our approach exemplarily to existing algorithms for finding SONC and SAGE decompositions via the primal cone $\mathcal{S}_A^+$, as well as to SOS bounds.

ACKNOWLEDGMENTS

We thank Thorsten Theobald for his help and input during the development of this article. We thank the anonymous referees for their helpful comments.

TdW is supported by the DFG grant WO 2206/1-1.

2 PRELIMINARIES

We display vectors in bold notation, e.g., $\boldsymbol{x}$ for $(x_1, \ldots, x_n)$. Throughout the article we write $\mathbb{R}_{>0} = \{x \in \mathbb{R} : x > 0\}$. Given a set $A \subseteq \mathbb{R}^n$ we denote by $\mathrm{conv}(A)$ its *convex hull*. We refer to the vertices of $\mathrm{conv}(A)$ as $\mathrm{Vert}(\mathrm{conv}(A))$. For a given linear space L we denote by $\check{L}$ its *dual space*, and, similarly, for a given cone $C \subseteq \mathbb{R}^n$, we denote by $\check{C}$ its *dual cone*. For the logarithmic function, we use the conventions $0 \ln(\frac{0}{y}) = 0$, $\ln(\frac{y}{0}) = \infty$ if $y > 0$ and $\ln(\frac{0}{0}) = 0$ and in addition $\ln(0) = -\infty$.

2.1 Nonnegativity and the SONC Cone

Let $A \subseteq \mathbb{R}^n$ be a finite set referred to as the *support set*; in what follows we set $d = \#A$. Recall that we consider exponential sums of the form (1). For such an f, we set $\mathrm{supp}(f) = A$ and denote the vector of coefficients as $\boldsymbol{c}$. If f is comprised by a single term, then we call it an *exponential monomial*.

Following the approach of *fewnomial theory* (also referred to as "*A-philosophy*" by Gelfand, Kapranov and Zelevinsky; see e.g., [15]), we fix $A \subseteq \mathbb{R}^n$ and consider the space $\mathbb{R}^A$ of all functions with support set A, i.e.,

$$\mathbb{R}^A = \mathrm{span}_{\mathbb{R}} \left(\left\{ e^{\langle \mathbf{x}, \boldsymbol{\alpha} \rangle} \ : \ \boldsymbol{\alpha} \in A \right\} \right).$$

Since A is fixed, every $f \in \mathbb{R}^A$ can be identified with its coefficient vector and hence there exists a canonical isomorphism $\mathbb{R}^A \simeq \mathbb{R}^d$, i.e., we denote both, vectors and functions, as elements in $\mathbb{R}^A$. If $A \subseteq \mathbb{N}^n$, $\mathbb{R}^A$ coincides with the space of real polynomials on the positive orthant supported on A.

Recall that the sparse nonnegativity cone $\mathcal{P}_A^+$ defined in (3) is a full-dimensional convex closed cone in $\mathbb{R}^A$. It is a well-known fact that $f \in \mathcal{P}_A^+$ only if all coefficients associated to vertices of $\mathrm{conv}(\mathrm{supp}(f))$ are positive; see e.g., [12] for a detailed proof. Thus, we make the assumption

$$\boldsymbol{\alpha} \in \mathrm{Vert}\,(\mathrm{conv}(\mathrm{supp}(f))) \implies c_{\boldsymbol{\alpha}} > 0. \tag{4}$$

Since deciding membership in $\mathcal{P}_A^+$ is NP-hard, we intend to *certify* membership in $\mathcal{P}_A^+$ via considering a subcone. For us, the main ingredient is an object called a *circuit function*. Recall that a subset A' of A is called a *circuit* if A is minimally affine dependent (i.e., all real subsets of A' are affinely independent); see e.g., [23]. A special version of circuit functions was first introduced under the name *simplicial AGI-form* by Reznick in [27], the general definition was given by Iliman and the last author in [16] focusing on polynomials. Here, we build on a recent, generalized notion by Forsgård and the last author [14].

Definition 2.1 (circuit function). A function $f \in \mathbb{R}^A$ is called a *circuit function* if $\mathrm{supp}(f)$ is a circuit, $\mathrm{conv}(\mathrm{supp}(f))$ is a simplex, and it satisfies (4).

In the special case $A \subseteq \mathbb{N}^n$, circuit functions are precisely *circuit polynomials* on $\mathbb{R}^n_{>0}$ as introduced in [16].

A crucial fact about a circuit function f is that its nonnegativity can be decided by an invariant Θ_f called the *circuit number* alone. Specifically, Iliman and the last author showed for the case of polynomials, which immediately generalizes to the case of circuit functions:

Theorem 2.2 ([16], Theorem 1.1). *Let $f = \sum_{j=0}^r c_{\boldsymbol{\alpha}(j)} \mathbf{x}^{\boldsymbol{\alpha}(j)} + c_{\boldsymbol{\beta}} \mathbf{x}^{\boldsymbol{\beta}}$ with $0 \le r \le n$ be a circuit polynomial with $\boldsymbol{\alpha}(\mathbf{0}), \ldots, \boldsymbol{\alpha}(\boldsymbol{r}) \in (2\mathbb{N})^n$, and let $\boldsymbol{\lambda} \in \mathbb{R}^r_{>0}$ denote the vector of barycentric coordinates of $\boldsymbol{\beta}$ in terms of the $\boldsymbol{\alpha}(\mathbf{0}), \ldots, \boldsymbol{\alpha}(\boldsymbol{r})$. Then f is nonnegative if and only if*

$$|c_{\boldsymbol{\beta}}| \ \le \ \Theta_f \ = \ \prod_{j=0}^{r} \left(\frac{c_{\boldsymbol{\alpha}(j)}}{\lambda_j} \right)^{\lambda_j}$$

or if f is a sum of monomial squares.

Note furthermore that a circuit polynomial is nonnegative on $\mathbb{R}^n$ if and only if it is nonnegative on $\mathbb{R}^n_{>0}$ (this is, of course, not the case for general polynomials). Thus, if one is specifically interested in certifying nonnegativity of polynomials on the entire $\mathbb{R}^n$ using circuit polynomials, then one needs to relax the problem first such that the minimum is attained on $\mathbb{R}^n_{>0}$. We refer readers who are interested in further details to the discussion in [16, Section 3.1].

We consider now the cone of all sums of nonnegative circuits.

Definition 2.3. We define the *SONC cone* $\mathcal{S}^+_A$ as the subset of all $f \in \mathcal{P}^+_A$, which can be written as a sum of nonnegative circuit functions or nonnegative exponential monomials.

It is easy to see that $\mathcal{S}^+_A$ indeed is a convex cone (compare e.g., [14, 16]), and it can be shown that $\dim(\mathcal{P}^+_A) = \dim(\mathcal{S}^+_A)$; see [8, Theorem 4.1] for the non-sparse, polynomial case, which generalizes verbatim to the sparse case considered here.

The SONC cone was studied over the past years by other authors using different approaches and terminology. We especially emphasize two of them:

(1) Katthän, Theobald, and the third author studied the *$\mathcal{S}$-cone* in [18]. This cone contains sums of nonnegative functions $f : \mathbb{R}^n \to \mathbb{R} \cup \{\infty\}$ of the form

$$f(\mathbf{x}) = \sum_{\boldsymbol{\alpha} \in \mathcal{A}} c_{\boldsymbol{\alpha}} |\mathbf{x}|^{\boldsymbol{\alpha}} + \sum_{\boldsymbol{\beta} \in \mathcal{B}} d_{\boldsymbol{\beta}} \mathbf{x}^{\boldsymbol{\beta}},$$

where $\mathcal{A} \subseteq \mathbb{R}^n$ and $\mathcal{B} \subseteq \mathbb{N}^n \setminus (2\mathbb{N})^n$ are finite sets of exponents, $\{c_{\boldsymbol{\alpha}} : \boldsymbol{\alpha} \in \mathcal{A}\} \subseteq \mathbb{R}$, $\{d_{\boldsymbol{\beta}} : \boldsymbol{\beta} \in \mathcal{B}\} \subseteq \mathbb{R}$ with either at most one $\boldsymbol{\beta} \in \mathcal{B}$ such that $d_{\boldsymbol{\beta}} \neq 0$ and $c_{\boldsymbol{\alpha}} \geq 0$ for every $\boldsymbol{\alpha} \in \mathcal{A}$, or $d_{\boldsymbol{\beta}} = 0$ for all $\boldsymbol{\beta} \in \mathcal{B}$ and there exists at most one $\boldsymbol{\alpha} \in \mathcal{A}$ such that $c_{\boldsymbol{\alpha}} < 0$. Since each term with exponent in $\mathcal{A}$ is isomorphic to an exponential monomial, and it is sufficient to test nonnegativity of these functions on $\mathbb{R}^n_{>0}$, the functions in the $\mathcal{S}$-cone can be regarded as an exponential sum of the form (1). Furthermore, one can show that for $\mathcal{B} = \emptyset$ the $\mathcal{S}$-cone coincides with the SONC cone as given in Definition 2.3.

(2) Chandrasekaran and Shah introduced an object called *SAGE cone* in [6], which was then studied further in follow-up articles by Chandrasekaran, Murray, and Wiermann [24, 25]. This cone contains sums of nonnegative *AGE functions*, where an AGE function is of the form

$$f(\mathbf{x}) = \sum_{\boldsymbol{\alpha} \in A'} c_{\boldsymbol{\alpha}} e^{\langle \mathbf{x}, \boldsymbol{\alpha} \rangle} + c_{\boldsymbol{\beta}} e^{\langle \mathbf{x}, \boldsymbol{\beta} \rangle} \in \mathbb{R}^A,$$

such that $A' \subseteq A \subseteq \mathbb{R}^n$, $\boldsymbol{\beta} \in A \setminus A'$, and $c_{\boldsymbol{\alpha}} > 0, c_{\boldsymbol{\beta}} \in \mathbb{R}$. Note that for an AGE-function to be nonnegative, it needs to hold that $\boldsymbol{\beta} \in \operatorname{conv}(A)$.
The SAGE cone coincides with the SONC cone $\mathcal{S}^+_A$. This was shown by Reznick in the case of AGI-forms already 1989 in [27]. AGI-forms are a special case of circuit polynomials when choosing $c_{\boldsymbol{\alpha}} = \lambda_j$ and $c_{\boldsymbol{\beta}} = -1$. For the general case it was first shown (but not explicitly stated) by Wang [31]. Briefly afterwards, Chandrasekaran, Murray, and Wiermann [24] finally were the first to explicitly state this fact, which was then observed again in the language of the $\mathcal{S}$-cone by Katthän, Theobald, and the third author in [18].

2.2 The Signed SONC Cone

As a next step, motivated by our approach from optimization, we make a restriction when investigating the SONC cone. For a fixed exponential sum f, which we intend to minimize, we have additional information on the signs of the coefficients of f. Since every coefficient corresponds to an element in A due to the isomorphism $\mathbb{R}^d \simeq \mathbb{R}^A$ described above, we obtain a decomposition

$$A = A^+ \cup A^- \tag{5}$$

with disjoint sets $\emptyset \neq A^+ \subseteq \mathbb{R}^n$, corresponding to positive coefficients $c_{\boldsymbol{\alpha}}$, and $A^- \subseteq \mathbb{R}^n$ corresponding to the remaining nonpositive coefficients $c_{\boldsymbol{\beta}}$ in the exponential sum that we consider. Thus, we represent exponential sums in this case as

$$f = \sum_{\boldsymbol{\alpha} \in A^+} c_{\boldsymbol{\alpha}} e^{\langle \mathbf{x}, \boldsymbol{\alpha} \rangle} + \sum_{\boldsymbol{\beta} \in A^-} c_{\boldsymbol{\beta}} e^{\langle \mathbf{x}, \boldsymbol{\beta} \rangle} \in \mathbb{R}^A. \tag{6}$$

If we minimize a given function f using the SONC approach, then we restrict to circuits respecting the sign-pattern indicated by f. This is the common, tractable approach used by various authors in previous works, e.g., [9, 17, 24, 25]; it motivates the following definition.

Definition 2.4 (Signed SONC cone). Let $A \subseteq \mathbb{R}^n$ be a finite set joint with a decomposition $A = A^+ \cup A^-$ in the sense of (5). Then the *signed SONC cone* $\mathcal{S}^+_{A^+,A^-}$ is the cone of all functions that can be written as a sum of nonnegative circuit functions of the form (6) or as nonnegative exponential monomials with support in A^+. In other words, $\mathcal{S}^+_{A^+,A^-}$ is the intersection of $\mathcal{S}^+_A$ with a particular orthant indicated by the pair (A^+, A^-). We denote the special case $A^- = \{\boldsymbol{\beta}\}$ as $\mathcal{S}^+_{A^+,\boldsymbol{\beta}}$.

In fact, by using a generalization of the circuit number and the subsequent notation, we can refine the representation of $\mathcal{S}^+_{A^+,\boldsymbol{\beta}}$.

Definition 2.5. For a non-empty finite set $A^+ \subseteq \mathbb{R}^n$ and $\boldsymbol{\beta} \in \mathbb{R}^n$ let $\Lambda(A^+, \boldsymbol{\beta})$ be the polytope

$$\Lambda(A^+, \boldsymbol{\beta}) = \left\{ \boldsymbol{\lambda} \in \mathbb{R}^{A^+}_{\geq 0} : \sum_{\boldsymbol{\alpha} \in A^+} \lambda_{\boldsymbol{\alpha}} \boldsymbol{\alpha} = \boldsymbol{\beta}, \sum_{\boldsymbol{\alpha} \in A^+} \lambda_{\boldsymbol{\alpha}} = 1 \right\}. \tag{7}$$

The polytope $\Lambda(A^+, \boldsymbol{\beta})$ is nonempty if and only if $\boldsymbol{\beta}$ is contained in the convex hull of A^+ and $\Lambda(A^+, \boldsymbol{\beta})$ consists of a single element whenever the elements in A^+ are affinely independent. Particularly, $\boldsymbol{\lambda} \in \Lambda(A^+, \boldsymbol{\beta})$ is, in general, not unique for functions in $\mathcal{S}^+_{A^+,\boldsymbol{\beta}}$.

Using (7), we may express $\mathcal{S}^+_{A^+,\boldsymbol{\beta}}$ as follows:

Theorem 2.6 ([18], Theorem 2.7). *Let $A = A^+ \cup \{\boldsymbol{\beta}\}$ be defined as in* (5). *The signed SONC cone is the set*

$$\mathcal{S}^+_{A^+,\boldsymbol{\beta}} = \left\{ \sum_{\boldsymbol{\alpha} \in A^+} c_{\boldsymbol{\alpha}} e^{\langle \mathbf{x}, \boldsymbol{\alpha} \rangle} + c_{\boldsymbol{\beta}} e^{\langle \mathbf{x}, \boldsymbol{\beta} \rangle} : \begin{array}{c} \exists \boldsymbol{\lambda} \in \Lambda(A^+, \boldsymbol{\beta}) \text{ such that} \\ \prod_{\boldsymbol{\alpha} \in A^+ : \lambda_{\boldsymbol{\alpha}} > 0} \left(\frac{c_{\boldsymbol{\alpha}}}{\lambda_{\boldsymbol{\alpha}}} \right)^{\lambda_{\boldsymbol{\alpha}}} \geq -c_{\boldsymbol{\beta}} \end{array} \right\}.$$

Note that nonnegativity of an AGE function f can be certified by using the analog of the circuit number $\Theta_f = \prod_{\boldsymbol{\alpha} \in A^+ : \lambda_{\boldsymbol{\alpha}} > 0} \left(\frac{c_{\boldsymbol{\alpha}}}{\lambda_{\boldsymbol{\alpha}}} \right)^{\lambda_{\boldsymbol{\alpha}}}$ given in the theorem. There is no need to decompose f into a sum of nonnegative circuit functions.

3 THE DUAL SONC CONE

In what follows, we study the dual SONC cone to show containment of the dual in the primal SONC cone (Section 3.2) and to obtain a fast linear approximation for global optimization (Section 4).

Due to our goals in this article, we discuss here duality with respect to the signed SONC cone. However, everything generalizes to the full SONC cone immediately.

3.1 Representations of the Dual SONC Cone

Definition 3.1 (The dual signed SONC cone). For an exponential sum $f \in \mathbb{R}^A$ with coefficient vector $\boldsymbol{c} \in \mathbb{R}^A$ we consider the *natural duality pairing*

$$\boldsymbol{v}(f) = \sum_{\boldsymbol{\alpha}\in A^+} v_{\boldsymbol{\alpha}} c_{\boldsymbol{\alpha}} + \sum_{\boldsymbol{\beta}\in A^-} v_{\boldsymbol{\beta}} c_{\boldsymbol{\beta}} \in \mathbb{R},$$

where, as in the primal case, $\boldsymbol{v}(\cdot) \in \check{\mathbb{R}}^A$ is canonically identified with its (dual) coefficient vector $\boldsymbol{v}$, and hence $\check{\mathbb{R}}^A \simeq \check{\mathbb{R}}^d$. Using this definition, the *dual signed SONC cone* is defined as the set

$$\check{\mathcal{S}}^+_{A^+,A^-} = \left\{\boldsymbol{v} \in \check{\mathbb{R}}^A : \boldsymbol{v}(f) \geq 0 \text{ for all } f \in \mathcal{S}^+_{A^+,A^-}\right\}.$$

For brevity, we refer to this cone simply as the *dual SONC cone*.

The following theorem provides two representations of this cone. We need the first one to show containment of the dual SONC cone in the primal one, and the second representation to obtain the linear program approximating the solution of our global optimization problem (2).

Theorem 3.2 (The dual SONC cone). *Let $A = A^+ \cup A^-$ be as in (5). The following sets are equal.*

(1) $\check{\mathcal{S}}^+_{A^+,A^-}$,

(2)
$$\left\{\boldsymbol{v} \in \check{\mathbb{R}}^A : \begin{array}{c} \forall \boldsymbol{\alpha} \in A^+, v_{\boldsymbol{\alpha}} \geq 0, \text{ and } \forall \boldsymbol{\beta} \in A^-, \forall \boldsymbol{\lambda} \in \Lambda(A^+, \boldsymbol{\beta}), \\ \ln(|v_{\boldsymbol{\beta}}|) \leq \sum_{\boldsymbol{\alpha}\in A^+} \lambda_{\boldsymbol{\alpha}} \ln(v_{\boldsymbol{\alpha}}) \end{array}\right\},$$

(3)
$$\left\{\boldsymbol{v} \in \check{\mathbb{R}}^A : \begin{array}{c} \forall \boldsymbol{\alpha} \in A^+, v_{\boldsymbol{\alpha}} \geq 0, \text{ and } \forall \boldsymbol{\beta} \in A^- \; \exists \boldsymbol{\tau} \in \mathbb{R}^n, \\ \forall\, \boldsymbol{\alpha} \in A^+, \ln\left(\frac{|v_{\boldsymbol{\beta}}|}{v_{\boldsymbol{\alpha}}}\right) \leq (\boldsymbol{\alpha}-\boldsymbol{\beta})^T \boldsymbol{\tau} \end{array}\right\}.$$

To prove these representations, we adapt the subsequent theorem from [24] to our setting, which basically states that a function in the SONC cone supported on $A = A^+ \cup A^-$ can be decomposed into a sum of nonnegative AGE functions supported on $A^+ \cup \{\boldsymbol{\beta}\}$, $\boldsymbol{\beta} \in A^-$, i.e., the decomposition only uses the support A and there is only one summand per element in A^-.

Theorem 3.3 ([24], Theorem 2). *Let $f \in \mathcal{S}^+_{A^+,A^-}$ with a vector of coefficients $\boldsymbol{c}$. Let $A^- \neq \emptyset$. Then there exist $\{f^{(\boldsymbol{\beta})} : \boldsymbol{\beta} \in A^-\} \subseteq \mathbb{R}^A$ with coefficient vectors $\{\boldsymbol{c}^{(\boldsymbol{\beta})} : \boldsymbol{\beta} \in A^-\}$ satisfying*

(1) $\boldsymbol{c} = \sum_{\boldsymbol{\beta}\in A^-} \boldsymbol{c}^{(\boldsymbol{\beta})}$,

(2) $f^{(\boldsymbol{\beta})} \in \mathcal{S}^+_{A^+,\boldsymbol{\beta}}$, *and*

(3) $c^{(\boldsymbol{\beta})}_{\boldsymbol{\alpha}} = 0$ *for all* $\boldsymbol{\alpha} \neq \boldsymbol{\beta}$ *in* A^-.

We obtain the following representation of the SONC cone and its dual.

Corollary 3.4.

(1) *The SONC cone is the Minkowski sum*
$$\mathcal{S}^+_{A^+,A^-} = \sum_{\boldsymbol{\beta}\in A^-} \mathcal{S}^+_{A^+,\boldsymbol{\beta}}.$$

(2) *The dual SONC cone is the set*
$$\check{\mathcal{S}}^+_{A^+,A^-} = \bigcap_{\boldsymbol{\beta}\in A^-} \check{\mathcal{S}}^+_{A^+,\boldsymbol{\beta}}.$$

Proof. The first statement is a direct consequence of Theorem 3.3. For the second statement note that Minkowski sum and intersection are dual operations; see, e.g., [28, Theorem 1.6.3]. □

In particular, this corollary tells us that every nonnegative AGE function is a sum of nonnegative circuit functions.

In order to finally prove Theorem 3.2 we need another statement, which essentially combines Lemma 3.6 and a part of the proof of Proposition 3.9 in [18].

Lemma 3.5 ([18]). *For $\boldsymbol{\beta} \in A^-$, the dual cone of nonnegative circuit functions $\check{\mathcal{S}}^+_{A^+,\boldsymbol{\beta}}$ consists of those $\boldsymbol{v} \in \check{\mathbb{R}}^A$, where $v_{\boldsymbol{\alpha}} \geq 0$ for all $\boldsymbol{\alpha} \in A^+$, $v_{\boldsymbol{\alpha}} = 0$ for all $\boldsymbol{\alpha} \in A^- \setminus \{\boldsymbol{\beta}\}$ and one of the following equivalent conditions hold:*

(1) $\ln(|v_{\boldsymbol{\beta}}|) \leq \sum_{\boldsymbol{\alpha}\in A^+} \lambda_{\boldsymbol{\alpha}} \ln(v_{\boldsymbol{\alpha}})$ *for all* $\boldsymbol{\lambda} \in \Lambda(A^+, \boldsymbol{\beta})$.

(2) *There exists* $\boldsymbol{\tau} \in \mathbb{R}^n$ *such that for all* $\boldsymbol{\alpha} \in A^+$: $\ln\left(\frac{|v_{\boldsymbol{\beta}}|}{v_{\boldsymbol{\alpha}}}\right) \leq (\boldsymbol{\alpha}-\boldsymbol{\beta})^T \boldsymbol{\tau}$.

Proof of Theorem 3.2. The statement follows by Corollary 3.4 and Lemma 3.5 . Namely, the first representation can be deduced from (1), and the second one from (2). □

3.2 The Dual SONC Cone is Contained in the Primal SONC Cone

For $A = A^+ \cup A^-$ defined as in (5), we identified the dual space of exponential sums supported on A with $\check{\mathbb{R}}^A$. Now we use the reverse identification. For every $\boldsymbol{v} \in \check{\mathbb{R}}^A$ we associate a function

$$f(\mathbf{x}) = \sum_{\boldsymbol{\alpha}\in A^+} v_{\boldsymbol{\alpha}} e^{\langle \mathbf{x}, \boldsymbol{\alpha}\rangle} + \sum_{\boldsymbol{\beta}\in A^-} v_{\boldsymbol{\beta}} e^{\langle \mathbf{x}, \boldsymbol{\beta}\rangle}. \tag{8}$$

Note that circuit functions and AGE-functions are special cases of these functions. With this consideration, we identify the dual cone $\check{\mathcal{S}}^+_{A^+,\boldsymbol{\beta}}$ of nonnegative circuit functions having exponents in $A^+ \cup \{\boldsymbol{\beta}\}$ with the cone of all functions of the form (8) having coefficients in $\check{\mathcal{S}}^+_{A^+,\boldsymbol{\beta}}$. In order to keep notation short, we write $\check{\mathcal{S}}^+_{A^+,\boldsymbol{\beta}}$ for this cone as well. For the cone $\check{\mathcal{S}}^+_{A^+,A^-}$ we use the same identification with the notation $\check{\mathcal{S}}^+_{A^+,A^-}$.

Proposition 3.6. *It holds that*

(1) $\check{\mathcal{S}}^+_{A^+,\boldsymbol{\beta}} \subseteq \mathcal{S}^+_{A^+,\boldsymbol{\beta}}$.

(2) $\check{\mathcal{S}}^+_{A^+,A^-} \subseteq \mathcal{S}^+_{A^+,A^-}$.

In particular, every function of the form (8) with coefficients in $\check{\mathcal{S}}^+_{A^+,\boldsymbol{\beta}}$ or $\check{\mathcal{S}}^+_{A^+,A^-}$ is nonnegative.

We point out that Proposition 3.6 was already observed by Katthän, Theobald, and the third author in [18, Remark 3.7] without providing a proof.

Proof.

(1) Let $f \in \check{\mathcal{S}}^+_{A^+,\boldsymbol{\beta}}$ with a corresponding vector of coefficients $\boldsymbol{v} \in \mathbb{R}^A$. By representation (2) of Theorem 3.2, we have $v_{\boldsymbol{\alpha}} \geq 0$ for all $\boldsymbol{\alpha} \in A^+$ and for all $\boldsymbol{\lambda} \in \Lambda(A^+, \boldsymbol{\beta})$ it holds that

$$\ln(|v_{\boldsymbol{\beta}}|) \leq \sum_{\boldsymbol{\alpha}\in A^+} \lambda_{\boldsymbol{\alpha}} \ln(v_{\boldsymbol{\alpha}}) \leq \sum_{\boldsymbol{\alpha}\in A^+, \lambda_{\boldsymbol{\alpha}}>0} \lambda_{\boldsymbol{\alpha}} \ln\left(\frac{v_{\boldsymbol{\alpha}}}{\lambda_{\boldsymbol{\alpha}}}\right) = \ln(\Theta_f),$$

where Θ_f denotes the circuit number of f. The last inequality holds as $\lambda_{\boldsymbol{\alpha}} \in [0,1]$ for every $\boldsymbol{\alpha} \in A^+$ and the logarithmic function is monotonically increasing. Thus, $-v_{\boldsymbol{\beta}} \leq |v_{\boldsymbol{\beta}}| \leq \Theta_f$. Applying Theorem 2.6 we obtain the claimed result.

(2) By Definition 2.4, Definition 3.1 and part (1), we obtain

$$\check{\mathcal{S}}^+_{A^+,A^-} \subseteq \check{\mathcal{S}}^+_{A^+,\boldsymbol{\beta}} \subseteq \mathcal{S}^+_{A^+,\boldsymbol{\beta}} \subseteq \mathcal{S}^+_{A^+,A^-}.$$

□

We remark that the reverse implication does not hold in general.

Example 3.7. Consider the function $f(x) := 1 - 2e^x + e^{2x}$ with the sets $A^+ = \{0, 2\}$, $A^- = \{1\}$ and $v_0 = v_2 = 1, v_1 = -2$. As

$$1 = \frac{1}{2}\cdot 0 + \frac{1}{2}\cdot 2 \text{ and } -v_{\boldsymbol{\beta}} = |v_{\boldsymbol{\beta}}| = (2^{1/2})^2,$$

we have $f \in \mathcal{S}^+_{A^+,A^-}$. But since

$$\sum_{\boldsymbol{\alpha}\in A^+} \lambda_{\boldsymbol{\alpha}} \ln(v_{\boldsymbol{\alpha}}) = 2\left(\frac{1}{2}\ln(1)\right) = 0 < \ln(2) = \ln(|v_1|)$$

it follows that $f \notin \check{\mathcal{S}}^+_{A^+,A^-}$.

4 OPTIMIZING OVER THE DUAL SONC CONE VIA LINEAR PROGRAMMING

In this section, we obtain a computationally fast approximation of the global optimization problem

$$\inf_{\mathbf{x}\in\mathbb{R}^n} f(\mathbf{x}) \tag{9}$$

for exponential sums $f \in \mathbb{R}^A$ and $A = A^+ \cup A^-$ defined as in (5) via the representations of the dual SONC cone in Theorem 3.2.

4.1 Formulation of the Optimization Problem

First, we prove that deciding membership in the dual SONC cone can be done via linear programming.

Proposition 4.1. *Let*

$$f = \sum_{\boldsymbol{\alpha}\in A^+} v_{\boldsymbol{\alpha}} e^{\langle \mathbf{x}, \boldsymbol{\alpha}\rangle} + \sum_{\boldsymbol{\beta}\in A^-} v_{\boldsymbol{\beta}} e^{\langle \mathbf{x}, \boldsymbol{\beta}\rangle}$$

with $\boldsymbol{v} \in \check{\mathbb{R}}^A$ and $v_{\boldsymbol{\alpha}} \geq 0$ for every $\alpha \in \mathrm{Vert}(\mathrm{conv}(A))$.

The following linear feasibility program in $\#A^-$ many variables $(\boldsymbol{\tau}^{(\boldsymbol{\beta})})_{\boldsymbol{\beta}\in A^-}$ verifies containment in the dual SONC cone.

$$\ln\left(\frac{|v_{\boldsymbol{\beta}}|}{v_{\boldsymbol{\alpha}}}\right) \leq (\boldsymbol{\alpha} - \boldsymbol{\beta})^T \boldsymbol{\tau}^{(\boldsymbol{\beta})} \quad \text{for all } \boldsymbol{\beta} \in A^-,\ \boldsymbol{\alpha} \in A^+ \tag{10}$$

Proof. The program checks the conditions of Theorem 3.2(3). Note that the assumptions "$v_{\boldsymbol{\alpha}} \geq 0$ for every $\alpha \in \mathrm{Vert}(\mathrm{conv}(A))$" on f are necessary due to (4). As $\boldsymbol{v} \in \check{\mathbb{R}}^A$ is fixed, the inequalities are linear and hence (10) is a linear program. Moreover, $v_{\boldsymbol{\alpha}} \geq 0$ for every $\boldsymbol{\alpha} \in A^+$ holds by assumption (or we know trivially that f does not belong to the dual SONC cone). The last inequalities in Theorem 3.2(3) are satisfied trivially. □

In particular, fixing the non-auxiliary variables $\boldsymbol{v}$ in a lifted version of the dual cone forms a polyhedron; see Theorem 3.2 and Proposition 4.1.

To show that Proposition 4.1 can be used to obtain an exact linear optimization problem over the dual SONC cone, observe that equivalently to (9), we can solve the optimization problem

$$\min\left\{\gamma \ :\ f(\mathbf{x}) + \gamma \geq 0 \ \text{ for all } \ \mathbf{x} \in \mathbb{R}^n\right\}.$$

Instead of using containment in the SONC cone as a certificate for nonnegativity, i.e., solving

$$\min\left\{\gamma \ :\ f(\mathbf{x}) + \gamma \in \mathcal{S}^+_{A^+,A^-}\right\},$$

we use the dual cone $\check{\mathcal{S}}^+_{A^+,A^-}$. Recall that $\check{\mathcal{S}}^+_{A^+,A^-} \subseteq \mathcal{S}^+_{A^+,A^-}$ by Proposition 3.6. In particular, we do not dualize the LP to approximate the solution but optimize f to be a function in the dual cone instead of the primal cone. Hence, we compute

$$-\check{\gamma}^* = \min\left\{\check{\gamma} \ :\ \boldsymbol{v} + \check{\gamma}\cdot e_0 \in \check{\mathcal{S}}^+_{A^+,A^-}\right\}, \tag{11}$$

where $e_0 \in \mathbb{R}^A$ is the unit vector corresponding to $e^{\langle \mathbf{x}, \mathbf{0}\rangle}$, i.e., $v_0 + \check{\gamma}$ is the coefficient corresponding to $e^{\langle \mathbf{x}, \mathbf{0}\rangle}$.

Consider $\boldsymbol{v}$ to be given via

$$f(\mathbf{x}) + \check{\gamma} = \sum_{\boldsymbol{\alpha}\in A^+} v_{\boldsymbol{\alpha}} e^{\langle \mathbf{x}, \boldsymbol{\alpha}\rangle} + \sum_{\boldsymbol{\beta}\in A^-} v_{\boldsymbol{\beta}} e^{\langle \mathbf{x}, \boldsymbol{\beta}\rangle} + \check{\gamma} = \sum_{\boldsymbol{\alpha}\in A^+\setminus\{\mathbf{0}\}} v_{\boldsymbol{\alpha}} e^{\langle \mathbf{x}, \boldsymbol{\alpha}\rangle} + \sum_{\boldsymbol{\beta}\in A^-\setminus\{\mathbf{0}\}} v_{\boldsymbol{\beta}} e^{\langle \mathbf{x}, \boldsymbol{\beta}\rangle} + (v_{\mathbf{0}} + \check{\gamma}).$$

Note that the constant term $v_{\mathbf{0}}$ of $f(\mathbf{x})$ can be zero. By Theorem 3.2(3), and assuming $w_{\mathbf{0}} := \check{\gamma} + v_{\mathbf{0}}$ and $w_\alpha := v_\alpha$ for all $\alpha \in A \setminus \{\mathbf{0}\}$, solving (11) is equivalent to solving

$$\min\left\{\check{\gamma} : \begin{array}{c} \forall \boldsymbol{\alpha} \in A^+, w_{\boldsymbol{\alpha}} \geq 0 \text{ and } \forall \boldsymbol{\beta} \in A^- \exists \boldsymbol{\tau} \in \mathbb{R}^n, \\ \forall \boldsymbol{\alpha} \in A^+, \ln\left(\frac{|w_{\boldsymbol{\beta}}|}{w_{\boldsymbol{\alpha}}}\right) \leq (\boldsymbol{\alpha} - \boldsymbol{\beta})^T \boldsymbol{\tau} \end{array}\right\}. \tag{12}$$

Before stating the corresponding optimization program, we emphasize the fact that $\mathbf{0}$ is not necessarily contained in A, i.e., for the next result we need to include it either in A^+ or A^-, although we have to determine later to which one of the sets it belongs.

First, we prove several statements addressing this choice.

Lemma 4.2. *Let $A = A^+ \cup A^- \subseteq \mathbb{R}^n$ as in (5) and $f \in \check{\mathcal{S}}^+_{A^+,A^-}$ with $\mathbf{0} \in A$. If f is a polynomial, then $\mathbf{0} \in A^+$.*

Proof. For a polynomial f, we have $A \subseteq \mathbb{N}^n$. As $\mathbf{0} \in A$, we necessarily have $\mathbf{0} \in \mathrm{Vert}(\mathrm{conv}(A))$. With (4) and the fact that $\check{\mathcal{S}}^+_{A^+,A^-} \subseteq \mathcal{S}^+_{A^+,A^-}$, we obtain the statement. □

Lemma 4.3. *Let $\mathbf{0} \in A = A^+ \cup A^- \subseteq \mathbb{R}^n$ as in (5) and $f \in \check{\mathcal{S}}^+_{A^+,A^-}$ with coefficient vector $\boldsymbol{v} \in \check{\mathbb{R}}^A$. For the optimal lower bound $-\check{\gamma}^* \leq f(x)$ (as defined in (11)) and $c^* = \ln(|v_{\mathbf{0}} + \check{\gamma}^*|)$, we have*

$$-\check{\gamma}^* = \begin{cases} v_{\mathbf{0}} - e^{c^*} & \text{if } \mathbf{0} \in A^+ \\ v_{\mathbf{0}} + e^{c^*} & \text{if } \mathbf{0} \in A^-. \end{cases} \tag{13}$$

Proof. If $\mathbf{0} \in A^+$, we have $v_{\mathbf{0}} + \check{\gamma}^* \geq 0$ implying $|v_{\mathbf{0}} + \check{\gamma}^*| = v_{\mathbf{0}} + \check{\gamma}^*$. If $\mathbf{0} \in A^-$, we have $v_{\mathbf{0}} + \check{\gamma}^* < 0$ implying $|v_{\mathbf{0}} + \check{\gamma}^*| = -v_{\mathbf{0}} - \check{\gamma}^*$. This yields the statement. □

From now on, for $A = A^+ \cup A^-$ defined as in (5) and a fixed exponential function

$$f = \sum_{\boldsymbol{\alpha} \in A^+ \setminus \{\mathbf{0}\}} v_{\boldsymbol{\alpha}} e^{\langle \mathbf{x}, \boldsymbol{\alpha} \rangle} + \sum_{\boldsymbol{\beta} \in A^- \setminus \{\mathbf{0}\}} v_{\boldsymbol{\beta}} e^{\langle \mathbf{x}, \boldsymbol{\beta} \rangle} + v_{\mathbf{0}},$$

where for all $\boldsymbol{\alpha} \in \operatorname{Vert}(\operatorname{conv}(A))$ we have $v_{\boldsymbol{\alpha}} \geq 0$ (i.e., f satisfies (4)), with lower bound $-\check{\gamma}^*$, we consider the following two linear programs in $\#A^- + 1$ variables $(\boldsymbol{\tau}^{(\boldsymbol{\beta})})_{\boldsymbol{\beta} \in A^-}$ and $c = \ln(|v_{\mathbf{0}} + \check{\gamma}|)$.

$$\begin{aligned} &\min\ c && (\mathrm{LP}_{A^+}) \\ &(1) \quad \forall\, \boldsymbol{\beta} \in A^-, \forall \boldsymbol{\alpha} \in A^+ \setminus \{\mathbf{0}\} : \\ \text{s. t.} \quad &\ln\left(\tfrac{|v_{\boldsymbol{\beta}}|}{v_{\boldsymbol{\alpha}}}\right) \leq (\boldsymbol{\alpha} - \boldsymbol{\beta})^T \boldsymbol{\tau}^{(\boldsymbol{\beta})}, \\ &(2) \quad \ln\left(|v_{\boldsymbol{\beta}}|\right) - c \leq (-\boldsymbol{\beta})^T \boldsymbol{\tau}^{(\boldsymbol{\beta})} \ \ \forall\, \boldsymbol{\beta} \in A^- \end{aligned}$$

if $\mathbf{0} \in A^+$ and

$$\begin{aligned} &\min\ c && (\mathrm{LP}_{A^-}) \\ &(1) \quad \forall\, \boldsymbol{\beta} \in A^- \setminus \{\mathbf{0}\}, \forall \boldsymbol{\alpha} \in A^+ : \\ \text{s. t.} \quad &\ln\left(\tfrac{|v_{\boldsymbol{\beta}}|}{v_{\boldsymbol{\alpha}}}\right) \leq (\boldsymbol{\alpha} - \boldsymbol{\beta})^T \boldsymbol{\tau}^{(\boldsymbol{\beta})}, \\ &(2) \quad c - \ln(v_{\boldsymbol{\alpha}}) \leq \boldsymbol{\alpha}^T \boldsymbol{\tau}^{(\mathbf{0})} \ \ \forall\, \boldsymbol{\alpha} \in A^+ \end{aligned}$$

if $\mathbf{0} \in A^-$.

Lemma 4.4. *Let*

$$f = \sum_{\boldsymbol{\alpha} \in A^+ \setminus \{\mathbf{0}\}} v_{\boldsymbol{\alpha}} e^{\langle \mathbf{x}, \boldsymbol{\alpha} \rangle} + \sum_{\boldsymbol{\beta} \in A^- \setminus \{\mathbf{0}\}} v_{\boldsymbol{\beta}} e^{\langle \mathbf{x}, \boldsymbol{\beta} \rangle} + v_{\mathbf{0}} e^{\langle \mathbf{x}, \mathbf{0} \rangle},$$

with $v_{\mathbf{0}} \neq -\check{\gamma}^*$ and $A = A^+ \cup A^-$ defined as in (5). At least on of the linear programs (LP_{A^+}) and (LP_{A^-}) has a solution for its corresponding assumption

(1) $\mathbf{0} \in A^+$ *or*
(2) $\mathbf{0} \in A^-$,

if and only if there exists some $\check{\gamma} \in \mathbb{R}$ such that $f + \check{\gamma} \in \check{\mathcal{S}}^+_{A^+,A^-}$.

For either assumption, the corresponding LP is infeasible if and only if for all $\check{\gamma} \in \mathbb{R}$ we have $f + \check{\gamma} \notin \check{\mathcal{S}}^+_{A^+,A^-}$.

Proof. Consider $f + \check{\gamma}^*$. As $v_{\mathbf{0}} \neq -\check{\gamma}^*$ we have that $\mathbf{0} \in A$. Hence, the inequalities are exactly the inequalities in Theorem 3.2, except for the fact that we use c instead of $\ln(v_{\mathbf{0}})$ due to the former substitution. □

We need to omit $v_{\mathbf{0}} = -\check{\gamma}^*$, because in this case the programs (1) and (2) in Lemma 4.4 are infeasible and unbounded, respectively. To still obtain a lower bound on the function f, one can verify containment in the dual SONC cone by testing feasibility via (10). If f is indeed an element in the dual SONC cone, then 0 is always a lower bound, but not necessarily the optimal bound on $\check{\mathcal{S}}^+_{A^+,A^-}$.

From the considerations above and Proposition 4.1 we can draw the following result.

Theorem 4.5. *Let*

$$f = \sum_{\boldsymbol{\alpha} \in A^+ \setminus \{\mathbf{0}\}} v_{\boldsymbol{\alpha}} e^{\langle \mathbf{x}, \boldsymbol{\alpha} \rangle} + \sum_{\boldsymbol{\beta} \in A^- \setminus \{\mathbf{0}\}} v_{\boldsymbol{\beta}} e^{\langle \mathbf{x}, \boldsymbol{\beta} \rangle} + v_{\mathbf{0}} e^{\langle \mathbf{x}, \mathbf{0} \rangle},$$

with $v_{\mathbf{0}} \neq -\check{\gamma}^*$ and $A = A^+ \cup A^-$ defined as in (5) and let $-\check{\gamma}^* \neq v_{\mathbf{0}}$ be the optimal value with $f \geq -\gamma *$ as defined in (11). The linear programs (LP_{A^+}) and (LP_{A^-}) solve the optimization problem (12).

Proof. We set $A := A \cup \{\mathbf{0}\}$. First, note that we do not know the value of $v_{\mathbf{0}} + \check{\gamma}^*$ before computing the optimal value, and particularly we do not know the sign of $v_{\mathbf{0}} + \check{\gamma}^*$. Thus, we cannot determine whether $\mathbf{0} \in A^+$ or $\mathbf{0} \in A^-$ before computing the optimal value.

As we made the assumption $v_{\mathbf{0}} \neq -\check{\gamma}^*$, according to Lemma 4.4, at least one of the problems (LP_{A^+}) and (LP_{A^-}) is feasible if and only if $f \in \check{\mathcal{S}}^+_{A^+,A^-}$. In the case that only one linear program is feasible, $\mathbf{0}$ is contained in the corresponding set and hence, this program yields the optimal value. If both programs are feasible, there exist $\check{\gamma}_1$ and $\check{\gamma}_2$ such that $v_{\mathbf{0}} + \check{\gamma}_1$ is nonnegative and $f + \check{\gamma}_1 \cdot e^{\langle \mathbf{x}, \mathbf{0} \rangle} \in \check{\mathcal{S}}^+_{A^+,A^-}$ for $\mathbf{0} \in A^+$, and $v_{\mathbf{0}} + \check{\gamma}_2$ is negative and $f + \check{\gamma}_2 \cdot e^{\langle \mathbf{x}, \mathbf{0} \rangle} \in \check{\mathcal{S}}^+_{A^+,A^-}$ for $\mathbf{0} \in A^-$.

Thus, we select the linear program which yields the better bound.

According to Lemma 4.3, the lower bound on the dual SONC cone is

$$-\check{\gamma}^* = \begin{cases} v_{\mathbf{0}} - e^{c^*} & \text{if } \mathbf{0} \in A^+ \\ v_{\mathbf{0}} + e^{c^*} & \text{if } \mathbf{0} \in A^- \end{cases}. \tag{14}$$

□

Note that optimizing over the dual cone does not yield the actual optimal value in every case. Consider for example the Motzkin polynomial

$$f(x, y) = x^2y^4 + x^4y^2 - 3x^2y^2 + 1. \tag{15}$$

This is a nonnegative polynomial on $\mathbb{R}^2$ with $\inf_{(x,y) \in \mathbb{R}^2} f(x, y) = 0$. Since in the polynomial case we always need $\mathbf{0} \in A^+$, the linear program (LP_{A^+}) for f is the following:

$$\begin{aligned} &\min\ c \\ \text{such that}\ &\ln(3) \leq 2\tau_2 \\ &\ln(3) \leq 2\tau_1 \\ &\ln(3) + 2\tau_1 + 2\tau_2 \leq c, \end{aligned}$$

returning the lower bound $f \geq -26$ on $\mathbb{R}^2$.

4.2 Numerical results

In what follows we present the results of numerical experiments of several examples.

Any LP solver can be used to solve the optimization problem in Theorem 4.5. Here, we used cvxpy [2, 7]; see also [1], in the software POEM [30] available at

http://www.iaa.tu-bs.de/AppliedAlgebra/POEM/

on a `Intel(R) Core(TM) i7-8700 CPU` with 3.20 GHz and 15 GB of RAM.

To compare our approach with existing results, we restrict our computations to the polynomial case, i.e., the case $A \subseteq \mathbb{N}^n$. Note

that in this setting the convex hull conv(supp(p)) of exponents of a polynomial $p \in \mathbb{R}^A$ is commonly referred to as the *Newton polytope* of p. We use selected examples, mainly from [29], to demonstrate our findings. Those polynomials that are not explicitly stated in the examples can be found online via

https://www3.math.tu-berlin.de/combi/RAAGConOpt/comparison_paper/.

The value opt computed here corresponds to $\check{\gamma}^*$ as described in (13) in the dual case and to a γ^* with $p(\mathbf{x}) \geq -\gamma^*$ in the SOS, SAGE and SONC case. Hence, a smaller value for opt means a better lower bound to the polynomial. To compute this lower bound we need to make a sign change.

In the examples that follow, we denote by "SAGE" the bound computed via solving the REP introduced by Chandrasekaran and Shah [6], which provides the optimal primal SONC/SAGE bound. With "SONC" we denote the covering algorithm for SONC described in [29, Algorithm 3.4]. This algorithm solves a GP providing a lower bound for the (optimal) primal SONC/SAGE bound, but (experimentally) with better runtimes and numerical behavior. Thus, it is in particular possible that the bound "SONC" is worse than the bound "Dual SONC". The bound "Dual SONC", however, is always at most as good as "SAGE", the optimal primal SONC/SAGE bound (if both bounds can be computed successfully).

The examples are chosen in a way to display that it depends on the particular instance, which approach yields the best bound or has the best runtime respectively.

Example 4.6 ([29], Example 4.1). Consider the following polynomial of degree 8 in two variables with three interior points.

$$p = 1 + 3 \cdot x_0^2 x_1^6 + 2 \cdot x_0^6 x_1^2 + 6 \cdot x_0^2 x_1^2 - 1 \cdot x_0^1 x_1^2 - 2 \cdot x_0^2 x_1^1 - 3 \cdot x_0^3 x_1^3$$

As expected, the bound returned by our dual approach is worse than the one computed via SONC and SAGE, but it is computed faster; see Table 1. The sum of squares (SOS) approach does not yield a result.

strategy	time	opt
SONC	0.02254	0.72732
SAGE	0.03820	−0.69316
SOS	0.30280	inf
Dual SONC	0.02706	4.51135

Table 1: Example 4.6: A polynomial in two variables of degree 8 with three inner terms.

Example 4.7.

$$p = -3 + 1.5 \cdot x_1^6 + 11.5 \cdot x_0^6 - 0.5 \cdot x_1^2 + 0.5 \cdot x_0^4$$

In this example all tested approaches yield similar results; see Table 2.

Since the SONC approach does, in general, not compute the optimal bound of a polynomial on the primal SONC cone, it is also possible that our approach yields better results. This is demonstrated in the following example.

strategy	time	opt
SONC	0.01458	3.11111
SAGE	0.01658	3.11111
SOS	0.12911	3.11111
Dual SONC	0.01391	3.28868

Table 2: Example 4.7: A polynomial in two variables of degree 6.

Example 4.8 ([29], Example 4.2). Consider a polynomial whose Newton polytope is a standard simplex with $n = 10$, $d = 30$, and 200 terms. The bound computed with the dual approach is much better than the one found via SONC. The SAGE approach yields no result, the computations for the SOS approach were aborted after 60 minutes; see Table 3.

strategy	time	opt
SONC	0.90452	1109.45
SAGE	72.90220	inf
SOS	> 3600	–[1]
Dual SONC	4.21717	−35.25153

Table 3: Example 4.8: A polynomial in 10 variables of degree 30, where conv(A) is the standard simplex.

Example 4.9 ([29], Example 4.5). Consider a polynomial with the computationally challenging *dwarfed cube*; see [3], in dimension 7 as its Newton polytope. For this polynomial, Equation (LP_{A^+}) is infeasible. The SAGE approach also fails; see Table 4.

strategy	time	opt
SONC	0.34685	−28.2779
SAGE	3.19388	inf
SOS	629.07700	−28.3181
Dual SONC	–	inf

Table 4: Example 4.9: A polynomial supported on the 7-dimensional dwarfed cube, scaled by a factor 4, with 63 inner terms.

To further illustrate the case of infeasibility in our linear program, consider the following example.

Example 4.10. Consider a polynomial supported on the dwarfed cube in dimension 2 with two additional interior points.

$$p = 0.5 \cdot x_0^2 x_1^4 + 2 \cdot x_0^4 + 1 \cdot x_0^4 x_1^2 + 2 + 2 \cdot x_1^4 - 1.0 \cdot x_0^1 x_1^1 - c \cdot x_0^3 x_1^1.$$

If we choose $c = 3$, Equation (LP_{A^+}) will be infeasible. For $c = 1$, however, we get the results presented in Table 5.

[1] Aborted after 60 minutes.

strategy	time	opt
SONC	0.02835	−1.58558
SAGE	0.02907	−1.92193
SOS	0.08322	−1.92193
Dual SONC	0.02486	0.37055

Table 5: Example 4.10: A polynomial supported on the dwarfed cube in dimension 2.

Example 4.11. Consider a polynomial supported on *Kirkman's Icosahedron*; see [13], with three additional interior points. While SOS approach was aborted after exceeding a runtime of 60 minutes, the dual and primal SONC approach and the SAGE approach all yield results very quickly; see Table 6.

$$\begin{aligned} p \;=\; & 0.5924068000899325x_0^{42}x_1^{36}x_2^{36} + 0.9040680744391449x_0^{6}x_1^{36}x_2^{36} \\ & + 0.6286297557636527x_0^{42}x_1^{12}x_2^{36} + 0.22136661817072706x_0^{42}x_1^{36}x_2^{12} \\ & + 1.9921397074037133x_0^{6}x_1^{12}x_2^{36} + 2.4444012612478447x_0^{42}x_1^{12}x_2^{12} \\ & + 0.7745809478318744x_0^{6}x_1^{36}x_2^{12} + 0.4168575879720979x_0^{6}x_1^{12}x_2^{12} \\ & + 2.131772858737973x_0^{48}x_1^{32}x_2^{24} + 0.5582642102257477x_1^{32}x_2^{24} \\ & + 0.39948625235355123x_0^{48}x_1^{16}x_2^{24} + 1.055352501861479x_1^{16}x_2^{24} \\ & + 0.5862781645697882x_0^{24}x_1^{48}x_2^{40} + 0.8297411574785997x_0^{24}x_2^{40} \\ & + 1.5885016970170502x_0^{24}x_1^{48}x_2^{8} + 0.5937153134426314x_0^{24}x_2^{8} \\ & + 0.7427966893909136x_0^{36}x_1^{24}x_2^{48} + 0.9341646224001856x_0^{12}x_1^{24}x_2^{48} \\ & + 0.48065798662872594x_0^{36}x_1^{24} + 0.6729615719188968x_0^{12}x_1^{24} \\ & - 1.477600441785058x_0^{9}x_1^{6}x_2^{1} - 0.1791748172699452x_0^{3}x_1^{4}x_2^{5} \\ & - 0.27468070265719946x_0^{9}x_1^{3}x_2^{7}. \end{aligned}$$

strategy	time	opt
SONC	4.22899	1.00391
SAGE	0.155045	0.50104
SOS	> 3600	–[2]
Dual SONC	0.15034	2.00542

Table 6: Example 4.11: A polynomial supported on Kirkman's Icosahedron.

We conclude that, as expected (compare (15)), the linear program optimizing over the dual of the SONC cone yields, in general, worse results than the SOS, and the primal SONC/SAGE approach. Since it only relies on solving LPs it is, however, less vulnerable to numerical issues than the primal SONC/SAGE approach (which is relying on GP/REP) and of course, than the SOS approach (relying on SDP). It yields promising runtimes, and it gives a result whenever a solution in the dual cone exists.

In particular, we obtain an algorithm which yields a bound *independent* of the existing primal SONC and SAGE bounds.

[2] Aborted after 60 minutes.

REFERENCES

[1] Agrawal, A., Diamond, S., and Boyd, S. Disciplined geometric programming. *Optimization Letters 13* (2019), 961–976.

[2] Agrawal, A., Verschueren, R., Diamond, S., and Boyd, S. A rewriting system for convex optimization problems. *Journal of Control and Decision 5*, 1 (2018), 42–60.

[3] Avis, D., Bremner, D., and Seidel, R. How good are convex hull algorithms? *Computational Geometry 7*, 5-6 (1997), 265–301.

[4] Blekherman, G., Parrilo, P., and Thomas, R. *Semidefinite Optimization and Convex Algebraic Geometry*, vol. 13 of *MOS-SIAM Series on Optimization*. SIAM and the Mathematical Optimization Society, Philadelphia, 2013.

[5] Boyd, S., Kim, S.-J., Vandenberghe, L., and Hassibi, A. A tutorial on geometric programming. *Optim. Eng. 8*, 1 (2007), 67–127.

[6] Chandrasekaran, V., and Shah, P. Relative entropy relaxations for signomial optimization. *SIAM J. Optim. 26*, 2 (2016), 1147–1173.

[7] Diamond, S., and Boyd, S. CVXPY: A Python-embedded modeling language for convex optimization. *Journal of Machine Learning Research 17*, 83 (2016), 1–5.

[8] Dressler, M., Iliman, S., and de Wolff, T. A Positivstellensatz for Sums of Nonnegative Circuit Polynomials. *SIAM J. Appl. Algebra Geom. 1*, 1 (2017), 536–555.

[9] Dressler, M., Iliman, S., and de Wolff, T. An approach to constrained polynomial optimization via nonnegative circuit polynomials and geometric programming. *J. Symb. Comput. 91* (2019), 149–172.

[10] Dressler, M., Naumann, H., and Theobald, T. The dual cone of sums of non-negative circuit polynomials, 2018. Preprint, arXiv:1809.07648.

[11] Duffin, R., and Peterson, E. Geometric programming with signomials. *J. Optim. Theory Appl. 11* (1973), 3–35.

[12] Feliu, E., Kaihnsa, N., Yürük, O., and de Wolff, T. The kinetic space of multistationarity in dual phosphorylation, 2020. Preprint, arXiv:2001.08285.

[13] Fetter, H. L. A polyhedron full of surprises. *Mathematics Magazine 85*, 5 (2012), 334–342.

[14] Forsgård, J., and de Wolff, T. The algebraic boundary of the sonc cone, 2019. Preprint, arXiv:1905.04776.

[15] Gelfand, I. M., Kapranov, M. M., and Zelevinsky, A. V. *Discriminants, resultants and multidimensional determinants*. Modern Birkhäuser Classics. Birkhäuser Boston Inc., Boston, MA, 1994.

[16] Iliman, S., and de Wolff, T. Amoebas, nonnegative polynomials and sums of squares supported on circuits. *Res. Math. Sci. 3* (2016), 3:9.

[17] Iliman, S., and de Wolff, T. Lower bounds for polynomials with simplex newton polytopes based on geometric programming. *SIAM J. Optim. 26*, 2 (2016), 1128–1146.

[18] Katthän, L., Naumann, H., and Theobald, T. A unified framework of SAGE and SONC polynomials and its duality theory, 2019. Preprint, arXiv:1903.08966.

[19] Lasserre, J. Global optimization with polynomials and the problem of moments. *SIAM J. Optim. 11*, 3 (2000/01), 796–817.

[20] Lasserre, J. *Moments, positive polynomials and their applications*. London: Imperial College Press, 2010.

[21] Lasserre, J. *An Introduction to Polynomial and Semi-Algebraic Optimization*, vol. 1 of *Cambridge Texts in Applied Mathematics*. Cambridge University Press, 2015.

[22] Laurent, M. Sums of squares, moment matrices and optimization over polynomials. In *Emerging Applications of Algebraic Geometry*, vol. 149 of *IMA Vol. Math. Appl.* Springer, New York, 2009, pp. 157–270.

[23] Loera, J. D., Rambau, J., and Santos, F. *Triangulations*, vol. 25 of *Algorithms and Computation in Mathematics*. Springer-Verlag, Berlin, 2010. Structures for algorithms and applications.

[24] Murray, R., Chandrasekaran, V., and Wierman, A. Newton Polytopes and Relative Entropy Optimization, 2018. Preprint, arXiv:1810.01614.

[25] Murray, R., Chandrasekaran, V., and Wierman, A. Signomial and Polynomial Optimization via Relative Entropy and Partial Dualization, 2019. Preprint, arXiv:1907.00814.

[26] Parrilo, P. Structured semidefinite programs and semialgebraic geometry methods in robustness and optimization, 2000. PhD Thesis, California Institute of Technology.

[27] Reznick, B. Forms Derived from the Arithmetic-Geometric Inequality. *Math. Ann. 283* (1989), 431–464.

[28] Schneider, R. *Convex Bodies: the Brunn–Minkowski Theory*. Cambridge University Press, 2014.

[29] Seidler, H., and de Wolff, T. An experimental comparison of SONC and SOS certificates for unconstrained optimization, 2018. Preprint, arXiv:1808.08431.

[30] Seidler, H., and de Wolff, T. POEM: Effective methods in polynomial optimization, version 0.2.1.0. http://www.iaa.tu-bs.de/AppliedAlgebra/POEM/, jul 2019.

[31] Wang, J. Nonnegative polynomials and circuit polynomials, 2018. preprint, arXiv:1804.09455.

An Additive Decomposition in Logarithmic Towers and Beyond

Hao Du[1], Jing Guo[2], Ziming Li[2], Elaine Wong[1]
[1]Johann Radon Institute (RICAM), Austrian Academy of Sciences, Altenberger Straße 69, 4040, Linz, Austria
[2]Key Laboratory of Mathematics and Mechanization, AMSS, Chinese Academy of Sciences
School of Mathematical Sciences, University of Chinese Academy of Sciences, Beijing, 100190, China
hao.du@ricam.oeaw.ac.at, JingG@amss.ac.cn, zmli@mmrc.iss.ac.cn, elaine.wong@ricam.oeaw.ac.at

ABSTRACT

We consider the additive decomposition problem in primitive towers and present an algorithm to decompose a function in a certain kind of primitive tower which we call S-primitive, as a sum of a derivative in the tower and a remainder which is minimal in some sense. Special instances of S-primitive towers include differential fields generated by finitely many logarithmic functions and logarithmic integrals. A function in an S-primitive tower is integrable in the tower if and only if the remainder is equal to zero. The additive decomposition is achieved by viewing our towers not as a traditional chain of extension fields, but rather as a direct sum of certain subrings. Furthermore, we can determine whether or not a function in an S-primitive tower has an elementary integral without the need to deal with differential equations explicitly. We also show that any logarithmic tower can be embedded into a particular extension where we can further decompose the given function. The extension is constructed using only differential field operations without introducing any new constants.

KEYWORDS

Additive decomposition, Primitive tower, Logarithmic tower, Symbolic integration, Elementary integral

ACM Reference Format:
Hao Du[1], Jing Guo[2], Ziming Li[2], Elaine Wong[1]. 2020. An Additive Decomposition in Logarithmic Towers and Beyond. In *International Symposium on Symbolic and Algebraic Computation (ISSAC '20), July 20–23, 2020, Kalamata, Greece.* ACM, New York, NY, USA, 8 pages. https://doi.org/10.1145/3373207.3404025

1 INTRODUCTION

Given a differential ring $(\mathcal{R}, ')$ and an element $f \in \mathcal{R}$, we ask if the indefinite integral of f belongs to $\mathcal{R}$ and compute one if it does. In order to do this, we start with a decision problem stated as:

$$\text{Given } f \in \mathcal{R}, \text{ decide if } f \in \mathcal{R}', \text{ where } \mathcal{R}' := \{g' \mid g \in \mathcal{R}\}. \quad (1)$$

One can see that a positive answer to (1) tells us that a $g \in \mathcal{R}$ exists where $f = g'$ and then we proceed to compute such a g. The decision together with the computation is known as the integrability problem. If (1) produces a negative answer, then we say that f is *not integrable* in $\mathcal{R}$.

In the latter case, we would still like to be able to say something about the given function. Is there any information to help us understand how far off we are from being successful? The answer lies in the additive decomposition problem:

$$\text{Compute } g, r \in \mathcal{R} \text{ such that } f = g' + r,$$

where

(i) r is minimal in some sense;
(ii) $f \in \mathcal{R}'$ if and only if $r = 0$.

We call such an r a *remainder* of f in $\mathcal{R}$ and write

$$f \equiv r \mod \mathcal{R}'.$$

So, it is clear that an algorithm for solving the problem of additive decomposition also provides a solution to the integrability problem. Remainders may help us find "closed form" expressions for integrals of elements in $\mathcal{R}$, in the sense that the integrals belong to some extensions of $\mathcal{R}$. They also play an important role in reduction-based methods for creative telescoping.

The first additive decomposition due to Ostrogradsky [13] and Hermite [12] is for the differential field $\mathcal{F} = (\mathbb{C}(x), d/dx)$. Given a rational function $f \in \mathcal{F}$, they proposed an algorithm to compute the remainder $r \in \mathcal{F}$ of f such that r is proper and has a squarefree denominator, and r is minimal in the sense that if $f \equiv \tilde{r} \mod \mathcal{F}'$ for some $\tilde{r} \in \mathcal{F}$, then the denominator of r divides that of $\tilde{r}$.

There has been a rapid development of additive decompositions in both symbolic integration and summation [1, 3, 4, 6–9, 11, 16]. Most of the articles were motivated by computing telescopers based on reduction [2]. In the cited literature, some classes of functions that were studied include hyperexponential [3], algebraic [9], Fuchsian D-finite [7], and D-finite [16]. Additive decomposition problems in these classes have been fully solved. We observe that the ring of D-finite functions is not closed under composition or taking reciprocals. For example, $\log x$ is D-finite, but $\log(\log(x))$ and $1/\log(x)$ are not. In this paper, we consider a class of functions that is closed under these two operations.

Singer et al. in 1985 and then Raab in 2012 gave some decision procedures for finding elementary integrals in some Liouvillian extensions [14, 15] and in extensions which contain some nonlinear generators [14]. They recursively solve Risch differential equations until one of them has no solution, or else the integral can be found. In the implementation of Raab's algorithm, the former case outputs an integrable part and collects all nonzero terms that prevent the differential equations from having a solution. Recently, Chen, Du and Li [6] were able to construct remainders in some primitive extensions (they termed them "straight towers" and "flat towers") without the need to deal with differential equations explicitly.

ISSAC '20, July 20–23, 2020, Kalamata, Greece

ACM ISBN 978-1-4503-7100-1/20/07…$15.00
https://doi.org/10.1145/3373207.3404025

In this article, we expand their work [6] by developing a new algorithm to construct remainders for functions in "S-primitive towers" (see Definition 4.3), which may not be straight or flat. Instances for S-primitive towers include differential field extensions generated by finitely many logarithmic functions and logarithmic integrals.

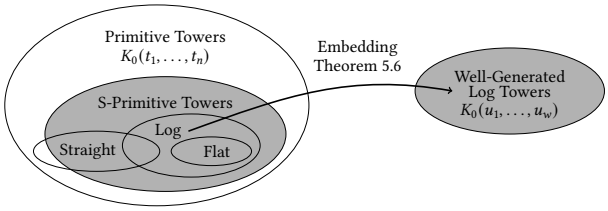


Figure 1: The gray ellipses on the left indicate the fields of functions for which we can construct a remainder. The embedding gives us a field extension ($n \le w$) where we can decompose functions further.

The organization of this article is as follows. In Sections 2 and 3, we give some relevant definitions associated to primitive towers, and then present a different way to view the towers. In Section 4, we give an algorithm for additive decompositions in S-primitive towers, and present a criterion for elementary integrability for the functions in such a field. In Section 5, we show how to construct a well-generated logarithmic tower to which a logarithmic tower can be embedded. Concluding remarks are given in Section 6.

2 PRELIMINARIES

Let K be a field of characteristic zero and $K(t)$ be the field of rational functions in t over K. An element of $K(t)$ is said to be *t-proper* if the degree of its denominator in t is higher than that of its numerator. In particular, zero is t-proper. For each $f \in K(t)$, there is a unique t-proper element $g \in K(t)$ and a unique polynomial $p \in K[t]$ with

$$f = g + p. \tag{2}$$

Let $'$ be a derivation on K. The pair $(K, ')$ is called a *differential field.* An element c of K is called a *constant* if $c' = 0$. The set of constants in K, denoted by C_K, is a subfield of K. Set

$$K' := \{f' \mid f \in K\},$$

which is a linear subspace over C_K. We call K' the *integrable subspace* of K.

Let (E, δ) be a differential field containing K. We say that E is a *differential field extension* of K if $\delta|_K = '$. The derivation δ is also denoted by $'$ when there is no confusion. For an element f of K, we call f a *logarithmic derivative* in K if $f = g'/g$ for some $g \in K\backslash\{0\}$. Let t be transcendental over K and $t' \in K[t]$, so that $p' \in K[t]$ for all $p \in K[t]$. A polynomial p in $K[t]$ is said to be *t-normal* if $\gcd(p, p') = 1$. By Theorem 3.2.2 in [5], $'$ can be uniquely extended to $K(t)$ such that $K(t)$ is a differential field extension of K. For $f \in K(t)$, we say that f is *t-simple* if it is t-proper and has a t-normal denominator.

We next define primitive and logarithmic generators, which are based on Definitions 5.1.1 and 5.1.2 in [5], respectively.

Definition 2.1. *Let $(K, ')$ be a differential field, and E be a differential field extension of K. An element t of E is said to be* primitive *over K if $t' \in K$. A primitive element t is called a* primitive generator *over K if it is transcendental over K and $C_{K(t)} = C_K$. Furthermore, a primitive generator t is called a* logarithmic generator *over K if t' is a C_K-linear combination of logarithmic derivatives in K.*

An immediate consequence of Theorem 5.1.1 in [5] is:

Proposition 2.2. *Let t be primitive over K. Then t is a primitive generator over K if and only if $t' \notin K'$. Assume that t is a primitive generator over K. Then $p \in K[t]$ is t-normal if and only if p is squarefree.*

For the rest of the section, assume that $(K, ')$ is a differential field, and that t is a primitive generator over K.

Remark 2.3. *Let p be a polynomial in $K[t]$. By Lemma 5.1.2 in [5], the degree of p' is equal to one less than the degree of p if the leading coefficient of p is a constant, otherwise their degrees are equal.*

By Theorem 5.3.1 in [5] and Lemma 2.1 in [6], for each $f \in K(t)$, there exists a unique t-simple element h such that

$$f \equiv h \mod \big(K(t)' + K[t]\big). \tag{3}$$

In the literature [6], h is referred to as the *Hermitian part* of f with respect to t. Thus, we will use the notation $\mathrm{hp}_t(f)$. It is easy to check that hp_t is a C_K-linear map on $K(t)$. Because of the uniqueness of Hermitian parts and Lemma 2.1 in [6], we have the following:

Lemma 2.4. *Let $f, g \in K(t)$. Then*

(i) $f \in K(t)' + K[t] \iff \mathrm{hp}_t(f) = 0$,
(ii) *f is t-simple* $\iff f = \mathrm{hp}_t(f)$, *and*
(iii) $f \equiv g \mod (K(t)' + K[t]) \iff \mathrm{hp}_t(f) = \mathrm{hp}_t(g)$.

The next two lemmas give some nice properties of proper elements and logarithmic derivatives.

Lemma 2.5. *If $f \in K(t)$ is t-proper, then $f - \mathrm{hp}_t(f) \in K(t)'$.*

Proof. Since t is a primitive generator over K, the derivative of a t-proper element of $K(t)$ is also t-proper. By (3), $f = \mathrm{hp}_t(f) + g' + p$ for some $g \in K(t)$ and $p \in K[t]$. Let r be the t-proper part of g. Thus, $f - \mathrm{hp}_t(f) - r' = p + (g - r)'$ whose left-hand side is t-proper and whose right-hand side is a polynomial in t. Thus, both sides must be zero. Consequently, $f - \mathrm{hp}_t(f) = r' \in K(t)'$. □

Lemma 2.6. *Let $f \in K(t)$ be a logarithmic derivative.*

(i) *f is t-proper $\iff$ f is t-simple.*
(ii) *There exists a t-simple logarithmic derivative $g \in K(t)$ and a logarithmic derivative $h \in K$ such that $f = g + h$.*

Proof. (i) The only thing we need to show is that the denominator of f is t-normal. By the logarithmic derivative identity [5, Theorem 3.1.1 (v)], the denominator of f is squarefree, which is also t-normal by Proposition 2.2.

(ii) By irreducible factorization and the logarithmic derivative identity, $f = \left(\sum_i m_i p_i'/p_i\right) + \alpha'/\alpha$, where $\alpha \in K$, $m_i \in \mathbb{Z}$, and $p_i \in K[t]$ are monic irreducible and pairwise coprime. Then each p_i'/p_i is t-simple by Remark 2.3 and (i). We get (ii) by setting $g = \sum_i m_i p_i'/p_i$ and $h = \alpha'/\alpha$. □

The following lemma will be useful when we construct our remainders. This is the same as Lemma 2.3 in [6] and can also be found in [5].

Lemma 2.7. *Let $p \in K[t]$. If $p \in K(t)'$, then the leading coefficient of p is equal to $ct' + b'$ for some $c \in C_K$ and $b \in K$. As a special case, if $p \in K \cap K(t)'$, then $p \equiv ct' \mod K'$.*

3 MATRYOSHKA DECOMPOSITIONS

We denote $\{1, 2, \ldots, n\}$ and $\{0, 1, 2, \ldots, n\}$ by $[n]$ and $[n]_0$, resp. Let K_0 be a field. For each $i \in [n]$, let $K_i = K_{i-1}(t_i)$, where t_i is transcendental over K_{i-1}. Then we have a tower of field extensions:

$$\begin{array}{ccccccc} K_0 & \subset & K_1 & \subset & \cdots & \subset & K_n \\ & & \| & & & & \| \\ & & K_0(t_1) & \subset & \cdots & \subset & K_{n-1}(t_n). \end{array} \tag{4}$$

We use $K_0(\bar{t})$ to denote the tower (4) generated by $\bar{t} := (t_1, \ldots, t_n)$.

For each $i \in [n]$, an element of K_n from (4) is said to be *t_i-proper* if it is free of $t_{i+1}, \ldots, t_n$ and the degree of its numerator in t_i is lower than that of its denominator. Let T_i denote the multiplicative monoid generated by $t_{i+1}, \ldots, t_n$ for all i with $0 \le i < n$, and set $T_n = \{1\}$. For each $i \in [n]$, let P_i be a non-unital subring of $K_i[t_{i+1}, \ldots, t_n]$ consisting of all the linear combinations of the elements of T_i whose coefficients are t_i-proper. Furthermore, let $P_0 = K_0[t_1, \ldots, t_n]$. A routine induction based on (2) shows

$$K_n = \bigoplus_{i=0}^{n} P_i. \tag{5}$$

Accordingly, we can not only view the tower as a chain of field extensions as described in (4), but also as a direct sum of rings as given in (5). The former enables us to describe a function recursively, and the latter helps us to decompose it additively.

Let π_i be the projection from K_n onto P_i with respect to (5). For every $f \in K_n$, we say $\pi_i(f)$ is the *i-th projection* of f, and write

$$f = \sum_{i=0}^{n} \pi_i(f),$$

which we call the *matryoshka decomposition* of f. Figure 2 illustrates this namesake. The property $P_i \cap \bigoplus_{i \neq j} P_j = \{0\}$ indicates zero as the (only) point of intersection, and is represented by a single dot in Figure 2. Viewing our towers in this way not only affords us a nice pictorial representation, but also allows us to describe the following ordering (which will later be used to define a remainder).

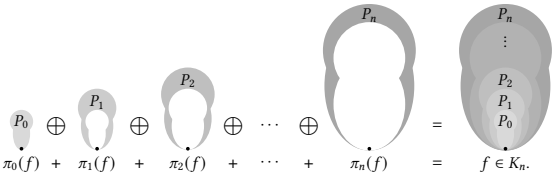


Figure 2: Matryoshka Decomposition

Example 3.1. *Let $f = (t_2 + x)(t_3^2 - t_1 t_3 + x t_2)/(x t_2 t_3)$ be in K_3 with $K_0 = \mathbb{Q}(x)$. Then the matryoshka decomposition of f is*

$$\pi_0(f) + \pi_1(f) + \pi_2(f) + \pi_3(f) = \frac{t_3 - t_1}{x} + 0 + \frac{t_3 - t_1}{t_2} + \frac{t_2 + x}{t_3}.$$

Suppose that $\prec$ is the purely lexicographic order on T_0, in which $t_1 \prec t_2 \prec \cdots \prec t_n$. Then $\prec$ is also a monomial order on each T_i, because $T_i \subseteq T_0$. For $f \in K_n$ and $i \in [n]_0$, the i-th projection of f can be viewed as a polynomial in $K_i[t_{i+1}, \ldots, t_n]$, which allows us to define the *i-th head monomial* of f, denoted by $\mathrm{hm}_i(f)$, to be the highest monomial in T_i that appears in $\pi_i(f)$ if $\pi_i(f)$ is non-zero, and zero if $\pi_i(f)$ is zero.

We define the *i-th head coefficient of f*, denoted by $\mathrm{hc}_i(f)$, to be the coefficient of $\mathrm{hm}_i(f)$ in $\pi_i(f)$ if $\pi_i(f)$ is non-zero, and zero if $\pi_i(f)$ is zero. By the matryoshka decomposition, $\mathrm{hc}_i(f)$ is t_i-proper for all $i \in [n]$.

The *head monomial* of f, denoted by $\mathrm{hm}(f)$, is defined to be the highest monomial among $\mathrm{hm}_0(f), \mathrm{hm}_1(f), \ldots, \mathrm{hm}_n(f)$, in which zero is regarded as the lowest "monomial". Let

$$I_f = \{i \in [n]_0 \mid \mathrm{hm}_i(f) = \mathrm{hm}(f)\}.$$

The *head coefficient* of f, $\mathrm{hc}(f)$, is defined to be $\sum_{i \in I_f} \mathrm{hc}_i(f)$.

Definition 3.2. *For $f, g \in K_n$, let d_f and d_g be the degrees of the denominators of f and g in t_n, respectively. We say that f is* lower than *g, denoted by $f \prec g$, if either $d_f < d_g$, or $d_f = d_g$ and $\mathrm{hm}(f) \prec \mathrm{hm}(g)$. We say that f is* not higher than *g, denoted by $f \preceq g$, if either $f \prec g$, or $d_f = d_g$ and $\mathrm{hm}(f) = \mathrm{hm}(g)$.*

For the rest of this article, we assume that $(K_0, ') = (C(x), d/dx)$ and each t_i in (4) is a primitive generator over K_{i-1} for all $i \in [n]$. Then we call K_n a *primitive extension* over K_0 and $K_0(\bar{t})$ a *primitive tower*. By Definition 2.1, $C_{K_n} = C_{K_0}$, which is equal to C. A primitive tower is said to be *logarithmic* if each t_i is a logarithmic generator over K_{i-1}.

Since $\prec$ on T_0 is a Noetherian total order, the partial order on K_n given by Definition 3.2 is also Noetherian, that is, every nonempty set in K_n has a minimal element with respect to $\prec$. We can use this order to define a desired remainder of the given function. Let $f \in K_n$ and

$$R_f := \{g \in K_n \mid g \equiv f \mod K_n'\}. \tag{6}$$

Thus, there exists a minimal element $r \in R_f$. We note that such a minimal element may not be unique. Furthermore, $\preceq$ is not a partial order, but rather a total preorder. Therefore, a minimal element of R_f with respect to $\prec$ is in fact a least element w.r.t. $\preceq$.

Definition 3.3. *Given $f \in K_n$, a minimal element of R_f is said to be a* remainder *of f. Moreover, let $r \in K_n$. Then we say that r is a remainder if r is a remainder of itself.*

As usual, t_i-simple elements play an important role when we construct remainders. Before we move on to the next section, we give a definition using the matryoshka decomposition.

Definition 3.4. *An element $f \in K_n$ is said to be* simple *if $\pi_i(f)$ is t_i-simple for all $i \in [n]_0$, where $t_0 = x$.*

Proposition 3.5. *Every logarithmic derivative in K_n is simple.*

Proof. We proceed by induction on n. Since every logarithmic derivative in K_0 is t_0-proper, the assertion holds for $n = 0$ by Lemma 2.6 (i). Assume that $n > 0$ and the assertion holds for $n - 1$. Let $f \in K_n$ be a logarithmic derivative. By Lemma 2.6 (ii), there exists a t_n-simple logarithmic derivative g and a logarithmic derivative $h \in K_{n-1}$ such that $f = g + h$. Then $\pi_n(f) = g$ by (2). Applying the induction hypothesis to h completes the induction. □

4 AN ADDITIVE DECOMPOSITION

Remainders in a tower are described in terms of minimality, which is not constructive. In this section, we will present an algorithm for constructing a remainder in an S-primitive tower based on Hermite reduction and integration by parts. To know when to terminate the algorithm, we need to be able to identify the first generator present in a given monomial (this is the same notion as *scale* in [6]).

Definition 4.1. *For a monomial $M = t_1^{d_1} \cdots t_n^{d_n} \in T_0$, the* indicator *of M, denoted by $\mathrm{ind}_n(M)$, is defined to be n if $M = 1$, or defined to be $\min\{i \in [n] \mid d_i \neq 0\}$.*

For $M \in T_0$, we set $K_n^{(\prec M)} := \{f \in K_n \mid \mathrm{hm}(f) \prec M\}$. Note that $K_n^{(\prec M)}$ is a C-linear subspace of K_n. The following lemma describes sufficient conditions for reducing a given term in a primitive tower with respect to $\prec$ via integration by parts.

Lemma 4.2. *Let $K_0(\bar{t})$ be a primitive tower, and $M \in T_0$ with indicator m. Then $(f+ct_m)'M$ belongs to $K_n' + K_n^{(\prec M)}$ for all $f \in K_{m-1}$ and $c \in C$.*

Proof. Let $M = t_m^{d_m} \cdots t_n^{d_n}$ for $d_m, \ldots, d_n \in \mathbb{N}$. Since K_n is a primitive extension over K_0, we see that $t_j' \in K_{j-1}$ for each j with $m \leq j \leq n$. Then $M' = \sum_{j=m}^{n} h_j N_j$, where $h_j \in K_{j-1}$, and

$$N_j = \begin{cases} t_j^{d_j - 1} t_{j+1}^{d_{j+1}} \cdots t_n^{d_n} & d_j > 0, \\ 0 & d_j = 0. \end{cases}$$

So $fh_j \in K_{j-1}$ and $N_j \prec M$. Consequently, $fh_jN_j \prec M$ for all j with $m \leq j \leq n$. We see that $fM' \prec M$, which, together with $f'M = (fM)' - fM'$, implies that

$$f'M \in K_n' + K_n^{(\prec M)}. \tag{7}$$

It remains to show $t_m'M \in K_n' + K_n^{(\prec M)}$. Let $M = t_m^d N$, where $d \in \mathbb{N}$ and $N \in T_m$. Then $\mathrm{ind}_n(N) \geq m$ and

$$t_m'M = g'N, \tag{8}$$

where $g = t_m^{d+1}/(d+1)$. If $\mathrm{ind}_n(N) = m$, then $n = m$ and $N = 1$. Thus, $t_m'M \in K_n'$ by (8). Otherwise, $g'N \in K_n' + K_n^{(\prec N)}$ by (7), in which f and M are replaced with g and N, respectively. Moreover, $\mathrm{ind}_n(N) > m$ implies $N \prec M$. Thus, $t_m'M \in K_n' + K_n^{(\prec M)}$ by (8). □

In order to obtain sufficient and necessary conditions, we impose an extra condition on the generators:

$$\mathrm{hm}(t_i') = 1 \text{ for all } i \in [n].$$

By Lemma 2.5 and the additive decomposition for rational functions in $C(x)$, for each $i \in [n]$, there exists a simple element h_i in K_{i-1} and an element $g_i \in K_{i-1}$ such that $t_i' = g_i' + h_i$. Let $u_i = t_i - g_i$. Then u_i is a primitive generator over K_{i-1} and $K_{i-1}(t_i) = K_{i-1}(u_i)$. Moreover, $K_0(\bar{t}) = K_0(\bar{u})$. Therefore, without loss of generality, we can further assume that each t_i' is simple for all $i \in [n]$.

Definition 4.3. *A tower $K_0(\bar{t})$ is said to be* S-primitive *if it is a primitive tower and t_i' is simple for all $i \in [n]$.*

Logarithmic towers are S-primitive by Proposition 3.5. Our next goal is to construct remainders in S-primitive towers based on a special property of simple elements.

Lemma 4.4. *Let $K_0(\bar{t})$ be an S-primitive tower. If $f \in K_n'$ is simple, then $f \in \mathrm{span}_C\{t_1', \ldots, t_n'\}$.*

Proof. Since f is simple, $\pi_n(f)$ is t_n-simple, So $\pi_n(f) = \mathrm{hp}_{t_n}(f)$ by the uniqueness of Hermitian parts. Since $f \in K_n'$, we see that $\mathrm{hp}_{t_n}(f) = 0$ by Lemma 2.4 (i). Thus, $\pi_n(f) = 0$, and $f \in K_{n-1}$.

We proceed by induction on n. If $n = 1$, then $f \in K_0 \cap K_1'$ is x-simple by Definition 3.4. By Lemma 2.7, there exists a $c \in C$ such that $f \equiv ct_1' \mod K_0'$. Since both f and t_1' are x-simple, we have that $f = ct_1'$ by Lemma 2.4 (ii) and (iii). Assume that $n > 1$ and the lemma holds for $n-1$. For f in $K_{n-1} \cap K_n'$, there is a $c \in C$ such that $f \equiv ct_n' \mod K_{n-1}'$ by Lemma 2.7. Then $f - ct_n' \in K_{n-1}'$. Since both f and t_n' are simple, $f - ct_n'$ is also simple. By the induction hypothesis, we have that $f - ct_n' \in \mathrm{span}_C\{t_1', \ldots, t_{n-1}'\}$, which implies that $f \in \mathrm{span}_C\{t_1', \ldots, t_n'\}$. □

The previous lemma gives us a direct way to determine whether or not a tower is S-primitive.

Corollary 4.5. *The tower $K_0(\bar{t})$ is S-primitive if and only if $t_1', \ldots, t_n'$ are C-linearly independent and each $t_i' \in K_{i-1}$ is simple.*

Proof. If $K_0(\bar{t})$ is an S-primitive tower, then it is primitive. By Proposition 2.2, $t_i' \notin K_{i-1}'$ for all $i \in [n]$. So $t_1', \ldots, t_n'$ are C-linearly independent. By Definition 4.3, t_i' is simple for all $i \in [n]$.

We prove the converse by induction on n. If $n = 1$, then t_1' is non-zero because it is C-linearly independent, which implies $t_1' \notin K_0'$, because it is x-simple. By Proposition 2.2, t_1 is a primitive generator over K_0. Hence, $K_0(t_1)$ is S-primitive. Suppose that $K_0(t_1, \ldots, t_{n-1})$ is S-primitive. Let us consider the tower $K_0(t_1, \ldots, t_{n-1}, t_n)$. By Lemma 4.4, $t_n' \notin \mathrm{span}_C\{t_1', \ldots, t_{n-1}'\}$ implies that $t_n' \notin K_{n-1}'$. Thus, t_n is a primitive generator over K_{n-1} by Proposition 2.2. The tower under consideration is S-primitive. □

The following lemma gives a sufficient and necessary condition in S-primitive towers for lowering an element with respect to $\prec$ modulo the integrable space K_n'.

Lemma 4.6. *Suppose that $K_0(\bar{t})$ is an S-primitive tower. Let $M \in T_0$ with $\mathrm{ind}_n(M) = m$ and $a \in K_{m-1}$ be simple. Then $aM \in K_n' + K_n^{(\prec M)}$ if and only if $a \in \mathrm{span}_C\{t_1', \ldots, t_m'\}$.*

Proof. The sufficiency follows from Lemma 4.2. Conversely, assume that $aM \in K_n' + K_n^{(\prec M)}$. If $M = 1$, then $m = n$ and $a \in K_n'$. By Lemma 4.4, $a \in \mathrm{span}_C\{t_1', \ldots, t_n'\}$. Otherwise, since $\mathrm{ind}_n(M) = m$, assume that $M = t_m^{d_m} \cdots t_n^{d_n}$ with $d_m > 0$.

We proceed by induction on n. For $n = 1$, $aM \in K_1' + K_1^{(\prec M)}$ implies that there exists a t_1-proper element $b \in K_1$ and $p \in K_0[t_1]$ with $\deg_{t_1}(p) < d_1$ such that $aM + b + p \in K_1'$. We further assume that b is t_1-simple, because $b - \mathrm{hp}_{t_1}(b) \in K_1'$ by Lemma 2.5. So, $b = 0$ by Lemma 2.4 (i). We see that $aM + p \in K_1'$. By Lemma 2.7, $a - ct_1' \in K_0'$ for some $c \in C$. Hence, $a = ct_1'$, because a and t_1' are both x-simple.

Assume that $n > 1$ and that the conclusion holds for $n-1$. Let $N = M/t_n^{d_n}$, which is a power product of $t_m, \ldots, t_{n-1}$. Since aM belongs to $K_n' + K_n^{(\prec M)}$, there is a t_n-proper element b and $p \in K_{n-1}[t_n]$ with $\mathrm{hm}(p) \prec M$ such that $aNt_n^{d_n} + b + p \in K_n'$. Similar to the base case, one can show that $aNt_n^{d_n} + p \in K_n'$. Let

$p = qt_n^{d_n} + r$ such that $q \in K_{n-1}$ with $\mathrm{hm}(q) \prec N$ and $r \in K_{n-1}[t_n]$ with $\deg_{t_n}(r) < d_n$. Then we have $(aN+q)t_n^{d_n} + r \in K'_n$. By Lemma 2.7, there exists $c \in C$ such that $aN + q - ct'_n \in K'_{n-1}$. So,

$$aN \equiv ct'_n \mod \left(K'_{n-1} + K_{n-1}^{(\prec N)}\right). \tag{9}$$

If $N = 1$, then $m = n$ and $a \in K'_n$. So $a \in \mathrm{span}_C\{t'_1, \ldots, t'_n\}$ by Lemma 4.4 and we are done. If $N \succ 1$, then $\mathrm{ind}_{n-1}(N) = m < n$. By (9), $aN \in K'_{n-1} + K_{n-1}^{(\prec N)}$, because $\mathrm{hm}(ct'_n) = 1$. It follows from the induction hypothesis that $a \in \mathrm{span}_C\{t'_1, \ldots, t'_m\}$. □

We can now specify a remainder in S-primitive towers and prove that the algorithm to construct it will terminate.

Proposition 4.7. *Let $K_0(\bar{t})$ be an S-primitive tower, and $r \in K_n$ be nonzero with $m = \mathrm{ind}_n(\mathrm{hm}(r))$. Then r is a remainder if $\pi_n(r)$ is t_n-simple, and $\mathrm{hc}(r - \pi_n(r))$ is simple and is not a nonzero element of $\mathrm{span}_C\{t'_1, \ldots, t'_m\}$.*

Proof. Let $f \in R_r$ as defined in (6). Since $\pi_n(r)$ is t_n-simple, $\mathrm{hp}_{t_n}(f) = \pi_n(r)$ by Lemma 2.4 (ii) and (iii). Then the denominator of r, which is associated to the denominator of $\pi_n(r)$ over K_{n-1}, divides the denominator of f by Theorem 5.3.1 in [5].

We further need to show that $\mathrm{hm}(r) \preceq \mathrm{hm}(f)$. Suppose the contrary. Let $M = \mathrm{hm}(r)$ and $a = \mathrm{hc}(r - \pi_n(r))$.

If $M = 1$, then $m = n$, $a = r - \pi_n(r)$, and $f = 0$, which implies that $r \in K'_n$. Then $\pi_n(r) = 0$ by Lemma 2.4 (i). So, $a \in K_{n-1} \cap K'_n$. By Lemma 4.4, we have that a belongs to $\mathrm{span}_C\{t'_1, \ldots, t'_n\}$. Thus, $a = 0$ and, consequently, $r = 0$, a contradiction.

Assume that $M \succ 1$. Then $\mathrm{hm}(r - f) = M$ and $\mathrm{hc}(r - f) = \mathrm{hc}(r)$ since $M \succ \mathrm{hm}(f)$. Hence, $\mathrm{hc}(r - f) = a$ because $M \succ 1$ and $\mathrm{hm}(\pi_n(r)) \preceq 1$. From $r - f \in K'_n$, we see that $aM \in K'_n + K_n^{(\prec M)}$. By Lemma 4.6, a belongs to $\mathrm{span}_C\{t'_1, \ldots, t'_m\}$, which implies that $a = 0$. Then $r = \pi_n(r)$ and $M = 1$, a contradiction. □

Theorem 4.8. *Let $K_0(\bar{t})$ be an S-primitive tower and let $f \in K_n$. Then one can construct a remainder of f with the properties described in Proposition 4.7 in a finite number of steps.*

Proof. By Lemma 2.5, $\pi_n(f) \equiv \mathrm{hp}_{t_n}(f) \mod K'_n$. Then

$$f \equiv \mathrm{hp}_{t_n}(f) + (f - \pi_n(f)) \mod K'_n. \tag{10}$$

The n-th projection of the right-hand side of the congruence is equal to $\mathrm{hp}_{t_n}(f)$, which is t_n-simple.

Let $M = \mathrm{hm}(f - \pi_n(f))$. We proceed by a Noetherian induction on M with respect to $\prec$. If $M = 0$, then $f = \pi_n(f)$. By (10) and Proposition 4.7, $\mathrm{hp}_{t_n}(f) \in P_n$ is a remainder of f.

Assume that $M \neq 0$, and for any $g \in K_n$ with $\mathrm{hm}(g) \prec M$, there is a remainder $\tilde{r}$ of g as described in Proposition 4.7.

Let $a = \mathrm{hc}(f - \pi_n(f))$ and $m = \mathrm{ind}_n(M)$. Since $a \in K_{m-1}$, its j-th projection is equal to zero for each $j \in \{m, \ldots, n\}$. By Lemma 2.5, $\pi_i(a) \equiv h_i \mod K'_i$ for some t_i-simple elements $h_i \in K_i$ for all $i \in [m-1]_0$ with $t_0 = x$. By Lemma 4.2,

$$f - \pi_n(f) \equiv bM \mod (K'_n + K_n^{(\prec M)}), \tag{11}$$

where $b = \sum_{i=0}^{m-1} h_i$. Note that b is simple by Definition 3.4.

If $b \in \mathrm{span}_C\{t'_1, \ldots, t'_m\}$, then $bM \in K'_n + K_n^{(\prec M)}$ by Lemma 4.2. So $f - \pi_n(f) \equiv g \mod K'_n$ for some g in $K_n^{(\prec M)}$ by (11). Accordingly, g has a remainder $\tilde{r}$ as described in Proposition 4.7 by the induction hypothesis. Thus, $\mathrm{hp}_{t_n}(f) + \tilde{r}$ is a remainder of f.

Assume that $b \notin \mathrm{span}_C\{t'_1, \ldots, t'_m\}$. It follows from (10) and (11) that $f \equiv \mathrm{hp}_{t_n}(f) + bM + g \mod K'_n$ for some g in $K_n^{(\prec M)}$. We may further assume that $\pi_n(g)$ is t_n-simple by Lemma 2.5. The right-hand side of the above congruence is a remainder as described in Proposition 4.7, because b is the head coefficient of $bM + (g - \pi_n(g))$.

Consequently, we construct a remainder of f in a finite number of steps because the ordering $\prec$ is Noetherian. □

We now present an algorithm to decompose an element in an S-primitive tower into a sum of a derivative and a remainder. The algorithm is a slight refinement of the proof of the above theorem. We refer the reader to the online supplementary material [1] for the implementation.

AddDecompInField$(f,\ K_0(\bar{t}))$
Input: An S-primitive tower $K_0(\bar{t})$, described as a list
$$[x, [t_1, \ldots, t_n], [t'_1, \ldots, t'_n]],$$
s.t. $t'_i \in K_{i-1}$ is simple for all $i \in [n]$, and $f \in K_n$.
Output: Two elements $g, r \in K_n$ such that $f = g' + r$ and r satisfies the conditions in Proposition 4.7.

(1) If $f = 0$, then return $(0, 0)$.
(2) Initialize: $M \leftarrow \mathrm{hm}(f)$, $a \leftarrow \mathrm{hc}(f)$, $m \leftarrow \mathrm{ind}_n(M)$, $d \leftarrow \deg_{t_m}(M)$, $B \leftarrow 0$, $H \leftarrow 0$, $\tilde{c} \leftarrow 0$.
(3) Let $a = \sum_{i=0}^{m} a_i$ be the matryoshka decomposition.
(4) Reduction: For each i from 0 to m, compute $b_i, h_i \in K_i$ s.t. $a_i = b'_i + h_i$, where h_i is t_i-simple, and decide whether $\exists\, c_1, \ldots, c_m \in C$ s.t. $h_i = \sum_{j=1}^{m} c_j t'_j$.
  Yes: Update $B \leftarrow B + b_i + \sum_{j=1}^{m-1} c_j t_j$ and $\tilde{c} \leftarrow \tilde{c} + c_m$.
  No: Update $B \leftarrow B + b_i$ and $H \leftarrow H + h_i$.
(5) Lower term: $\ell \leftarrow f - aM - BM' - \frac{\tilde{c}}{d+1} \cdot t_m^{d+1} \cdot \left(M/t_m^d\right)'$.
  Recursion: $\{\tilde{g}, \tilde{r}\} \leftarrow$ AddDecompInField$(\ell,\ K_0(\bar{t}))$.
(6) Return $g = BM + \frac{\tilde{c}}{d+1} \cdot t_m \cdot M + \tilde{g}$ and $r = H \cdot M + \tilde{r}$.

Example 4.9. *Find an additive decomposition for*

$$f = \frac{1}{\log(x)\mathrm{Li}(x)} + \frac{\mathrm{Li}(x) - 2x\log(x)}{(\log(x))^2} + \log(\log(x)),$$

viewed as an element of the S-primitive tower

$$K_3 = C(x)(\underbrace{\log(x)}_{t_1}, \underbrace{\mathrm{Li}(x)}_{t_2}, \underbrace{\log(\log(x))}_{t_3}),$$

and we can write $f = 1/(t_1 t_2) + (t_2 - 2xt_1)/t_1^2 + t_3 \in K_3$. By the above algorithm, we have that

$$f = \left(xt_3 + \frac{t_2^2}{2} - t_2 - \frac{xt_2 + x^2}{t_1}\right)' + \frac{1}{t_1 t_2}. \tag{12}$$

The nonzero remainder $r = 1/(t_1 t_2)$ implies f has no integral in K_3.

An element $f \in K$ is said to have an *elementary integral over K* if there exists an elementary extension E of K and an element g of E such that $f = g'$ (see [5, Definition 5.1.4]). We can use the remainder from Theorem 4.8 to determine whether or not a function has an elementary integral over an S-primitive tower.

[1] https://wongey.github.io/add-decomp-sprimitive/

Theorem 4.10. *Let $K_0(\bar{t})$ be an S-primitive tower and C be algebraically closed. Let $f \in K_n$ have a remainder r as described in Proposition 4.7. Then f has an elementary integral over K_n if and only if $r \in \operatorname{span}_C\{t_1', \ldots, t_n'\} + L_n$, where L_n stands for the C-linear subspace spanned by all logarithmic derivatives in K_n.*

Proof. The sufficiency is obvious. Conversely, there exists an $h \in L_n$ such that $f \equiv h \mod K_n'$ by Liouville's Theorem [5, Theorem 5.5.2]. Then it suffices to show that $r - h \in \operatorname{span}_C\{t_1', \ldots, t_n'\}$.

Since r is a remainder of f, we have that $h \equiv r \mod K_n'$ and $\operatorname{hm}(r) \preceq \operatorname{hm}(h)$. By Proposition 3.5, h is simple, which implies that $\operatorname{hm}(h) \preceq 1$. So $\operatorname{hm}(r) \preceq 1$. If $\operatorname{hm}(r) = 0$, then $r = 0$. Otherwise, $\operatorname{hm}(r) = 1$. Then r is simple by Proposition 4.7. Thus, $r - h$ is simple and integrable in K_n. It is in $\operatorname{span}_C\{t_1', \ldots, t_n'\}$ by Lemma 4.4. □

The proof of Theorem 4.10 gives us an alternate necessary condition, namely $\operatorname{hm}(r) \preceq 1$, to enable a quick check for the elementary integrability of f.

Example 4.11. *Let us reconsider the function f and the tower K_3 in Example 4.9 under the assumption that C is algebraically closed. The remainder is $r = t_2'/t_2$. By Theorem 4.10, f has an elementary integral over K_3. It follows from* (12) *that*

$$\int f\,dx = x\log(\log(x)) + \frac{\operatorname{Li}(x)^2}{2} - \operatorname{Li}(x) - \frac{x\operatorname{Li}(x) + x^2}{\log(x)} + \log(\operatorname{Li}(x)).$$

The Mathematica implementation by Raab based on work in [14] computes the same result. But the "int( )" command in Maple and the "Integrate[ ]" command in Mathematica both leave the integral unevaluated.

As illustrated in Example 4.9, the function f therein has a nonzero remainder in $K_0(t_1, t_2, t_3)$. By Example 4.11, we see that zero is the remainder of f in $K_0(t_1, t_2, t_3)(t_4)$, where $t_4 = \log(t_2)$. However, to determine whether an element belongs to L_n given in Theorem 4.10, one needs the Rothstein-Trager resultant and algebraic numbers over C in general (see [5, Theorem 4.4.3] and [6, §6]), which may be complicated. We seek an easier way to find new generators.

5 LOGARITHMIC TOWERS

Let $K_0(\bar{t})$ and $K_0(\bar{u})$ be two primitive towers over K_0, and ϕ be a differential homomorphism from $K_0(\bar{t})$ to $K_0(\bar{u})$, which means ϕ is a field homomorphism and $\phi(f') = \phi(f)'$ for all $f \in K_0(\bar{t})$. For an element f of $K_0(\bar{t})$ with a remainder r, any remainder of $\phi(f)$ in $K_0(\bar{u})$ is always not higher than $\phi(r)$ with respect to $\prec$, because ϕ embeds the integrable subspace of $K_0(\bar{t})$ into that of $K_0(\bar{u})$.

In practice, determining generators for our towers depends heavily on the given function. In other words, the choice of generators can be done via a clever inspection of the function itself, as the following example shows.

Example 5.1. *Consider the following function in x:*

$$f = \frac{\log((x+1)\log(x))}{x\log(x)}.$$

For this function, there are at least two ways to construct the tower over $\mathbb{Q}(x)$ containing f:

(i) $t_1 = \log(x), t_2 = \log((x+1)\,t_1)$;
(ii) $u_1 = \log(x), u_2 = \log(x+1), u_3 = \log(u_1)$.

The tower $K_0(t_1, t_2)$ can be differentially embedded into $K_0(u_1, u_2, u_3)$ via $t_1 \mapsto u_1$ and $t_2 \mapsto u_2 + u_3$. In the first tower, $f = t_2/(xt_1)$ is already a remainder by Proposition 4.7. In the second tower, AddDecompInField computes a remainder $u_2/(xu_1)$ that is lower than f because $\operatorname{span}_C\{t_1', t_2'\}$ is properly contained in $\operatorname{span}_C\{u_1', u_2', u_3'\}$.

With the aid of the logarithmic derivative identity and the matryoshka decomposition, we are going to show in Theorem 5.6 that, given a logarithmic tower $K_0(t_1, \ldots, t_n)$, one can construct another logarithmic tower $K_0(u_1, \ldots, u_w)$ and a differential homomorphism ϕ such that, for all $i \in [n-1]_0$, $j \in [n]$, the image of $\pi_i(t_j')$ under ϕ belongs to $\operatorname{span}_C\{u_1', \ldots, u_w'\}$, which provides us with more possibilities to reduce a given function by Lemma 4.6. This motivates the following representation of our towers in terms of the generators.

Definition 5.2. *Let $K_0(\bar{t})$ be a primitive tower. The $n \times n$ matrix*

$$A = \left(\pi_i(t_j')\right)_{0 \le i \le n-1, 1 \le j \le n}$$

is called the matrix associated *to $K_0(\bar{t})$.*

$$\begin{array}{c} \\ P_0 \\ P_1 \\ \vdots \\ P_{n-1} \end{array} \begin{array}{c} \begin{array}{cccc} t_1' & t_2' & \cdots & t_n' \\ \downarrow & \downarrow & & \downarrow \end{array} \\ \begin{pmatrix} \star & \star & \cdots & \star \\ & \star & \cdots & \star \\ & & \ddots & \vdots \\ & & & \star \end{pmatrix} \end{array}$$

Figure 3: A labeled associated matrix of a primitive tower. The $\star$ represents a possibly nonzero element.

The associated matrix records all information about the derivation on $K_0(\bar{t})$, because $\pi_n(t_1') = \cdots = \pi_n(t_n') = 0$. Since $t_j' \in K_{j-1}$ for all $j \in [n]$, the associated matrix A is in upper triangular form as in Figure 3. Furthermore, if $K_0(\bar{t})$ is a logarithmic tower, then the entries of A are all C-linear combinations of logarithmic derivatives by Lemma 2.6 (ii).

In the following discussion, a tower with a different set of generators $\bar{v} = (v_1, \ldots, v_n)$ will appear. We say that $K_0(\bar{t})$ *is equal to* $K_0(\bar{v})$ if they are equal as a field, and that $K_0(\bar{t})$ is equal to $K_0(\bar{v})$ *as a tower* if $K_0(t_1, \ldots, t_i) = K_0(v_1, \ldots, v_i)$ for all $i \in [n]$. We will invoke the superscript notation to distinguish between different sets of generators (for example, $\pi_i^{\bar{t}}$ for projections in $K_0(\bar{t})$).

Definition 5.3. *Let $K_0(\bar{t})$ be a primitive tower and $f \in K_n \setminus \{0\}$. The* significant index *of f is*

$$\operatorname{si}^{\bar{t}}(f) := \max\{i \in [n]_0 \mid \pi_i(f) \neq 0\}.$$

The vector

$$\operatorname{sv}(\bar{t}) := \left(\operatorname{si}^{\bar{t}}(t_1'), \ldots, \operatorname{si}^{\bar{t}}(t_n')\right)$$

is called the significant vector *of $K_0(\bar{t})$. Suppose $\operatorname{sv}(\bar{t})$ is equal to $(k_1, \ldots, k_n)$. The sequence*

$$\operatorname{sc}(\bar{t}) := \left(\pi_{k_1}^{\bar{t}}(t_1'), \ldots, \pi_{k_n}^{\bar{t}}(t_n')\right)$$

is called the significant component sequence *of $K_0(\bar{t})$.*

The significant vector and significant component sequence are unique with respect to the generators by the matryoshka decomposition.

Example 5.4. *Consider the field*

$$C(x)\,(\log(x), \log(\log(x)), \log((x+1)\log(x)))\,.$$

We set $t_1 = \log(x), t_2 = \log(t_1)$, *and* $t_3 = \log((x+1)\,t_1)$. *Then* $C(x)(t_1, t_2, t_3)$ *is a logarithmic tower whose significant vector is equal to* $(0, 1, 1)$ *and whose significant component sequence is*

$$(1/x, 1/(xt_1), 1/(xt_1)).$$

Definition 5.5. *A logarithmic tower* $K_0(\bar{t})$ *is said to be* well-generated *if*

(CLI) $\mathrm{sc}(\bar{t})$ *is* C*-linearly independent,*
(MI) $\mathrm{sv}(\bar{t})$ *is (weakly) monotonically increasing, and*
(ONE) *each column of its associated matrix contains exactly one non-zero element.*

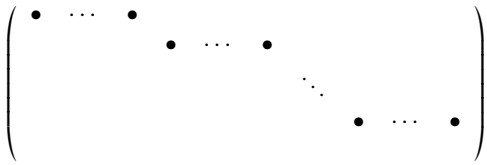


Figure 4: The associated matrix of a well-generated tower is in the form of a "staircase" where the •'s are C-linearly independent and other entries are zero.

We will show that any logarithmic tower $K_0(\bar{t})$ can be embedded into a well-generated one. To this end, we impose the usual lexicographical order on two significant vectors [10, Ch. 2, Def. 3].

Theorem 5.6. *Let* $K_0(\bar{t})$ *be a logarithmic tower. Then there exists a well-generated logarithmic tower* $K_0(\bar{u})$, *where* $\bar{u} = (u_1, \ldots, u_w)$ *and* $n \le w \le n(n+1)/2$, *and a differential monomorphism* ϕ *from* $K_0(\bar{t})$ *into* $K_0(\bar{u})$ *with* $\phi|_{K_0} = \mathrm{id}_{K_0}$.

Proof. This proof will be separated into two parts. The first part will show that each primitive (specifically, logarithmic) tower is equal (as a field) to one where properties (CLI) and (MI) are satisfied. This will enable us to embed the resulting logarithmic tower into a well-generated one, which makes up the second part of the proof.

If $K_0(\bar{t})$ does not satisfy (CLI) and (MI), then $\exists\, v_1, \ldots, v_n \in K_0(\bar{t})$ such that $K_0(\bar{v})$ is primitive and equals to $K_0(\bar{t})$, and $\mathrm{sv}(\bar{v})$ is lower than $\mathrm{sv}(\bar{t})$. Since the order of the significant vectors is Noetherian, we can eventually reach a primitive tower that satisfies both (CLI) and (MI).

We start by supposing that $\mathrm{sc}(\bar{t})$ is C-linearly dependent. Since all components of $\mathrm{sc}(\bar{t})$ are different from 0 by definition, there exists an $i \in \{2, \ldots, n\}$ and constants $c_1, \ldots, c_{i-1}$ such that

$$\mathrm{sc}_i = \sum_{j=1}^{i-1} c_j \mathrm{sc}_j,$$

where sc_j is the j-th element in $\mathrm{sc}(\bar{t})$. Moreover, $\mathrm{si}^{\bar{t}}(c_j t_j') = \mathrm{si}^{\bar{t}}(t_i')$ for all j with nonzero c_j. We remove the last non-zero projection of t_i' by setting $v_k := t_k$ for all $k \in [n] \setminus \{i\}$ and $v_i := t_i - \sum_{j=1}^{i-1} c_j t_j$. Thus, $K_0(\bar{v}) = K_0(\bar{t})$. Also, $\mathrm{si}^{\bar{v}}(v_k') = \mathrm{si}^{\bar{t}}(t_k')$ for all k in $[n] \setminus \{i\}$ and $\mathrm{si}^{\bar{v}}(v_i') < \mathrm{si}^{\bar{t}}(t_i')$. We conclude that $K_0(\bar{v})$ is a primitive tower with a lower significant vector than $K_0(\bar{t})$.

Next, we assume that $\mathrm{sv}(\bar{t})$ is not monotonically increasing. Then there exists an $i \in [n]$ such that $\mathrm{si}^{\bar{t}}(t_1') \le \cdots \le \mathrm{si}^{\bar{t}}(t_i')$ and $\mathrm{si}^{\bar{t}}(t_{i+1}') < \mathrm{si}^{\bar{t}}(t_i')$. We switch the i-th and $(i+1)$-st generators by setting $v_k := t_k$ for all $k \in [n] \setminus \{i, i+1\}$ and

$$v_i := t_{i+1};\ v_{i+1} := t_i.$$

Thus, $K_0(\bar{v})$ is equal to $K_0(\bar{t})$. Also, $\mathrm{si}^{\bar{v}}(v_j') = \mathrm{si}^{\bar{t}}(t_j')$ for $j \in [i-1]$ and $\mathrm{si}^{\bar{v}}(v_i') < \mathrm{si}^{\bar{t}}(t_i')$. Thus, $K_0(\bar{v})$ is a primitive tower with a lower significant vector than $K_0(\bar{t})$.

If the original primitive tower from the argument is logarithmic, then the new generators from the above process are also logarithmic generators. This implies the new tower must be logarithmic satisfying (CLI) and (MI), and this is what we assume about $K_0(\bar{t})$ from this point forward.

For the second part of the proof, we show that $K_0(\bar{t})$ can be embedded into a well-generated tower. We find the C-basis of the associated matrix $\left(\pi_i(t_j')\right)$ by letting $b_1 = \pi_0(t_1')$ and identifying all C-linearly independent elements $b_2, \ldots, b_w$, ordered by searching the matrix from left to right and top to bottom. Since $K_0(\bar{t})$ is primitive, $n \le w \le n(n+1)/2$. Since $K_0(\bar{t})$ satisfies (CLI) and (MI), there exist $\ell_1, \ldots, \ell_n \in [w]$ such that $\ell_1 = 1, \ell_n = w$,

$$\ell_1 < \ell_2 < \cdots < \ell_n \text{ and } \left(b_{\ell_1}, \ldots, b_{\ell_n}\right) = \mathrm{sc}(\bar{t}). \tag{13}$$

By the definition of the associated matrix and the ordering of $\{b_1, \ldots, b_w\}$, for all $j \in [n]$ there exist $c_{j,k} \in C$ such that

$$t_j' = b_{\ell_j} + \sum_{k=1}^{\ell_j - 1} c_{j,k} b_k. \tag{14}$$

Let $u_1, \ldots, u_w$ be algebraically independent indeterminates over K_0, and $\bar{u} := (u_1, \ldots, u_w)$. Let $v_j := u_{\ell_j} + \sum_{k=1}^{\ell_j - 1} c_{j,k} u_k$ for all $j \in [n]$. Then $v_1, \ldots, v_n$ are algebraically independent over K_0, because u_{ℓ_j} does not appear in the expressions defining $v_1, \ldots, v_{j-1}$. It follows that $\phi : K_0(\bar{t}) \to K_0(\bar{u})$ defined by $f(t_1, \ldots, t_n) \mapsto f(v_1, \ldots, v_n)$ is a monomorphism and $\phi|_{K_0} = \mathrm{id}_{K_0}$. For every $k \in [w]$, we define

$$u_k' = \phi(b_k). \tag{15}$$

Since $u_1, \ldots, u_w$ are algebraically independent over K_0, by Corollary 1′ in [17, page 124], the field $K_0(u_1, \ldots, u_w)$ can be uniquely turned into a differential field such that its derivation agrees with the one on K_0 and also satisfies (15). By (14), $\phi(t_j') = v_j'$ for all $j \in [n]$. Thus, ϕ is a differential monomorphism.

Lastly, we show that $K_0(\bar{u})$ is a well-generated tower over K_0. Set $\ell_0 = 0$. For each $k \in [w]$, there exists a $j \in [n]$ such that $\ell_{j-1} < k \le \ell_j$. Then $s := \mathrm{si}^{\bar{t}}(b_k) \le \mathrm{si}^{\bar{t}}(t_j') < j$ and b_k is t_s-proper. Since ϕ is a monomorphism, it preserves degrees. By (15), u_k' is u_{ℓ_s}-proper, where $\ell_s \le \ell_{j-1} < k$ since $s < j$. Hence, $u_k' \in K_0(u_1, \ldots, u_{k-1})$. Since ϕ is differential and b_k is a C-linear combination of logarithmic derivatives, so is u_k' by (15). In particular, u_k' is u_{ℓ_s}-simple by Lemma 2.6 (i). Moreover, $b_1, \ldots, b_w$ are C-linearly independent, and so are $\phi(b_1), \ldots, \phi(b_w)$ because ϕ is a monomorphism. It follows from (15) that $u_1', \ldots, u_w'$ are C-linearly independent, which

implies that $K_0(\bar{u})$ is a logarithmic tower by Corollary 4.5. In addition, $\pi_i(u_k') = 0$ for all $k \in [w]$ and $i \in [w] \setminus \{\ell_s\}$, because u_k' is u_{ℓ_s}-proper. Consequently, $K_0(\bar{u})$ is well-generated. □

The next example illustrates the results of the embedding algorithm and AddDecompInField in both towers.

Example 5.7. *Consider the logarithmic tower*

$$\mathcal{F} = C(x)\Big(\underbrace{\log(x)}_{t_1}, \underbrace{\log(xt_1)}_{t_2}, \underbrace{\log\big((x+1)(t_1+1)\log(xt_1)\big)}_{t_3}\Big).$$

By Theorem 5.6, there exists a well-generated tower

$$\mathcal{E} = C(x)\Big(\underbrace{\log(x)}_{u_1}, \underbrace{\log(x+1)}_{u_2}, \underbrace{\log(u_1)}_{u_3}, \underbrace{\log(u_1+1)}_{u_4}, \underbrace{\log(u_1+u_3)}_{u_5}\Big)$$

and a differential homomorphism ϕ from $\mathcal{F}$ to $\mathcal{E}$ given by $\phi(t_1) = u_1$, $\phi(t_2) = u_1 + u_3$ and $\phi(t_3) = u_2 + u_4 + u_5$. The associated matrices of $\mathcal{F}$ and $\mathcal{E}$ are, respectively,

$$\begin{pmatrix} \frac{1}{x} & \frac{1}{x} & \frac{1}{x+1} \\ 0 & \frac{t_1'}{t_1} & \frac{t_1'}{t_1+1} \\ 0 & 0 & \frac{1+t_1}{xt_1t_2} \end{pmatrix} \quad \text{and} \quad \begin{pmatrix} \frac{1}{x} & \frac{1}{x+1} & 0 & 0 & 0 \\ 0 & 0 & \frac{u_1'}{u_1} & \frac{u_1'}{u_1+1} & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & \frac{(u_1+u_3)'}{u_1+u_3} \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix}.$$

Let

$$f_1 = \frac{(t_1+1)^2 + t_1t_2}{xt_1(t_1+1)t_2} \quad \text{and} \quad f_2 = \frac{t_3}{x}$$

be two elements of $\mathcal{F}$. Then $\phi(f_1)$ and $\phi(f_2)$ are

$$\frac{(u_1+1)^2 + u_1(u_1+u_3)}{xu_1(u_1+1)(u_1+u_3)} \quad \text{and} \quad \frac{u_2+u_4+u_5}{x},$$

respectively. Using AddDecompInField, we compute the respective remainders of f_1 and f_2 to obtain

$$r_1 = f_1 \quad \text{and} \quad r_2 = \frac{t_1}{-(x+1)} + \frac{1}{x(t_1+1)} + \frac{-(t_1+1)}{xt_2}.$$

In the same vein, we get the remainders of $\phi(f_1)$ and $\phi(f_2)$,

$$\tilde{r}_1 = 0 \quad \text{and} \quad \tilde{r}_2 = \frac{u_1}{-(x+1)} + \frac{-(u_1+1)}{x(u_1+u_3)},$$

respectively. Note that $\phi(r_1) \neq 0$ but $\tilde{r}_1 = 0$, which implies that $\tilde{r}_1 \prec \phi(r_1)$. While $\mathrm{hm}(\tilde{r}_2) = \mathrm{hm}(\phi(r_2))$, we observe that $\tilde{r}_2$ has fewer nonzero projections than $\phi(r_2)$.

6 CONCLUSIONS

In this article, we have developed an additive decomposition in S-primitive towers. The decomposition algorithm is based on the matryoshka decomposition of functions, Hermite reduction and integration by parts. It provides an alternative method to Risch's algorithm for determining in-field (resp. elementary) integrability in (resp. over) an S-primitive tower. Moreover, we embed a logarithmic tower into a well-generated one, where functions can be decomposed further.

We observe that the notion of remainders is defined according to a partial order among multivariate rational functions. It would be possible to refine this notion so that all remainders of a given function share more common properties. Moreover, we plan to investigate whether our additive decomposition is applicable to compute telescopers for elements in an S-primitive tower, as carried out in [6]. We also hope to develop an additive decomposition in exponential extensions.

ACKNOWLEDGMENTS

We are grateful to Shaoshi Chen, Christoph Koutschan and Clemens Raab for their valuable comments and suggestions. The authors would also like to thank the anonymous reviewers for their remarks, which helped us to greatly improve the manuscript. H. Du and E. Wong were supported by the Austrian Science Fund (FWF): F5011-N15. J. Guo and Z. Li were supported by two NSFC Grants 11871067 and 11771433.

REFERENCES

[1] S.A. Abramov. Indefinite sums of rational functions. *Proceedings of the 1995 International Symposium on Symbolic and Algebraic Computation.* New York, NY, USA: ACM, 1995: 303-308.
[2] A. Bostan, S. Chen, F. Chyzak and Z. Li. Complexity of creative telescoping for bivariate rational functions. *Proceedings of the 2010 International Symposium on Symbolic and Algebraic Computation.* New York, NY, USA: ACM, 2010: 203-210.
[3] A. Bostan, S. Chen, F. Chyzak, Z. Li and G. Xin. Hermite reduction and creative telescoping for hyperexponential functions. *Proceedings of the 2013 International Symposium on Symbolic and Algebraic Computation.* New York, NY, USA: ACM, 2013: 77-84.
[4] A. Bostan, F. Chyzak, P. Lairez and B. Salvy. Generalized Hermite reduction, creative telescoping and definite integration of D-finite functions. *Proceedings of the 2018 International Symposium on Symbolic and Algebraic Computation.* New York, NY, USA: ACM, 2018: 95-102.
[5] M. Bronstein. *Symbolic Integration I*: transcendental functions. Berlin: Springer-Verlag, 2005.
[6] S. Chen, H. Du and Z. Li. Additive decompositions in primitive extensions. *Proceedings of the 2018 International Symposium on Symbolic and Algebraic Computation.* New York, USA: ACM, 135-142.
[7] S. Chen, M. van Hoeij, M. Kauers and C. Koutschan. Reduction-based creative telescoping for Fuchsian D-finite functions. *Journal of Symbolic Computation*, 2018, 85:108 - 127.
[8] S. Chen, H. Huang, M. Kauers and Z. Li. A modified Abramov-Petkovšek reduction and creative telescoping for hypergeometric terms. *Proceedings of the 2015 International Symposium on Symbolic and Algebraic Computation.* New York, NY, USA: ACM, 2015: 117-124.
[9] S. Chen, M. Kauers and C. Koutschan. Reduction-based creative telescoping for algebraic functions. *Proceedings of the 2016 International Symposium on Symbolic and Algebraic Computation.* New York, NY, USA: ACM, 2016: 175-182.
[10] D. Cox, J. Little, D. O'Shea. *Ideals, Varieties and Algorithms.* Fourth Edition, Springer, 2015.
[11] H. Du, H. Huang and Z. Li. A q-analogue of the modified Abramov-Petkovšek reduction. *Advances in Computer Algebra.* S. Schneider and C. Zima (eds.) Springer International Publishing, 2018: 105-129.
[12] C. Hermite. Sur l'intégration des fractions rationnelles. *Ann. Sci. École Norm. Sup.(2)*, 1872(1): 215-218.
[13] M. V. Ostrogradsky. De l'intégration des fractions rationnelles. *Bull. de la classe physico-mathématique de l'Acad. Impériale des Sciences de Saint-Pétersbourg*, 1845, 4: 145-167, 286-300.
[14] C. Raab. *Definite Integration in Differential Fields.* PhD thesis, RISC, Johannes Kepler University, Linz, Austria, 2012.
[15] M. Singer, S. David and B. Caviness. An extension of Liouville's theorem on integration in finite terms. *SIAM J. Comput.* 1985, 14: 966-990
[16] J. van der Hoeven. Constructing reductions for creative telescoping. *Applicable Algebra in Engineering, Communication and Computing.* 2020 https://doi.org/10.1007/s00200-020-00413-3.
[17] O. Zariski and P. Samuel. *Commutative Algebra I.* Graduate Texts in Mathematics, Springer, 1975.

Numerical Equality Tests for Rational Maps and Signatures of Curves

Timothy Duff
tduff3@gatech.edu
School of Mathematics, Georgia Tech
Atlanta, Georgia, USA

Michael Ruddy
michael.ruddy@mis.mpg.de
Max Planck Institute for Mathematics in the Sciences
Leipzig, Germany

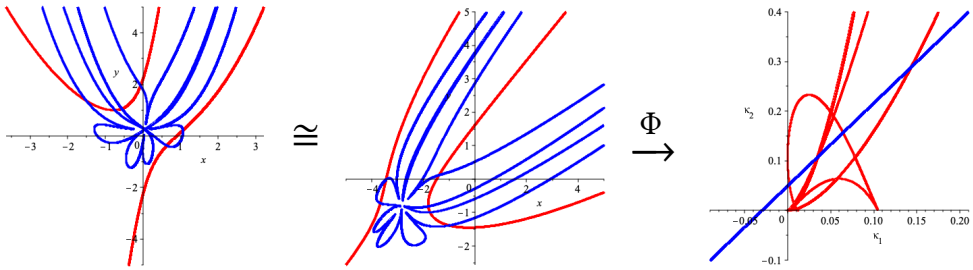


Figure 1: Two curves and their signature in red. A line and its pullback in blue.

ABSTRACT

We apply numerical algebraic geometry to the invariant-theoretic problem of detecting symmetries between two plane algebraic curves. We describe an efficient equality test which determines, with "probability-one", whether or not two rational maps have the same image up to Zariski closure. The application to invariant theory is based on the construction of suitable signature maps associated to a group acting linearly on the respective curves. We consider two versions of this construction: differential and joint signature maps. In our examples and computational experiments, we focus on the complex Euclidean group, and introduce an algebraic joint signature that we prove determines equivalence of curves under this action. We demonstrate that the test is efficient and use it to empirically compare the sensitivity of differential and joint signatures to noise.

KEYWORDS

differential invariants, invariant theory, numerical algebraic geometry, polynomial systems, Euclidean group, computer algebra, homotopy continuation

ACM Reference Format:
Timothy Duff and Michael Ruddy. 2020. Numerical Equality Tests for Rational Maps and Signatures of Curves. In *International Symposium on Symbolic and Algebraic Computation (ISSAC '20), July 20–23, 2020, Kalamata, Greece.* ACM, New York, NY, USA, 8 pages. https://doi.org/10.1145/3373207.3404050

ISSAC '20, July 20–23, 2020, Kalamata, Greece

ACM ISBN 978-1-4503-7100-1/20/07…$15.00
https://doi.org/10.1145/3373207.3404050

1 INTRODUCTION

This paper studies two related problems.

Problem 1. Given two irreducible algebraic varieties, $X_0 \subset \mathbb{C}^{n_0}$ and $X_1 \subset \mathbb{C}^{n_1}$, and two rational maps, $\Phi_0 : X_0 \dashrightarrow \mathbb{C}^m$ and $\Phi_1 : X_1 \dashrightarrow \mathbb{C}^m$, decide if $\overline{\operatorname{im}\Phi_0} = \overline{\operatorname{im}\Phi_1}$.

Problem 2. Given a positive dimensional algebraic group $G \subset \mathcal{PGL}_3(\mathbb{C})$ acting linearly on $\mathbb{C}^2$ and two plane algebraic curves $C_0, C_1 \subset \mathbb{C}^2$, decide if there exists $g \in G$ such that $C_0 = \overline{g \cdot C_1}$.

In the context of *differential invariant theory*, we can reduce Problem 2 to Problem 1 by constructing a suitable *signature map* for the action of G on the curves C_1, C_2. For Problem 1, the field of *numerical algebraic geometry* furnishes a suite of "probability-one" tests. In this article, we explain the aforementioned approaches to these problems in detail and demonstrate that they yield practical *equality tests* for both problems.

In Problem 1, $\overline{\operatorname{im}\Phi_i}$ denotes the Zariski closure of the image of Φ_i. We do not address the more delicate problem of deciding equality of the constructible sets $\operatorname{im}\Phi_i$.

A formally correct algorithmic solution to Problem 1 clearly depends on how the input is "given" to us and what type of guarantee we seek. A natural route via symbolic computation is to compute the ideal of implicit equations for each map and check if these ideals are equal. This is a standard application of Gröbner bases; resultants and more specialized techniques may provide useful alternatives.

Our approach to Problem 1 via numerical algebraic geometry is in the same spirit as previous works [8, 17, 18], where the cost of implicitization is replaced by the cost of computing certain *witness sets*. A key feature of our approach is that it requires a pre-computed witness set for only one of the maps, say Φ_1. This feature is motivated mainly by our interest in Problem 2. We view computing a witness set for Φ_1 as an *offline* cost. The *online* cost of testing equality via Algorithm 1 is typically negligible by comparison. This

is advantageous in a scenario where we wish to test Φ_1 against many different choices of Φ_0.

To reduce Problem 2 to Problem 1, one may use the maps obtained by restricting a pair of independent, rational differential invariants for G to C_0 and C_1 [22], which can be explicitly constructed via the Fels-Olver moving frame method [11] or its algebraic formulation [20]. The image of an algebraic curve C under this map is the curve's *differential signature*. In greater generality, differential signatures may be constructed for smooth submanifolds of some ambient space equipped with a Lie group action. The differential signature locally characterizes the manifold's equivalence class under the action, meaning that manifolds with the same signature are locally equivalent under the Lie group [11]. For an algebraic group acting on $\mathbb{C}^2$ and a plane curve $C \subset \mathbb{C}^2$, such a construction yields a rational map $\Phi : \mathbb{C}^2 \dashrightarrow \mathbb{C}^2$. In this special case, local equivalence implies global equivalence.

Example 1.1. In Figure 1, the red curve on left depicts real points (x, y) such that $8x^3-20xy+2y^2+5x-10 = 0$. Applying a real rotation and translation yields the curve in the middle. Thus these curves are equivalent under the linear action of the complex Euclidean group $\mathcal{E}_2(\mathbb{C})$. The closed image of their respective differential signature maps is the red curve of degree 48 depicted on the right.

Differential signatures of curves have been successfully applied to object recognition under noise, with applications ranging from jigsaw puzzle reconstruction [19] to medical imaging [13]. Differential signatures have also been used to solve classical invariant theory problems such as determining equivalence of binary and ternary forms [4, 21, 29]. The setting of algebraic curves is a useful testing ground for algorithms in this subject. In [7] the notion of a signature polynomial was introduced to determine equivalence of plane algebraic curves via implicitization methods. In [22] it is shown that this reduction to implicitization can always be done for any group acting as in Problem 2.

In this paper we show that the numerical algorithm for Problem 1 yields an effective way for solving Problem 2 using differential signatures, even when implicitization is not practically feasible. We also consider *joint signatures*, which are obtained by constructing rational maps using joint invariants of the induced action of G on the product $\mathbb{C}^2 \times \ldots \times \mathbb{C}^2$ [30]. While we focus on plane curves, in principle the numerical equality test can be used to determine equivalence of higher dimensional varieties through differential or joint signatures, provided one can find a suitable set of *rational* differential or joint invariants.

In Section 2, we review notions from numerical algebraic geometry and describe a general solution to Problem 1 (Algorithm 1.) Section 3 considers the signature approach to Problem 2. In 3.1 we follow the construction in [7, 22] to describe a differential signature for plane algebraic curves using a *classifying pair* of differential invariants. In 3.2 we describe how joint signatures can be used to determine equivalence of plane curves using lower order differential invariant functions, with a detailed analysis in the case of the complex Euclidean group $\mathcal{E}_2(\mathbb{C})$. In Section 4, we describe an implementation in Macaulay2 [12], which has been successful for studying both classes of maps on curves of degree up to 10. Our (reproducible) experiments show that offline witness computation for plane curves of various degrees is feasible, that the online equality test gives a fast alternative to symbolic methods, and that the numerical approach is robust in a certain regime of noise.[1]

2 NUMERICAL EQUALITY

2.1 Background

In this subsection we fix notation and terminology related to algebraic varieties and witness sets. A more comprehensive overview of numerical algebraic geometry may be found in the survey [32] or books [3, 33]. A general system of polynomial equations is denoted by a c-tuple $f = (f_1, \ldots, f_c)$ for $f_1, \ldots, f_c \in \mathbb{C}[x_1, \ldots, x_n]$. Where convenient, we may identify f with a map $\mathbb{C}^n \to \mathbb{C}^c$. The vanishing locus $V(f) := \{x \in \mathbb{C}^n \mid f_1(x) = \cdots = f_c(x) = 0\}$ is a closed subvariety of $\mathbb{C}^n$. If c is the codimension of $V(f)$, then f is said to be a *regular sequence* and the variety $V(f)$ is a *complete intersection*. For polynomial systems $f = (f_1, \ldots, f_k)$ and $g = (g_1, \ldots, g_{k'})$ we write $(f \mid g) := (f_1, \ldots, f_k, g_1, \ldots, g_{k'})$, yielding a polynomial system whose vanishing locus is $V(f) \cap V(g)$. A property is said to hold generically on an irreducible variety X if it holds on some nonempty Zariski-open $U \subset X$.We say that f is *generically reduced* along X if there exists a point $x \in X$ such that the tangent space $T_x(f) = \ker\left(df_i/dx_j\right)$ has dimension $n - c$.

The main data structures in numerical algebraic geometry are variations on the notion of a *witness set*. The overarching idea is to represent an irreducible variety $X \subset \mathbb{C}^n$ by its intersection with a generic affine linear subspace of complementary dimension. The number of points in such an intersection is the degree, $\deg X$.

We define a c-slice in $\mathbb{C}^n$ to be a polynomial system consisting of c affine hyperplanes, $L = (l_1, \ldots, l_c)$ with $l_i \in \mathbb{C}[x_1, \ldots, x_n]_{\leq 1}$. For convenience we write L in place of $V(L(x))$ and also use the notation L^c. For X an irreducible variety of codimension c and a generic slice L^c, the intersection $X \cap L^c$ is *transverse*, consisting of $\deg X$ isolated, nonsingular points.

The standard definition of a witness set for a variety assumes that defining equations for the variety of interest are known. A more flexible notion is that of a *pseudo-witness set* for a rational map. This was first studied for linear projections in [17]. Our Definition 2.1 differs from that used in [3, 17, 18]; to distinguish our setup, we provisionally use the term *weak pseudowitness set*.

Definition 2.1. Let $V(f) \subset \mathbb{C}^n$ be Zariski-closed, $X \subset V(f)$ be one of its irreducible components, and $\Phi : X \dashrightarrow \mathbb{C}^m$ be a rational map. Set $c = \operatorname{codim} V(f)$, $d = \dim \overline{\operatorname{im}}\, \Phi$. A weak pseudowitness set for Φ is a quadruple $(f, \Phi, (L|L'), \{w_1, \ldots, w_e\})$, where L is a generic affine $(m-d)$-slice of $\overline{\operatorname{im}}\, \Phi$, L' is a generic affine $(c-m+d)$-slice of X, and such that $w_1, \ldots, w_e$ are points in $X \cap L'$ where Φ is defined such that $\overline{\operatorname{im}}\, \Phi \cap L = \{\Phi(w_1), \ldots, \Phi(w_e)\}$ and $e = \deg \overline{\operatorname{im}}\, \Phi$.

The data in Definition 2.1 are already sufficient for testing queries of the form $y \in \overline{\operatorname{im}}\, \Phi$, as noted in [17, Remark 2]. For testing, $y \in \operatorname{im} \Phi$ and other applications, the stronger notion is required [18]. Further applications of pseudowitness sets are in [6, 8].

In our context, equations defining $\overline{\operatorname{im}}\, \Phi$ are seldom known, so in what follows we may informally refer to the objects of Definition 2.1 and their multiprojective counterparts in Definition 2.2 as "witness sets" without ambiguity.

[1] Obtain the code at https://github.com/timduff35/NumericalSignatures.

Following [15, 16, 25], we give a multiprojective generalization of Definition 2.1. For irreducible $X \subset \mathbb{C}^n$, we fix $(n_1, \ldots, n_k)$, an integer partition of n, and consider X in the affine space $\mathbb{C}^{n_1} \times \cdots \times \mathbb{C}^{n_k}$. We consider slices $L^{\boldsymbol{e}} = L^{e_1} | \cdots | L^{e_k}$, where $\boldsymbol{e} = (e_1, \ldots, e_k) \in \mathbb{N}^k$ is an integral vector such that $e_1 + \cdots e_k = \dim X$, and L^{e_j} is a e_j-slice consisting of e_j affine hyperplanes in the coordinates of $\mathbb{C}^{n_j}$. We say that $\boldsymbol{e}$ is a *multidimension* of X if for generic $L^{\boldsymbol{e}}$ the intersection $X \cap L^{\boldsymbol{e}}$ is a finite set of nonsingular points; the number of points for such $L^{\boldsymbol{e}}$ is a constant called the $\boldsymbol{e}$*-multidegree* $\deg_{\boldsymbol{e}} X$.

Definition 2.2. Let f, X, c, L', Φ be as in 2.1, and $\boldsymbol{e}$ be a multidimension of $\overline{\operatorname{im} \Phi}$ corresponding to some partition of n. An $\boldsymbol{e}$-weak pseudowitness set for Φ consists of $\left(f, \Phi, (L^{\boldsymbol{e}} | L'), \{w_1, \ldots, w_e\}\right)$, such that $\overline{\operatorname{im} \Phi} \cap L^{\boldsymbol{e}} = \{\Phi(w_1), \ldots, \Phi(w_e)\}$ and $e = \deg_{\boldsymbol{e}} \overline{\operatorname{im} \Phi}$.

The general membership test for multiprojective varieties proposed in [16] uses the stronger notion of a witness collection. This is required since for an arbitrary point $x \in X$ there may not exist transverse slices $L^{\boldsymbol{e}} \ni x$ for $\boldsymbol{e}$ ranging over all multidimensions of X—see [16, Example 3.1]. This subtlety is not encountered for generic $x \in X$; we record this basic fact in Proposition 2.3.

Proposition 2.3. *Fix irreducible $X \subset \mathbb{C}^{n_1} \times \cdots \times \mathbb{C}^{n_k}$ and $\boldsymbol{e}$ some multi-dimension of X. For $x = (x_1, \cdots, x_k) \in X$ generic, there exists an $\boldsymbol{e}$-slice $L^{\boldsymbol{e}} \ni x$ such that $\dim(X \cap L^{\boldsymbol{e}}) = 0$. Moreover, for $x \notin X_{sing}$, we also have that $x \notin (X \cap L^{\boldsymbol{e}})_{sing}$ for generic $L^{\boldsymbol{e}}$.*

Proof. For generic x_1 in the image of $\pi_1 : X \to \mathbb{C}^{n_1}$ we have that the fiber $\pi_1^{-1}(x_1)$ has dimension $\dim X - \dim \pi_1(X)$. Choose such an x_1 and let $L^{e_1} \ni x_1$ be generic so that $\pi_1(X) \cap L^{e_1}$ has dimension $\dim \pi_1(X) - e_1$. It follows that $(X \cap L^{e_1}$ has dimension $\dim X - e_1$. This construction holds for all x_1 on some Zariski open $U_1 \subset \pi_1(X)$. Repeating this construction for the remaining factors yields $U_2, \ldots, U_k$ such that the first part holds for all $x \in U_1 \times \cdots \times U_k$. The second part follows from the appropriate Bertini theorem, cf. [14, Thm 17.16]. □

2.2 A general equality test

Now let $\Phi_0 : X_0 \dashrightarrow \mathbb{C}^m$ and $\Phi_1 : X_1 \dashrightarrow \mathbb{C}^m$ denote two rational maps with each $X_i \subset \mathbb{C}^{n_i}$ of codimension c_i. Problem 1 from the introduction asks us to decide whether or not their images are equal up to Zariski closure. A probabilistic procedure is given in Algorithm 1. This equality test refines general membership and equality tests from numerical algebraic geometry, which are summarized in [33, Ch. 13, 15] and [3, Ch. 8,16]. Our setup is motivated by an efficient solution to Problem 2. Following the standard terminology, our test correctly decides equality with "probability-one" in an idealized model of computation. This is the content of Theorem 2.4. Standard disclaimers apply, since any implementation must rely on numerical approximations in floating-point [3, Ch. 3, pp. 43-45].

Algorithm 1 assumes different representations for the two maps. The map Φ_1 is represented by a witness set in the sense of Definition 2.1, say $(f_1, \Phi_1, (L_1 | L_1'), \{w_1, \ldots, w_e\})$. In fact, the only data needed by Algorithm 1 are the map itself Φ_1, the slice L_1, and the points $w_1, \ldots, w_e$. For the map Φ_0, we need only a sampling oracle that produces generic points on X_0 and $\operatorname{codim}(X_0)$-many reduced equations vanishing on X_0.

Suppose $\dim \overline{\operatorname{im} \Phi_0} = \dim \overline{\operatorname{im} \Phi_1} = d$. There is a probabilistic membership test for queries of the form $\Phi_0(x_0) \in \overline{\operatorname{im} \Phi_1}$ based on homotopy continuation. The relevant homotopy depends parametrically on L_1, a $(m-d)$-slice $L_0 \ni \Phi_0(x_0)$, a $(c_0 - m + d)$-slice $L_0' \ni x_0$, and a regular sequence $f_0 = (f_{0,1}, \ldots, f_{0,c_0})$ which is generically reduced with respect to X_0. The homotopy H is defined as

$$H(x; t) = \left(f_0 \,\middle|\, L_0' \,\middle|\, t\, L_1 \circ \Phi_0 + (1-t)\, L_0 \circ \Phi_0\right)(x). \tag{1}$$

In simple terms, H moves a slice through $\Phi_0(x_0)$ to the slice witnessing $\overline{\operatorname{im} \Phi_1}$ as t goes from 0 to 1. A solution curve associated to (1) is a smooth map $x : [0, 1] \to \mathbb{C}^n$ such that $H(x(t), t) = 0$ for all t. For generic parameters L_0, L_1, L_0' the Jacobian $H_x(x, t)$ is invertible for all $t \in [0, 1]$, solution curves satisfy the ODE

$$x'(t) = -H_x(x, t)^{-1} H_t(x, t),$$

and each of the points $w_1, \ldots, w_e$ is the endpoint of some solution curve x with $x(0) \in X \cap L_0'$. These statements follow from more general results on *coefficient-parameter homotopy*, as presented in [27] or [33, Thm 7.1.1]. We assume a subroutine TRACK(H, x_0) which returns $x(1)$ for the solution curve based at x_0. In practice, the curve $x(t)$ is approximated by numerical predictor/corrector methods [1, 26]. We allow our TRACK routine to fail; this will occur, for instance, when $\Phi_0(x_0)$ is a singular point on $\overline{\operatorname{im} \Phi_0}$. However, it will succeed for generic (and hence *almost all*) choices of parameters and $x_0 \in \mathbb{C}^{n_0}$. Algorithm 1 exploits this fact.

Algorithm 1. Probability-1 equality test

Input: Let $X_0 \subset \mathbb{C}^{n_0}, X_1 \subset \mathbb{C}^{n_1}$ be irreducible algebraic varieties, and $\Phi_0 : X_0 \to \mathbb{C}^m$, $\Phi_1 : X_1 \to \mathbb{C}^m$ be rational maps, represented via the following ingredients:

1) $(L_1, \{w_1, \ldots, w_e\})$ with $\overline{\operatorname{im} \Phi_1} \cap L_1 = \{\Phi_1(w_1), \ldots, \Phi_e(w_e)\}$ and $e = \deg \overline{\operatorname{im} \Phi_1}$ (cf. Definition 2.1),
2) $f_{0,1}, \ldots, f_{0,c_0} \in \mathbb{C}[x_1, \ldots, x_{n_0}]$: a generically reduced regular sequence such that $\operatorname{codim}(X_0) = c_0$ and $X_0 \subset V(f_1, \ldots, f_{c_0})$,
3) an oracle for sampling a point $x_0 \in X_0$, and
4) explicit rational functions representing each map Φ_i.

Output: YES if $\overline{\operatorname{im} \Phi_0} = \overline{\operatorname{im} \Phi_1}$ and NO if $\overline{\operatorname{im} \Phi_0} \neq \overline{\operatorname{im} \Phi_1}$.

1: sample $x_0 \in X_0$
2: $T_{x_0}(f) \leftarrow \ker (D f)_{x_0}$
3: $d \leftarrow \operatorname{rank} (D \Phi_0)_{x_0}\big|_{T_{x_0}(f)}$
4: **if** $d \neq \dim \overline{\operatorname{im} \Phi_1}$ **then return** NO
5: $H(x; t) \leftarrow$ the homotopy from equation 1
6: $x_1 \leftarrow$ TRACK (H, x_0)
7: **if** $\Phi_0(x_1) \in \{\Phi_1(w_1), \ldots, \Phi_1(w_e)\}$ **return** YES
else return NO

Theorem 2.4. *For generic x_0, L_0, L_0', L_1, Algorithm 1 correctly decides if $\overline{\operatorname{im} \Phi_0} = \overline{\operatorname{im} \Phi_1}$.*

Remark 2.5. The set of "non-generic" L_1 depends on Φ_0 and Φ_1. In practice, an oracle for sampling generic points could be provided by either a parametrization or by homotopy continuation with known equations for X_0. The dimension $\dim \overline{\operatorname{im} \Phi_1}$ is implicit in the description of the witness set.

Proof. Since x_0 is generic and f_0 is generically reduced, we may assume that that $d = \dim \overline{\operatorname{im} \Phi_0}$. Noting line 4, we are done unless

$d = \dim \overline{\operatorname{im} \Phi_1}$. In this case, since the $\overline{\operatorname{im} \Phi_i}$ are irreducible,

$$\dim\left(\overline{\operatorname{im} \Phi_0} \cap \overline{\operatorname{im} \Phi_1}\right) = d \quad \Leftrightarrow \quad \overline{\operatorname{im} \Phi_0} = \overline{\operatorname{im} \Phi_1}. \tag{2}$$

As previously mentioned, generic slices give that the solution curve $x(t)$ associated to 1 with initial value x_0 exists and satisfies $x(t) \in V(f) \setminus V(f)_{\text{sing}}$ for all $t \in [0,1]$. The endpoint x_1 is, *a priori*, a point of $V(f)$. Since $X_0 \setminus (X_0)_{\text{sing}}$ is a connected component of $V(f) \setminus V(f)_{\text{sing}}$ in the complex topology and $x_0 \in X_0$, so also must $x_1 \in X_0$. Hence $\Phi_0(x_1) \in \overline{\operatorname{im} \Phi_0} \cap L_1$. Now if $\overline{\operatorname{im} \Phi_0} = \overline{\operatorname{im} \Phi_1}$, then clearly we must have

$$\Phi_0(x_1) \in \overline{\operatorname{im} \Phi_1} \cap L_1 = \{\Phi_1(w_1), \ldots, \Phi_1(w_e)\}, \tag{3}$$

as is tested on line 7. Conversely, if (3) holds, then

$$\dim(\overline{\operatorname{im} \Phi_0} \cap \overline{\operatorname{im} \Phi_1} \cap L_1) \geq 0,$$

which by (2) and the genericity of L_1 implies $\overline{\operatorname{im} \Phi_0} = \overline{\operatorname{im} \Phi_1}$. □

In the multiprojective setting, we may give a similar argument, noting that Proposition 2.3 and genericity of $\Phi_0(x_0)$ are needed so that $H_x(x_0, 0)$ is invertible.

3 SIGNATURES OF CURVES

3.1 Differential signatures

In what follows, all plane curves are complex algebraic, irreducible, and of degree greater than one. Let $G \subset \mathcal{PGL}_3(\mathbb{C})$ be a positive dimensional algebraic group acting linearly on $\mathbb{C}^2$ with action $g \cdot (x, y) = (\overline{x}, \overline{y})$.

Definition 3.1. Two curves C_0, C_1 are said to be *G-equivalent*, denoted $C_0 \cong_G C_1$, if there exists a $g \in G$ such that $C_0 = \overline{g \cdot C_1}$.

A differential signature that determines G-equivalence of curves can be constructed from a set of classifying invariants (Definition 3.6). We let J^n denote the nth order jet space, a complex vector space of dimension $(n+2)$ with coordinates $(x, y, y^{(1)}, \ldots, y^{(n)})$. Letting $\Omega(J^n)$ denote the set of complex-differentiable functions from J^n to $\mathbb{C}$, the *total derivative operator* $\frac{d}{dx} : \Omega(J^n) \to \Omega(J^{n+1})$ is the unique $\mathbb{C}$-linear map satisfying the product rule and the relations $\frac{d}{dx}(x) = 1$, $\frac{d}{dx}(y^{(k)}) = y^{(k+1)}$ for $k \geq 0$, cf. [28, Ch. 7].

The *prolonged* action of G on J^n is given by

$$g \cdot (x, y, y^{(1)}, \ldots, y^{(n)}) = (\overline{x}, \overline{y}, \overline{y}^{(1)}, \ldots, \overline{y}^{(n)})$$

where

$$\overline{y}^{(1)} = \frac{\frac{d}{dx}\left[\overline{y}(g, x, y)\right]}{\frac{d}{dx}\left[\overline{x}(g, x, y)\right]},$$

$$\overline{y}^{(k+1)} = \frac{\frac{d}{dx}\left[\overline{y}^{(k)}(g, x, y, y^{(1)}, \ldots, y^{(k)})\right]}{\frac{d}{dx}\left[\overline{x}(g, x, y)\right]} \text{ for } k = 1, \ldots, n-1.$$

Definition 3.2. A *differential invariant* for the action of G is a function on J^n that is invariant under the prolonged action of G on J^n. The *order* of a differential invariant is the maximum k such that the function depends explicitly on $y^{(k)}$.

Definition 3.3. The *n-th jet* of an algebraic curve C is the image of the map $j^n_C : C \dashrightarrow J^n$ given (where defined) by

$$(x, y) \mapsto (x, y, y^{(1)}_C(x, y), y^{(2)}_C(x, y), \ldots, y^{(n)}_C(x, y)),$$

where $y^{(k)}_C(x, y)$ is the k-th derivative of y with respect to x at the point $(x, y) \in C$.

The prolonged action of G is defined such that

$$g \cdot j^n_C(C) = j^n_{g \cdot C}(g \cdot C).$$

Definition 3.4. The *restriction* of a differential invariant K of order n to a curve C is the map $K|_C : C \dashrightarrow \mathbb{C}^2$ given by $K|_C = K \circ j^n_C$.

The coordinates of the n-th jet map j^n_C are rational functions of x and y that can be computed via implicit differentiation:

$$y^{(1)}_C = \frac{-\partial_x F}{\partial_y F} \quad \text{and} \quad y^{(k+1)}_C = \partial_x y^{(k)}_C + \partial_y y^{(k)}_C \, y^{(1)}_C. \tag{4}$$

where $\mathcal{I}_C = \langle F \rangle$. Thus, if K is a *rational* differential invariant of order n, meaning it is a rational function in the coordinates of J^n, then $K|_C$ is a rational function in x and y.

Definition 3.5. We say that a set of differential invariants $\mathcal{I}$ *separates orbits* for the prolonged action on a nonempty Zariski-open $W \subset J^n$ if, for all $p, q \in W$,

$$K(p) = K(q) \ \ \forall K \in \mathcal{I} \quad \Leftrightarrow \quad \exists g \in G \text{ such that } p = g \cdot q.$$

Definition 3.6. Let an r-dimensional algebraic group G act on $\mathbb{C}^2$. A pair of rational differential invariants $\mathcal{I} = \{K_1, K_2\}$ is said to be *classifying* if K_1 separates orbits on $U_k \subset J^k$ for some $k < r$ and $\mathcal{I}$ separates orbits on $U_r \subset J^r$.

For a particular action of G, such a pair of classifying invariants always exists, and one can explicitly construct a pair by computing generators for the field of rational invariants for the prolonged action of G [22, Thm 2.20], using algorithms such as those found in [9] and [20]. It should be noted that $\mathcal{I}$ is not unique, and different choices can lead to different differential signatures.

Definition 3.7. For a pair of classifying invariants $\mathcal{I} = \{K_1, K_2\}$, an algebraic curve C is said to be *non-exceptional* if all but finitely many points on $p \in C$ satisfy

$$j^k_C(p) \in U_k, \ j^r_C(p) \in U_r, \text{ and } \frac{\partial K_1}{\partial y^{(k)}}, \frac{\partial K_2}{\partial y^{(r)}} \neq 0 \text{ at } j^r_C(p).$$

A generic curve of degree d where $\binom{d+2}{2} - 2 \geq r$ is non-exceptional with respect to a given classifying set [22, Thm 2.27].

Definition 3.8. Let $\mathcal{I} = \{K_1, K_2\}$ be a pair of classifying invariants for the action of G on $\mathbb{C}^2$ and C a non-exceptional algebraic curve with respect to $\mathcal{I}$. Then the image of C under the map

$$(K_1|_C, K_2|_C) : C \dashrightarrow \mathbb{C}^2$$

is the *differential signature* of C and is denoted $\mathcal{S}_C$.

The following appears as Theorem 2.37 in [22].

Theorem 3.9. *If algebraic curves C_0, C_1 are non-exceptional with respect to a classifying set of rational differential invariants $\mathcal{I} = \{K_1, K_2\}$ under an action of G on $\mathbb{C}^2$ then*

$$C_0 \cong_G C_1 \quad \Leftrightarrow \quad \overline{\mathcal{S}_{C_0}} = \overline{\mathcal{S}_{C_1}}.$$

Example 3.10. Consider the action of the Euclidean group $\mathcal{E}_2$ of complex translations, rotations, and reflections on $\mathbb{C}^2$ where the action of $g \in \mathcal{E}_2(\mathbb{C})$ is given by

$g \cdot (x, y) = (cx + sy + a, -sx + cy + b)$ or $g \cdot (x, y) = (-cx + sy + a, sx + cy + b)$,

where $c^2 + s^2 = 1$ and $c, s, a, b \in \mathbb{C}$. The pair $\mathcal{I} = \{K_1, K_2\}$ defined below is derived from classical Euclidean curvature and is classifying for the action of $\mathcal{E}_2$. Here $y^{(1)} = y_x$, $y^{(2)} = y_{xx}$, and $y^{(3)} = y_{xxx}$:

$$K_1 = \frac{y_{xx}^2}{\left(1 + y_x^2\right)^3}, \quad K_2 = \frac{\left(y_{xxx}\left(1 + y_x^2\right) - 3y_x y_{xx}^2\right)^2}{\left(1 + y_x^2\right)^6} \tag{5}$$

Moreover, there are no $\mathcal{I}$-exceptional algebraic curves—for details see [31]. By Theorem 3.9, the equivalence class of an algebraic curve C under $\mathcal{E}_2(\mathbb{C})$ is determined by $\mathcal{S}_C$.

3.2 Joint signatures

In [30], the author considers the use of *joint* differential signatures to determine equivalence. As an example, for the action of G on $\mathbb{C}^2$ given by $g \cdot (x, y) = (\overline{x}, \overline{y})$, consider the induced action on the Cartesian product space $(\mathbb{C}^2)^n = \mathbb{C}^2 \times \mathbb{C}^2 \times \ldots \times \mathbb{C}^2$ given by

$$g \cdot (x_1, y_1, x_2, y_2, \ldots, x_n, y_n) = (\overline{x}_1, \overline{y}_1, \overline{x}_2, \overline{y}_2, \ldots, \overline{x}_n, \overline{y}_n)$$

where $\overline{x}_i = \overline{x}|_{x=x_i, y=y_i}$ and $\overline{y}_i = \overline{y}|_{x=x_i, y=y_i}$. For a curve $C \subset \mathbb{C}^2$ denote the Cartesian product by $C^n = C \times C \times \ldots \times C \subset (\mathbb{C}^2)^n$. Then we can see that two curves C_0 and C_1 are G-equivalent if and only if their Cartesian products C_0^n, C_1^n are G-equivalent under the induced action on $(\mathbb{C}^2)^n$.

The advantage of considering G-equivalence of products of the curve C is that the order of the differential invariants needed to define a differential signature on this space can be reduced. Though the number of invariants required may increase, the lower order of the differential invariants can result in a more noise-resistant differential signature. In fact, for a large enough product space, it is often possible to construct a differential signature from '0-th order' differential invariants, or *joint invariants*, which we refer to as a *joint signature*.

Consider the action of $\mathcal{E}_2(\mathbb{C})$ on $\mathbb{C}^2$ as defined in Example 3.10. This induces an action on the product space $(\mathbb{C}^2)^n$ whose joint invariants for this action are the squared inter-point distance functions

$$d_{jk}(x_j, y_j, x_k, y_k) = (x_j - x_k)^2 + (y_j - y_k)^2,$$

where $j < k$ and $j, k \in \{1, \ldots, n\}$. Let the map $d_n : C^n \to \mathbb{C}^{n(n-1)/2}$ be the map which takes an n-tuple of points on C and outputs all the inter-point distances, i.e.

$$(x_1, y_1, \ldots, x_n, y_n) \mapsto (d_{12}, d_{13}, \ldots, d_{1n}, \ldots, d_{(n-1)n}). \tag{6}$$

Additionally let W_n be the Zariski-open subset of $(\mathbb{C}^2)^n$ where all the inter-point distances do not vanish:

$$W_n = \{p \in (\mathbb{C}^2)^n \mid d_{jk}(p) \neq 0 \text{ for } j < k \text{ and } j, k \in \{1, \ldots, n\}\},$$

with the convention that $W_1 = \mathbb{C}^2$. To define a joint signature for curves under $\mathcal{E}_2(\mathbb{C})$, we take $n = 4$ and follow a similar construction as the joint signature of smooth curves in $\mathbb{R}^2$ under the action of $\mathcal{E}_2(\mathbb{R})$ (see [30, Ex. 8.2]).

Definition 3.11. The *Euclidean joint signature* of an algebraic curve $C \subset \mathbb{C}^2$ under the action of $\mathcal{E}_2(\mathbb{C})$, which we denote $\mathcal{J}_C$, is the image of the polynomial map $d_4 : C^4 \to \mathbb{C}^6$ defined as in (6).

We first show that these invariant functions characterize almost all orbits of the action of $\mathcal{E}_2(\mathbb{C})$ on $(\mathbb{C}^2)^3$ and $(\mathbb{C}^2)^4$.

Proposition 3.12. *The polynomial invariants $\mathcal{I}_3 = \{d_{12}, d_{13}, d_{23}\}$ separates orbits on W_3 for the induced action of $\mathcal{E}_2$ on $(\mathbb{C}^2)^3$ and the set*

$$\mathcal{I}_4 = \{d_{12}, d_{13}, d_{23}, d_{14}, d_{24}, d_{34}\}$$

separates orbits in W_4 for the induced action of $\mathcal{E}_2(\mathbb{C})$ on $(\mathbb{C}^2)^4$.

Proof. Consider two triples of points $p = (p_i)_{i=1}^3$ and $q = (q_i)_{i=1}^3 \in (\mathbb{C}^2)^3$, where $p_i = (x_i^p, y_i^p)$ and q_i is denoted similarly, that take the same values on $\mathcal{I}_3$ and lie in W_3. Note that W_3 excludes isotropic triples such as $(0, 0), (1, i), (1, -i)$. We will show that both triples of points necessarily lie in the same orbit. Since $d_{12} \neq 0$ we can choose a representative from the orbit of p under $\mathcal{E}_2$ such that $p_1 = (0, 0)$ and $p_2 = (0, y_2^p)$ by applying the transformation in $\mathcal{E}_2(\mathbb{C})$ given by

$$c = \frac{y_2^p - y_1^p}{\sqrt{d_{12}}}, \; s = \frac{x_2^p - x_1^p}{\sqrt{d_{12}}}, \; a = -x_1^p, \; b = -y_1^p, \tag{7}$$

and similarly we can assume for q that $q_1 = (0, 0)$ and $q_2 = (0, y_2^q)$. Since $p, q \in W_3, y_2^p, y_2^q \neq 0$. Thus $d_{12}(p) = d_{12}(q)$ gives that $(y_2^p)^2 = (y_2^q)^2$ meaning $y_2^p = \pm y_2^q$. Therefore, by reflecting about x-axis if necessary, we can assume $y_2^p = y_2^q$. The equations $d_{13}(p) = d_{13}(q)$ and $d_{23}(p) = d_{23}(q)$ give

$$(x_3^p)^2 + (y_3^p)^2 = (x_3^q)^2 + (y_3^q)^2$$
$$(x_3^p)^2 + (y_2^p - y_3^p)^2 = (x_3^q)^2 + (y_2^q - y_3^q)^2.$$

Subtracting these yields $(y_2^p)^2 - 2y_2^p y_3^p = (y_2^q)^2 - 2y_2^q y_3^q$ which implies $y_3^p = y_3^q$. Thus, from $d_{13}(p) = d_{13}(q)$, we have $(x_3^p)^2 = (x_3^q)^2$. From this we conclude, reflecting about the y-axis if necessary, that $x_3^p = x_3^q$. We have now shown that p and q are in the same orbit.

Suppose we have two 4-tuples of points $p = (p_i)_{i=1}^4$ and $q = (q_i)_{i=1}^4 \in (\mathbb{C}^2)^3$ that take the same values on $\mathcal{I}_4$ and lie in W_4. By the previous argument we can assume that p_1, p_2 have the same form as above and that $p_i = q_i$ for $i = 1, 2, 3$. As before the equations $d_{14}(p) = d_{14}(q)$ and $d_{24}(p) = d_{24}(q)$ imply that and $y_4^q = y_4^p$ and $x_4^p = \pm x_4^q$. If $x_4^p = -x_4^q$ and $x_3^p, x_3^q = 0$, then a reflection about the y-axis preserves the other values in q and sends x_4^q to $-x_4^q$. Otherwise subtracting the equations $d_{14}(p) = d_{14}(q)$ and $d_{34}(p) = d_{34}(q)$ yields $-2x_3^p x_4^p = -2x_3^q x_4^q$, which implies that $x_4^p = x_4^q$. Thus p and q must lie in the same orbit. □

Lemma 3.13. *For an algebraic curve $C \subset \mathbb{C}^2$ and $n > 1$, a generic n-tuple of points on C^n lies inside W_n. Additionally for any fixed $(n-1)$-tuple of points in $(p_1, \ldots, p_{n-1}) \in W_{n-1} \cap C^{n-1}$ and a generic point $p_n \in C$, the n-tuple $(p_1, \ldots, p_n)$ lies in W_n.*

Proof. For $n = 2$, fix any $p_1 = (x_1, y_1) \in C$. If $d_{1,2} = 0$ for all $(x_2, y_2) \in C$, then C must lie in a union of lines defined by

$$\{(x_2, y_2) \in \mathbb{C}^2 \mid (x_1 - x_2 + iy_1 - iy_2)(x_1 - x_2 - iy_1 + iy_2) = 0\}.$$

Since C is irreducible, this contradicts $\deg(C) > 1$. Thus the set $U_{2,p_1} = \{p_2 \in C \mid d_{1,2} \neq 0\}$, which is Zariski-open in C, is also nonempty. Thus, for any particular $p_1 \in C$, there exists p_2 with

$(p_1, p_2) \in W_2 \cap C^2$, from which both claims follow. Inductively, we fix any $(p_1, \ldots, p_{n-1}) \in W_{n-1} \cap C^{n-1}$. As before, the sets

$$U_{i,p_1,\ldots p_{n-1}} = \{p_n \in C \mid d_{in} \neq 0\}$$

are open and nonempty. Thus a generic $p_n \in C$ lies in their intersection, and hence $(p_1, \ldots, p_n) \in W_n$. □

Proposition 3.14. *The stabilizer of $p \in W_2$ or $p \in W_3$ under the action of $\mathcal{E}_2(\mathbb{C})$ is a finite subgroup.*

Proof. The stabilizer of a point $p \in (\mathbb{C}^2)^2$ is the subgroup of $\mathcal{E}_2(\mathbb{C})$ given by

$$\mathcal{E}_2(\mathbb{C})_p = \{g \in \mathcal{E}_2(\mathbb{C}) \mid g \cdot p = p\}.$$

The size of the stabilizer of a point is preserved by the action of the group. Since $d_{12}(p) \neq 0$, by applying the transformation in (7), we can assume p has the form $p = (p_1, p_2) = (0, 0, 0, y_2)$ where $y_2 \neq 0$. Given the parameterization of $\mathcal{E}_2(\mathbb{C})$ in Example 3.10, $g \cdot p = p$ immediately implies that $a = b = 0$ and that $sy_2 = 0$ Thus $\mathcal{E}_2(\mathbb{C})_p$ consists of either the identity transformation or a reflection about the y-axis. The same result immediately follows for $p \in W_3$, since $(p_1, p_2, p_3) \in W_3$ implies that $(p_1, p_2) \in W_2$. □

Lemma 3.15. *For plane curves C_0, C_1, suppose that there exists $p = (p_1, p_2) \in C_0^2, C_1^2$ such that $p \in W_2$ and*

$$d_3(p_1 \times p_2 \times C_0) = d_3(p_1 \times p_2 \times C_1).$$

Then there exists $g \in \mathcal{E}_2(\mathbb{C})$ such that $g \cdot C_0 = C_1$.

Proof. By Lemma 3.13, for a generic point $q \in C_0$, the 3-tuple $(p_1, p_2, q) \in W_3$. Since both curves have the same image under d_3, there exists a point $r \in C_1$ such that $r \in d_3^{-1}(p_1, p_2, q)$. By Proposition 3.12, both triples (p_1, p_2, q) and (p_1, p_2, r) lie in the same orbit under $\mathcal{E}_2(\mathbb{C})$, and hence there exists $g \in \mathcal{E}_2(\mathbb{C})$ such that $g \cdot (p_1, p_2, q) = (p_1, p_2, r)$. However, this implies that $g \in \mathcal{E}_2(\mathbb{C})_{(p_1,p_2)}$. By Proposition 3.14, $\mathcal{E}_2(\mathbb{C})_{(p_1,p_2)} = \{e, h\}$ where $h \in \mathcal{E}_2(\mathbb{C})$ is a reflection about the line containing p_1 and p_2. Therefore $q = r$ or $h \cdot q = r$, implying that C_1 shares infinitely many points with C_0 or $h \cdot C_0$, proving the lemma. □

Lemma 3.16. *For plane curves C_0, C_1, suppose that there exists a 3-tuple $p = (p_1, p_2, p_3) \in C_0^3, C_1^3$ such that $p \in W_3$ and*

$$d_4(p_1 \times p_2 \times p_3 \times C_0) = d_4(p_1 \times p_2 \times p_3 \times C_1).$$

Then there exists $g \in \mathcal{E}_2(\mathbb{C})$ such that $g \cdot C_0 = C_1$.

Proof. The proof follows similarly as in Lemma 3.15 by applying Propositions 3.12 and 3.14. □

Proposition 3.17. *Two plane curves $C_0, C_1 \subset \mathbb{C}^2$ of degree $d > 2$ are $\mathcal{E}_2(\mathbb{C})$-equivalent if and only if $\overline{\mathcal{J}_{C_0}} = \overline{\mathcal{J}_{C_1}}$.*

Proof. Since the map $d_4 : C_i^4 \to \mathbb{C}^6$ for $i = 0, 1$ is defined by $\mathcal{E}_2(\mathbb{C})$-invariants the forward direction is clear. For the remainder of the proof assume that $\overline{\mathcal{J}_{C_0}} = \overline{\mathcal{J}_{C_1}} := \mathcal{J}$. We deal with two cases. Either the image of the map $d_3 : C_0^3 \to \mathbb{C}^3$ lies in a Zariski-closed subset of dimension ≤ 2 or is Zariski-dense in $\mathbb{C}^3$.

First suppose that $d_3(C_0^3)$ (and hence $d_3(C_1^3)$) is Zariski-dense in $\mathbb{C}^3$. This implies $\dim(\mathcal{J})$ equals 3 or 4. Consider the projection $\pi_{12} : \mathcal{J} \to \mathbb{C}$ of $\mathcal{J}$ onto the first coordinate d_{12}. Let $\mathcal{H}_{12} = \pi_{12}^{(-1)}(r)$ be the pullback of a generic point so that $\dim(\mathcal{H}_{12} \cap \mathcal{J})$ equals 2 or 3. Appealing to Bertini's Theorem as in Proposition 2.3, the singular points of $\mathcal{H}_{12} \cap \mathcal{J}$ are also singular points of $\mathcal{J}$. For similarly defined $\mathcal{H}_{13}$ and $\mathcal{H}_{23}$ let $\mathcal{Y} = \mathcal{H}_{12} \cap \mathcal{H}_{13} \cap \mathcal{H}_{23} \cap \mathcal{J}$. Then $\dim(\mathcal{Y})$ equals 0 or 1, and the singular points of $\mathcal{Y}$ are singular points of $\mathcal{J}$.

Consider a generic 4-tuple of points $p = (p_1, p_2, p_3, p_4) \in C_0^4$. Since the $d_4(C_i)$ agree on a dense set, we may assume $d_4(p) \in d_4(C_0) \cap d_4(C_1)$. Taking generic $\mathcal{H}_{12} \cap \mathcal{H}_{13} \cap \mathcal{H}_{23}$ through $d_3(p)$ and $\mathcal{Y}$ as in the previous paragraph, we have that $d_4(p)$ is a non-singular point of $\mathcal{Y}$. Let $q = (q_1, q_2, q_3, q_4)$ be a point on C_1^4 in the inverse image $d_4^{-1}(d_4(p))$. By Proposition 3.12 and Lemma 3.13, there exists some $g \in \mathcal{E}_2(\mathbb{C})$ such that $g \cdot q = p$. Let $C_2 = g \cdot C_1$.

Note that $\dim(d_4(p_1 \times p_2 \times p_3 \times C_0)) = 0$ implies that the function $d_{14}(x) = d_{14}(p_1, p_2, p_3, x)$ is constant on C_0, and similarly so are $d_{24}(x)$ and $d_{34}(x)$. By Proposition 3.12, for a generic point $x \in C_0$, the 4-tuples (p_1, p_2, p_3, x) are all related by an element of $\mathcal{E}_2(\mathbb{C})$. However this is a contradiction, since by Proposition 3.14 there are finitely many such elements. Thus both sets $d_4(p_1 \times p_2 \times p_3 \times C_0)$ and (by a similar argument) $d_4(p_1 \times p_2 \times p_3 \times C_1)$ are of dimension 1 lying in $\mathcal{Y}$. Since $\dim(\mathcal{Y}) = 1$, both sets are also dense in some irreducible component of $\mathcal{Y}$. Since $d_4(p)$ is a non-singular point of $\mathcal{Y}$, it is necessarily contained in exactly one irreducible component of $\mathcal{Y}$. Therefore

$$d_4(p_1 \times p_2 \times p_3 \times C_0) = d_4(p_1 \times p_2 \times p_3 \times C_2).$$

By Lemma 3.16, $C_0 = C_2 = g \cdot C_1$, completing the proof for the case where $d_3(C_0^3) \subset \mathbb{C}^3$ is Zariski dense. The remaining case follows analogously (take $\mathcal{Y} = \mathcal{H}_{12} \cap d_3(C_0^3)$ and apply Lemma 3.15.) □

4 IMPLEMENTATION, EXAMPLES, AND EXPERIMENTS

Our implementation of Algorithm 1 treats only the special case where the domain of each rational map is some Cartesian product of irreducible plane curves, say $X_i = C_i^k$ for some integer k. Our results showcase features of the `NumericalAlgebraicGeometry` ecosystem in Macaulay2 (aka `NAG4M2`, see [23, 24] for an overview.) We rely extensively on the core path-tracker and the packages `SLPexpressions` and `MonodromySolver`. All of our examples and experiments deal with differential and joint signatures for the Euclidean group.[2] However, the current functionality should make it easy to study other group actions and variations on the signature construction in the future.

For the purpose of our implementation, the various ingredients for the input to Algorithm 1 are easily provided. Suppose $\mathcal{I}_{C_i} = \langle f_i \rangle$ for $i = 0, 1$. Then the reduced regular sequence we need is given by $(f_0(x_1, y_1), \ldots, f_0(x_k, y_k))$. Sampling from X_0 amounts to sampling k times from C_0; we sample the curve C_0 using homotopy continuation from a linear-product start system [33, 8.4.3]. Finally, a witness set for the image of the signature map Φ_1 can be computed using methods of numerical algebraic geometry. Heuristics based on *monodromy* allow us to make this offline computation relatively efficient; `MonodromySolver` implements a general framework described in [5, 10]. We also observe that a witness set for the signature of a particular curve may be computed if we have already computed a witness set for the corresponding signature of some *generic* curve of the same degree. This is yet another application of

[2] For details we refer to the code: https://github.com/timduff35/NumericalSignatures.

d	$\deg \mathcal{S}$	time (s)	$\deg_{(1,0)} \mathcal{S}$	time (s)
2	6	0.3	3	0.1
3	72	2	36	0.5
4	144	9	72	2
5	240	21	120	4
6	360	55	180	7

Figure 2: Degrees and monodromy timings for differential signatures.

d	$\deg \mathcal{J}$	time (s)	$\deg_{\boldsymbol{e}_1} \mathcal{J}$	time (s)	$\deg_{\boldsymbol{e}_2} \mathcal{J}$	time (s)
2	42	4	24	2	26	2
3	936	33	576	17	696	16
4	3024	139	1920	57	2448	87
5	7440	463	4800	206	6320	276
6	15480	1315	10080	748	13560	791

Figure 3: Degrees and monodromy timings for joint signatures (see Conjecture 4.2.)

d	track time (ms)	lookup time (ms)	track K_1	lookup K_1
2	191	0.35	127	0.25
3	177	0.37	121	0.31
4	276	0.42	145	0.36
5	472	0.39	203	0.43
6	597	0.40	284	0.37

Figure 4: Equality test timings for differential signatures S_d.

d	track time (ms)	lookup time (ms)	track $\boldsymbol{e}_1$	lookup $\boldsymbol{e}_1$
2	230	0.36	208	0.34
3	283	0.38	213	0.35
4	335	0.39	288	0.40
5	409	0.32	357	0.32
6	507	0.32	462	0.33

Figure 5: Equality test timings for joint signatures $\mathcal{J}_d$.

coefficient parameter homotopy. [27] The efficiency of these two methods is compared in Example 4.1.

We explain some aspects of our implementation that appear to give reasonable numerical stability. A key feature is that polynomials and rational maps are given by straight-line programs as opposed to their coefficient representations. This is especially crucial in the case of differential signatures, where we can do efficient evaluation using the formulas in equation 4; we note that expanding these rational functions in the monomial basis involves many terms and does not suggest a natural evaluation scheme. We also homogenize the equations of our plane curves and work in a random affine chart. Finally, in our sampling procedure we discard samples which map too close to the origin in the codomain of our maps, as these tend to produce nearly-singular points on the image.

Example 4.1. The code below computes a witness set for the differential signature of a "generic" quartic (whose coefficients are random complex numbers of modulus 1.)

```
(d, k) = (4, 1);
dom = domain(d, k);
Map = diffEuclideanSigMap dom;
H = witnessHomotopy(dom, Map);
W = runMonodromy H;
```

To compute a witness set for the differential signature of the Fermat quartic $V(x^4+y^4+z^4) \subset \mathbb{P}(\mathbb{C}^3)$, we use the previous computation.

```
R = QQ[x,y,z];
f=x^4+y^4+z^4;
Wf = witnessCollect(f, W)
```

The output resulting from the last line reads

```
witness data w/ 18 image points (144 preimage points)
```

indicating that the differential signature map is generically 8 to 1, which is equivalent to the Fermat curve having eight Euclidean symmetries [22, Thm 2.38]. We timed these witness set computations at 5 and 0.5 seconds, respectively. For joint signatures, the analagous computations were timed at 95 and 17 seconds.

Figures 2 and 3 give degrees and single-run timings for monodromy computations on curves up to degree 6. We also considered multiprojective witness sets for $\mathcal{S} \subset \mathbb{C}^1 \times \mathbb{C}^1$ and $\mathcal{J} \subset (\mathbb{C}^1)^6$, where fewer witness points are needed. For the differential signatures, we considered $(1,0)$-slices which fix the value of K_1 in (5). For joint signatures, there are two combinatorially distinct classes of $(\mathbb{C}^1)^6$ witness sets determined by which $d_{i,j}$ are fixed; the undirected graph of fixed distances must either be the 3-pan (a 3-cycle with pendant edge) or the 4-cycle. We fix corresponding multidimensions $\boldsymbol{e}_1 = (1,1,1,1,0,0)$ and $\boldsymbol{e}_2 = (0,1,1,1,1,0)$.

The timings in figures 2 and 3 are not optimal for a number of reasons. For instance, some multiprojective witness sets have an *imprimitive* monodromy action, meaning that additional symmetries can be exploited [2]. We successfully ran monodromy (with less conservative settings) for both signature maps on curves of degree up to 10. These computations suggested formulas for the degrees. For the joint signature, we state these formulas in the form of a conjecture. For the case of differential signatures, see [22]; degrees for $d = 2$ are corrected by a factor of 4.

Conjecture 4.2. *Let $\mathcal{J}_d$ denote the joint signature for a generic plane curve of degree d. For $d \geq 3$:*

$\deg \overline{\mathcal{J}_d} = 12d(d^3-1)$
$\deg_{\boldsymbol{e}_1} \mathcal{J}_d = 8d^2(d^2-1)$
$\deg_{\boldsymbol{e}_2} \mathcal{J}_d = 4d(d-1)(3d^2+d-1)$.

To assess the speed and robustness of the online equality test, we conducted an experiment where, for degrees $d = 2, \ldots, 6$, curves $C_1, \ldots, C_{10}$ were generated with coefficients drawn uniformly from the unit sphere in $\mathbb{R}^{(d+2)(d+1)/2}$. For each C_i, we computed a witness set via parameter homotopy from a generic degree d curve. We then applied 20 random transformations from $\mathcal{E}_2(\mathbb{R})$ to the C_i and perturbed the resulting coefficients by random real $\vec{\epsilon}$ with $\|\vec{\epsilon}\|_2 \in \{0, 10^{-7}, 10^{-6}, \ldots, 10^{-3}\}$, thus obtaining curves $\widetilde{C_{i,1,\epsilon}}, \ldots, \widetilde{C_{i,20,\epsilon}}$. With all numerical tolerances fixed, we ran the equality test for each $\widetilde{C_{i,j,\epsilon}}$ against each C_i.

Figures 4 and 5 summarize the timings for the equality tests in this experiment. Overall, these tests run on the order of sub-seconds.

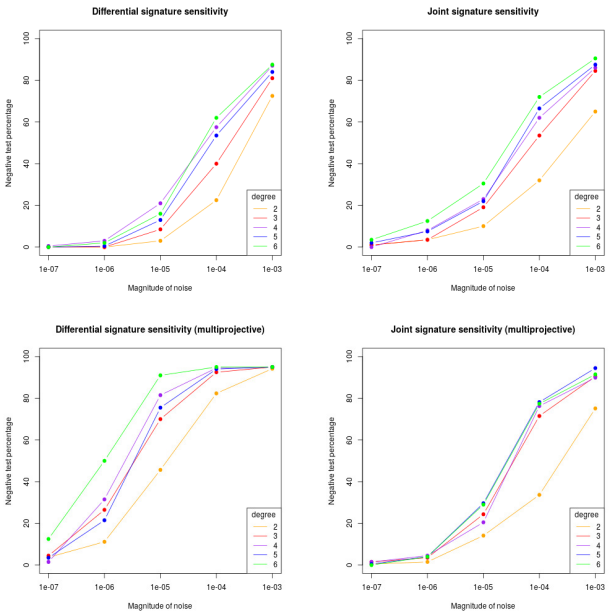


Figure 6: Sensitivity of the equality test to noise.

Most of the time is spent on path-tracking. The tracking times reported give the total time spent on lines 1 and 5 of Algorithm 1. The only other possible bottleneck is the lookup on line 7. This is negligible, even for large witness set sizes, if an appropriate data structure is used. The runtimes for all cases considered seem comparable, although using differential signatures and multiprojective slices appear to give a slight edge over the respective alternatives.

The plots in Figure 6 illustrate the results of our sensitivity analysis. The respective axes are the magnitude of the noise ϵ and the percentage of $C_{i,j,\epsilon}$ deemed to be not equivalent to C_i. Note that the horizontal axis is given on a log scale, and excludes the noiseless case $\epsilon = 0$; for this case, among all tests in the experiment, only one false negative was reported for the differential signatures with $d = 6$. We include a trend line to make the plots more readable. In general, we observe a threshold phenomenon, where most tests are positive for sufficiently low noise and are negative for sufficiently high noise.

The thresholds displayed in Figure 6 clearly depend on the numerical tolerances used (for this experiment, defaults provided by `NAG4M2`), the type of map, and the type of witness set. Besides the multiprojective differential signature (depicted in the bottom-left), we observe a similar stability profile for this type of random perturbation. We speculate that similar analyses, based on a more meaningful model of noise, may highlight further differences between the joint and differential signatures.

ACKNOWLEDGMENTS

Research of T. Duff is supported in part by NSF DMS-1719968, a fellowship from the Algorithms and Randomness Center at Georgia Tech, and by the Max Planck Institute for Mathematics in the Sciences in Leipzig.

REFERENCES

[1] E. L. Allgower and K. Georg. 2012. *Numerical continuation methods: an introduction.* Vol. 13. Springer Science & Business Media.
[2] C. Améndola and J. I. Rodriguez. 2016. Solving parameterized polynomial systems with decomposable projections. *arXiv preprint arXiv:1612.08807* (2016).
[3] D. J. Bates, , A. J. Hauenstein, Jonathan D Sommese, and C. W. Wampler. 2013. *Numerically solving polynomial systems with Bertini.* SIAM.
[4] I. A. Berchenko (Kogan) and P. J. Olver. 2000. Symmetries of Polynomials. *Journal of Symbolic Computations* 29 (2000), 485–514.
[5] N. Bliss, T. Duff, A. Leykin, and J. Sommars. 2018. Monodromy solver: sequential and parallel. In *Proceedings of the 2018 ACM International Symposium on Symbolic and Algebraic Computation.* 87–94.
[6] T. Brysiewicz. 2018. Numerical Software to Compute Newton Polytopes. In *International Congress on Mathematical Software.* Springer, 80–88.
[7] J. M. Burdis, I. A. Kogan, and H. Hong. 2013. Object-image correspondence for algebraic curves under projections. *SIGMA Symmetry Integrability Geom. Methods Appl.* 9 (2013), Paper 023, 31.
[8] J. Chen and J. Kileel. 2019. Numerical implicitization for Macaulay2. *Journal of Software for Algebra and Geometry* 9 (2019), 55–65.
[9] H. Derksen and G. Kemper. 2015. *Computational invariant theory* (enlarged ed.). Encyclopaedia of Mathematical Sciences, Vol. 130. Springer, Heidelberg. xxii+366 pages.
[10] T. Duff, C. Hill, A. Jensen, K. Lee, A. Leykin, and J. Sommars. 2019. Solving polynomial systems via homotopy continuation and monodromy. *IMA J. Numer. Anal.* 39, 3 (2019), 1421–1446.
[11] M. Fels and P. J. Olver. 1999. Moving Coframes. II. Regularization and Theoretical Foundations. *Acta Appl. Math.* 55 (1999), 127–208.
[12] D. Grayson and M. Stillman. 1997. Macaulay 2–a system for computation in algebraic geometry and commutative algebra.
[13] A. Grim and C. Shakiban. 2017. Applications of signature curves to characterize melanomas and moles. In *Applications of computer algebra.* Springer Proc. Math. Stat., Vol. 198. Springer, Cham, 171–189.
[14] J. Harris. 2013. *Algebraic geometry: a first course.* Vol. 133. Springer Science & Business Media.
[15] J. D. Hauenstein, A. Leykin, J. I. Rodriguez, and F. Sottile. 2019. A numerical toolkit for multiprojective varieties. *To appear in Mathematics of Computation* (2019).
[16] J. D. Hauenstein and J. I. Rodriguez. 2019. Multiprojective witness sets and a trace test. *To appear in Advances in Geometry. arXiv preprint arXiv:1507.07069* (2019).
[17] J. D. Hauenstein and A. J. Sommese. 2010. Witness sets of projections. *Appl. Math. Comput.* 217, 7 (2010), 3349–3354.
[18] J. D. Hauenstein and A. J. Sommese. 2013. Membership tests for images of algebraic sets by linear projections. *Appl. Math. Comput.* 219, 12 (2013), 6809–6818.
[19] D. J. Hoff and P. J. Olver. 2014. Automatic solution of jigsaw puzzles. *J. Math. Imaging Vision* 49, 1 (2014), 234–250.
[20] E. Hubert and I. A. Kogan. 2007. Smooth and algebraic invariants of a group action: local and global construction. *Foundation of Computational Math. J.* 7:4 (2007), 345–383.
[21] I. A. Kogan and M. Moreno Maza. 2002. Computation of canonical forms for ternary cubics. In *Proceedings of the 2002 International Symposium on Symbolic and Algebraic Computation.* ACM, New York, 151–160.
[22] I. A. Kogan, M. Ruddy, and C. Vinzant. 2020. Differential Signatures of Algebraic Curves. *SIAM J. Appl. Algebra Geom.* 4, 1 (2020), 185–226.
[23] A. Leykin. 2011. Numerical algebraic geometry. *Journal of Software for Algebra and Geometry* 3, 1 (2011), 5–10.
[24] A. Leykin. 2018. Homotopy Continuation in Macaulay2. In *International Congress on Mathematical Software.* Springer, 328–334.
[25] A. Leykin, J. I. Rodriguez, and F. Sottile. 2018. Trace test. *Arnold Mathematical Journal* 4, 1 (2018), 113–125.
[26] A. Morgan. 2009. *Solving polynomial systems using continuation for engineering and scientific problems.* Vol. 57. SIAM.
[27] A. P. Morgan and A. J. Sommese. 1989. Coefficient-parameter polynomial continuation. *Appl. Math. Comput.* 29, 2 (1989), 123–160.
[28] P. J. Olver. 1995. *Equivalence, invariants and symmetry.* Cambridge University Press.
[29] P. J. Olver. 1999. *Classical invariant theory.* London Mathematical Society Student Texts, Vol. 44. Cambridge University Press, Cambridge. xxii+280 pages.
[30] P. J. Olver. 2001. Joint invariant signatures. *Found. Comput. Math.* 1, 1 (2001), 3–67.
[31] M. Ruddy. 2019. *The Equivalence Problem and Signatures of Algebraic Curves.* Ph.D. Dissertation. North Carolina State University.
[32] A. J. Sommese, J. Verschelde, and C. W. Wampler. 2005. Introduction to numerical algebraic geometry. In *Solving polynomial equations.* Springer, 301–337.
[33] I. C. W. Wampler et al. 2005. *The Numerical solution of systems of polynomials arising in engineering and science.* World Scientific.

On Fast Multiplication of a Matrix by its Transpose

Jean-Guillaume Dumas
Université Grenoble Alpes
Laboratoire Jean Kuntzmann, CNRS
UMR 5224, 38058 Grenoble, France

Clément Pernet
Université Grenoble Alpes
Laboratoire Jean Kuntzmann, CNRS
UMR 5224, 38058 Grenoble, France

Alexandre Sedoglavic
Université de Lille
UMR CNRS 9189 CRISTAL
59650 Villeneuve d'Ascq, France

ABSTRACT

We present a non-commutative algorithm for the multiplication of a 2×2-block-matrix by its transpose using 5 block products (3 recursive calls and 2 general products) over $\mathbb{C}$ or any field of prime characteristic. We use geometric considerations on the space of bilinear forms describing 2×2 matrix products to obtain this algorithm and we show how to reduce the number of involved additions. The resulting algorithm for arbitrary dimensions is a reduction of multiplication of a matrix by its transpose to general matrix product, improving by a constant factor previously known reductions. Finally we propose schedules with low memory footprint that support a fast and memory efficient practical implementation over a prime field. To conclude, we show how to use our result in $L \cdot D \cdot L^{\mathsf{T}}$ factorization.

CCS CONCEPTS

• **Computing methodologies** → **Exact arithmetic algorithms**; **Linear algebra algorithms**.

KEYWORDS

algebraic complexity, fast matrix multiplication, SYRK, rank-k update, Symmetric matrix, Gram matrix, Wishart matrix

ACM Reference Format:
Jean-Guillaume Dumas, Clément Pernet, and Alexandre Sedoglavic. 2020. On Fast Multiplication of a Matrix by its Transpose. In *International Symposium on Symbolic and Algebraic Computation (ISSAC '20), July 20–23, 2020, Kalamata, Greece.* ACM, New York, NY, USA, 8 pages. https://doi.org/10.1145/3373207.3404021

1 INTRODUCTION

Strassen's algorithm [20], with 7 recursive multiplications and 18 additions, was the first sub-cubic time algorithm for matrix product, with a cost of $O(n^{2.81})$. Summarizing the many improvements which have happened since then, the cost of multiplying two arbitrary $n \times n$ matrices $O(n^\omega)$ will be denoted by $\mathrm{MM}_\omega(n)$ (see [17] for the best theoretical value of ω known to date).

We propose a new algorithm for the computation of the product $A \cdot A^{\mathsf{T}}$ of a 2×2-block-matrix by its transpose using only 5 block multiplications over some base field, instead of 6 for the natural divide & conquer algorithm. For this product, the best previously known complexity bound was dominated by $\frac{2}{2^\omega - 4}\mathrm{MM}_\omega(n)$ over any field (see [11, § 6.3.1]). Here, we establish the following result:

ISSAC '20, July 20–23, 2020, Kalamata, Greece

ACM ISBN 978-1-4503-7100-1/20/07…$15.00
https://doi.org/10.1145/3373207.3404021

Theorem 1.1. *The product of an $n \times n$ matrix by its transpose can be computed in $\frac{2}{2^\omega - 3}\mathrm{MM}_\omega(n)$ field operations over a base field for which there exists a skew-orthogonal matrix.*

Our algorithm is derived from the class of Strassen-like algorithms multiplying 2×2 matrices in 7 multiplications. Yet it is a reduction of multiplying a matrix by its transpose to general matrix multiplication, thus supporting any admissible value for ω. By exploiting the symmetry of the problem, it requires about half of the arithmetic cost of general matrix multiplication when ω is $\log_2 7$.

We focus on the computation of the product of an $n \times k$ matrix by its transpose and possibly accumulating the result to another matrix. Following the terminology of the BLAS3 standard [10], this operation is a symmetric rank k update (SYRK for short).

2 MATRIX PRODUCT ALGORITHMS ENCODED BY TENSORS

Considered as 2×2 matrices, the matrix product $C = A \cdot B$ could be computed using Strassen algorithm by performing the following computations (see [20]):

$$\begin{array}{ll} \rho_1 \leftarrow a_{11}(b_{12} - b_{22}), & \\ \rho_2 \leftarrow (a_{11} + a_{12})b_{22}, & \rho_4 \leftarrow (a_{12} - a_{22})(b_{21} + b_{22}), \\ \rho_3 \leftarrow (a_{21} + a_{22})b_{11}, & \rho_5 \leftarrow (a_{11} + a_{22})(b_{11} + b_{22}), \\ \rho_6 \leftarrow a_{22}(b_{21} - b_{11}), & \rho_7 \leftarrow (a_{21} - a_{11})(b_{11} + b_{12}), \\ \multicolumn{2}{c}{\begin{pmatrix} c_{11} & c_{12} \\ c_{21} & c_{22} \end{pmatrix} = \begin{pmatrix} \rho_5+\rho_4-\rho_2+\rho_6 & \rho_6+\rho_3 \\ \rho_2+\rho_1 & \rho_5+\rho_7+\rho_1-\rho_3 \end{pmatrix}.} \end{array} \tag{1}$$

In order to consider this algorithm under a geometric standpoint, we present it as a tensor. Matrix multiplication is a bilinear map:

$$\begin{array}{ccc} \mathbb{K}^{m\times n} \times \mathbb{K}^{n\times p} & \to & \mathbb{K}^{m\times p}, \\ (X, Y) & \to & X \cdot Y, \end{array} \tag{2}$$

where the spaces $\mathbb{K}^{a\times b}$ are finite vector spaces that can be endowed with the Frobenius inner product $\langle M, N\rangle = \mathrm{Trace}(M^{\mathsf{T}} \cdot N)$. Hence, this inner product establishes an isomorphism between $\mathbb{K}^{a\times b}$ and its dual space $(\mathbb{K}^{a\times b})^\star$ allowing for example to associate matrix multiplication and the trilinear form $\mathrm{Trace}(Z^{\mathsf{T}} \cdot X \cdot Y)$:

$$\begin{array}{ccc} \mathbb{K}^{m\times n} \times \mathbb{K}^{n\times p} \times (\mathbb{K}^{m\times p})^\star & \to & \mathbb{K}, \\ (X, Y, Z^{\mathsf{T}}) & \to & \langle Z, X \cdot Y\rangle. \end{array} \tag{3}$$

As by construction, the space of trilinear forms is the canonical dual space of order three tensor product, we could associate the Strassen multiplication algorithm (1) with the tensor $\mathcal{S}$ defined by:

$$\begin{array}{l} \sum_{i=1}^{7} S_{i1}\otimes S_{i2}\otimes S_{i3} = \left(\begin{smallmatrix}1&0\\0&0\end{smallmatrix}\right)\otimes\left(\begin{smallmatrix}0&1\\0&-1\end{smallmatrix}\right)\otimes\left(\begin{smallmatrix}0&0\\1&1\end{smallmatrix}\right)+ \\ \left(\begin{smallmatrix}1&1\\0&0\end{smallmatrix}\right)\otimes\left(\begin{smallmatrix}0&0\\0&1\end{smallmatrix}\right)\otimes\left(\begin{smallmatrix}-1&0\\1&0\end{smallmatrix}\right)+\left(\begin{smallmatrix}0&0\\1&1\end{smallmatrix}\right)\otimes\left(\begin{smallmatrix}1&0\\0&0\end{smallmatrix}\right)\otimes\left(\begin{smallmatrix}0&1\\0&-1\end{smallmatrix}\right)+ \\ \left(\begin{smallmatrix}0&1\\0&-1\end{smallmatrix}\right)\otimes\left(\begin{smallmatrix}0&0\\1&1\end{smallmatrix}\right)\otimes\left(\begin{smallmatrix}1&0\\0&0\end{smallmatrix}\right)+\left(\begin{smallmatrix}1&0\\0&1\end{smallmatrix}\right)\otimes\left(\begin{smallmatrix}1&0\\0&1\end{smallmatrix}\right)\otimes\left(\begin{smallmatrix}1&0\\0&1\end{smallmatrix}\right)+ \\ \left(\begin{smallmatrix}0&0\\0&1\end{smallmatrix}\right)\otimes\left(\begin{smallmatrix}-1&0\\1&0\end{smallmatrix}\right)\otimes\left(\begin{smallmatrix}1&1\\0&0\end{smallmatrix}\right)+\left(\begin{smallmatrix}-1&0\\1&0\end{smallmatrix}\right)\otimes\left(\begin{smallmatrix}1&1\\0&0\end{smallmatrix}\right)\otimes\left(\begin{smallmatrix}0&0\\0&1\end{smallmatrix}\right) \end{array} \tag{4}$$

in $(\mathbb{K}^{m\times n})^\star \otimes (\mathbb{K}^{n\times p})^\star \otimes \mathbb{K}^{m\times p}$ with $m=n=p=2$. Given any couple (A,B) of 2×2-matrices, one can explicitly retrieve from tensor $\mathcal{S}$ the Strassen matrix multiplication algorithm computing $A\cdot B$ by the *partial* contraction $\{\mathcal{S}, A\otimes B\}$:

$$\begin{gathered}\Big((\mathbb{K}^{m\times n})^\star\otimes(\mathbb{K}^{n\times p})^\star\otimes\mathbb{K}^{m\times p}\Big)\otimes\big(\mathbb{K}^{m\times n}\otimes\mathbb{K}^{n\times p}\big)\to\mathbb{K}^{m\times p},\\ \mathcal{S}\otimes(A\otimes B)\to\textstyle\sum_{i=1}^{7}\langle S_{i1},A\rangle\langle S_{i2},B\rangle S_{i3},\end{gathered}\tag{5}$$

while the *complete* contraction $\{\mathcal{S}, A\otimes B\otimes C^\intercal\}$ is $\mathrm{Trace}(A\cdot B\cdot C)$.

The tensor formulation of matrix multiplication algorithm gives explicitly its symmetries (a.k.a. *isotropies*). As this formulation is associated to the trilinear form $\mathrm{Trace}(A\cdot B\cdot C)$, given three invertible matrices U,V,W of suitable sizes and the classical properties of the trace, one can remark that $\mathrm{Trace}(A\cdot B\cdot C)$ is equal to:

$$\begin{gathered}\mathrm{Trace}\big((A\cdot B\cdot C)^\intercal\big)=\mathrm{Trace}(C\cdot A\cdot B)=\mathrm{Trace}(B\cdot C\cdot A),\\ \text{and }\mathrm{Trace}\big(U^{-1}\cdot A\cdot V\cdot V^{-1}\cdot B\cdot W\cdot W^{-1}\cdot C\cdot U\big).\end{gathered}\tag{6}$$

These relations illustrate the following theorem:

Theorem 2.1 ([8, § 2.8]). *The isotropy group of the $n\times n$ matrix multiplication tensor is $\mathrm{PSL}^\pm(\mathbb{K}^n)^{\times 3}\rtimes\mathfrak{S}_3$, where PSL stands for the group of matrices of determinant ±1 and $\mathfrak{S}_3$ for the symmetric group on 3 elements.*

The following definition recalls the *sandwiching* isotropy on matrix multiplication tensor:

Definition 2.1. *Given $\mathrm{g}=(U\times V\times W)$ in $\mathrm{PSL}^\pm(\mathbb{K}^n)^{\times 3}$, its action $\mathrm{g}\diamond\mathcal{S}$ on a tensor $\mathcal{S}$ is given by $\sum_{i=1}^{7}\mathrm{g}\diamond(S_{i1}\otimes S_{i2}\otimes S_{i3})$ where the term $\mathrm{g}\diamond(S_{i1}\otimes S_{i2}\otimes S_{i3})$ is equal to:*

$$(U^{-\intercal}\cdot S_{i1}\cdot V^\intercal)\otimes(V^{-\intercal}\cdot S_{i2}\cdot W^\intercal)\otimes(W^{-\intercal}\cdot S_{i3}\cdot U^\intercal).\tag{7}$$

Remark 2.1. *In $\mathrm{PSL}^\pm(\mathbb{K}^n)^{\times 3}$, the product $\circ$ of two isotropies g_1 defined by $u_1\times v_1\times w_1$ and g_2 by $u_2\times v_2\times w_2$ is the isotropy $g_1\circ g_2$ equal to $u_1\cdot u_2\times v_1\cdot v_2\times w_1\cdot w_2$. Furthermore,the complete contraction $\{g_1\circ g_2, A\otimes B\otimes C\}$ is equal to $\{g_2, {g_1}^\intercal\diamond A\otimes B\otimes C\}$.*

The following theorem shows that all 2×2-matrix product algorithms with 7 coefficient multiplications could be obtained by the action of an isotropy on Strassen tensor:

Theorem 2.2 ([9, § 0.1]). *The group $\mathrm{PSL}^\pm(\mathbb{K}^n)^{\times 3}$ acts transitively on the variety of optimal algorithms for the computation of 2×2-matrix multiplication.*

Thus, isotropy action on Strassen tensor may define other matrix product algorithm with interesting computational properties.

2.1 Design of a specific 2×2-matrix product

This observation inspires our general strategy to design specific algorithms suited for particular matrix product.

Strategy 2.1. *By applying an undetermined isotropy:*

$$\mathrm{g}=U\times V\times W=\begin{pmatrix}u_{11}&u_{12}\\u_{21}&u_{22}\end{pmatrix}\times\begin{pmatrix}v_{11}&v_{12}\\v_{21}&v_{22}\end{pmatrix}\times\begin{pmatrix}w_{11}&w_{12}\\w_{21}&w_{22}\end{pmatrix}\tag{8}$$

on Strassen tensor $\mathcal{S}$, we obtain a parameterization $\mathcal{T}=\mathrm{g}\diamond\mathcal{S}$ of all matrix product algorithms requiring 7 coefficient multiplications:

$$\mathcal{T}=\sum_{i=1}^{7}T_{i1}\otimes T_{i2}\otimes T_{i3},\ T_{i1}\otimes T_{i2}\otimes T_{i3}=\mathrm{g}\diamond S_{i1}\otimes S_{i2}\otimes S_{i3}.\tag{9}$$

Then, we could impose further conditions on these algorithms and check by a Gröbner basis computation if such an algorithm exists. If so, there is subsequent work to do for choosing a point on this variety; this choice can be motivated by the additive cost bound and the scheduling property of the evaluation scheme given by this point.

Let us first illustrate this strategy with the well-known Winograd variant of Strassen algorithm presented in [22].

Example 1. *Apart from the number of multiplications, it is also interesting in practice to reduce the number of additions in an algorithm. Matrices S_{11} and S_{61} in tensor (4) do not increase the additive cost bound of this algorithm. Hence, in order to reduce this complexity in an algorithm, we could try to maximize the number of such matrices involved in the associated tensor. To do so, we recall Bshouty's results on additive complexity of matrix product algorithms.*

Theorem 2.3 ([6]). *Let $e_{(i,j)}=(\delta_{i,k}\delta_{j,l})_{(k,l)}$ be the single entry elementary matrix. A 2×2 matrix product tensor could not have 4 such matrices as first (resp. second, third) component ([6, Lemma 8]). The additive complexity bound of first and second components are equal ([6, eq. (11)]) and at least $4=7-3$. The total additive complexity of 2×2-matrix product is at least 15 ([6, Theorem 1]).*

Following our strategy, we impose on tensor $\mathcal{T}$ (9) the constraints

$$T_{11}=e_{1,1}=\left(\begin{smallmatrix}1&0\\0&0\end{smallmatrix}\right),\quad T_{12}=e_{1,2},\quad T_{13}=e_{2,2}\tag{10}$$

and obtain by a Gröbner basis computation [13] that such tensors are the images of Strassen tensor by the action of the following isotropies:

$$\mathrm{w}=\left(\begin{smallmatrix}1&0\\0&1\end{smallmatrix}\right)\times\left(\begin{smallmatrix}1&-1\\0&-1\end{smallmatrix}\right)\times\left(\begin{smallmatrix}w_{11}&w_{12}\\w_{21}&w_{22}\end{smallmatrix}\right).\tag{11}$$

The variant of the Winograd tensor [22] presented with a renumbering as Algorithm 1 is obtained by the action of w with the specialization $w_{12}=w_{21}=1=-w_{11}, w_{22}=0$ on the Strassen tensor $\mathcal{S}$. While the original Strassen algorithm requires 18 additions, only 15 additions are necessary in the Winograd Algorithm 1.

Algorithm 1 : $C=\mathrm{W}(A,B)$

Input: $A=\left(\begin{smallmatrix}a_{11}&a_{12}\\a_{21}&a_{22}\end{smallmatrix}\right)$ and $B=\left(\begin{smallmatrix}b_{11}&b_{12}\\b_{21}&b_{22}\end{smallmatrix}\right)$;
Output: $C=A\cdot B$

$s_1\leftarrow a_{11}-a_{21}$, $s_2\leftarrow a_{21}+a_{22}$, $s_3\leftarrow s_2-a_{11}$, $s_4\leftarrow a_{12}-s_3$,
$t_1\leftarrow b_{22}-b_{12}$, $t_2\leftarrow b_{12}-b_{11}$, $t_3\leftarrow b_{11}+t_1$, $t_4\leftarrow b_{21}-t_3$.
$p_1\leftarrow a_{11}\cdot b_{11}$, $p_2\leftarrow a_{12}\cdot b_{21}$, $p_3\leftarrow a_{22}\cdot t_4$, $p_4\leftarrow s_1\cdot t_1$,
$p_5\leftarrow s_3\cdot t_3$, $p_6\leftarrow s_4\cdot b_{22}$, $p_7\leftarrow s_2\cdot t_2$.
$c_1\leftarrow p_1+p_5$, $c_2\leftarrow c_1+p_4$, $c_3\leftarrow p_1+p_2$, $c_4\leftarrow c_2+p_3$,
$c_5\leftarrow c_2+p_7$, $c_6\leftarrow c_1+p_7$, $c_7\leftarrow c_6+p_6$.
return $C=\left(\begin{smallmatrix}c_3&c_7\\c_4&c_5\end{smallmatrix}\right)$.

As a second example illustrating our strategy, we consider now the matrix squaring that was already explored by Bodrato in [3].

Example 2. *When computing A^2, the contraction (5) of the tensor $\mathcal{T}$ (9) with $A\otimes A$ shows that choosing a subset J of $\{1,\dots,7\}$ and imposing $T_{i1}=T_{i2}$ as constraints with i in J (see [3, eq 4]) can save $|J|$ operations and thus reduce the computational complexity.*

The definition (9) of $\mathcal{T}$, these constraints, and the fact that U,V and W's determinant are 1, form a system with $3+4|J|$ equations and 12 unknowns whose solutions define matrix squaring algorithms.

The algorithm [3, § 2.2, eq 2] is given by the action of the isotropy:

$$\mathrm{g}=\left(\begin{smallmatrix}0&1\\-1&0\end{smallmatrix}\right)\times\left(\begin{smallmatrix}1&1\\0&1\end{smallmatrix}\right)\times\left(\begin{smallmatrix}1&0\\1&1\end{smallmatrix}\right)\tag{12}$$

on Strassen's tensor and is just Chatelin's algorithm [7, Appendix A], with $\lambda = 1$ (published 25 years before [3], but not applied to squaring).

Remark 2.2. *Using symmetries in our strategy reduces the computational cost compared to the resolution of Brent's equations [4, § 5, eq 5.03] with an undetermined tensor $\mathcal{T}$. In the previous example by doing so, we should have constructed a system of at most 64 algebraic equations with $4(3\,(7-|J|)+2\,|J|)$ unknowns, resulting from the constraints on $\mathcal{T}$ and the relation $\mathcal{T} = \mathcal{S}$, expressed using Kronecker product as a single zero matrix in $\mathbb{K}^{8\times 8}$.*

We apply now our strategy on the 2×2 matrix product $A\cdot A^\intercal$.

2.2 2×2-matrix product by its transpose

Applying our Strategy 2.1, we consider (9) a generic matrix multiplication tensor $\mathcal{T}$ and our goal is to reduce the computational complexity of the partial contraction (5) with $A\otimes A^\intercal$ computing $A\cdot A^\intercal$.

By the properties of the transpose operator and the trace, the following relations hold:

$$\begin{aligned}\left\langle T_{i2}, A^\intercal\right\rangle &= \mathrm{Trace}\big(T_{i2}{}^\intercal\cdot A^\intercal\big) = \mathrm{Trace}\big((A\cdot T_{i2})^\intercal\big),\\ &= \mathrm{Trace}\big(A\cdot T_{i2}\big) = \mathrm{Trace}\big(T_{i2}\cdot A\big) = \left\langle T_{i2}{}^\intercal, A\right\rangle.\end{aligned}\tag{13}$$

Thus, the partial contraction (5) satisfies here the following relation:

$$\sum_{i=1}^{7}\left\langle T_{i1},A\right\rangle\left\langle T_{i2},A^\intercal\right\rangle T_{i3} = \sum_{i=1}^{7}\left\langle T_{i1},A\right\rangle\left\langle T_{i2}{}^\intercal,A\right\rangle T_{i3}.\tag{14}$$

2.2.1 Supplementary symmetry constraints. Our goal is to save computations in the evaluation of (14). To do so, we consider the subsets J of $\{1,\dots,7\}$ and H of $\left\{(i,j)\in\{2,\dots,7\}^2 | i\neq j, i\notin J, j\notin J\right\}$ in order to express the following constraints:

$$T_{i1} = T_{i2}{}^\intercal,\ i\in J,\quad T_{j1} = T_{k2}{}^\intercal,\ T_{k1} = T_{j2}{}^\intercal,\ (j,k)\in H.\tag{15}$$

The constraints of type J allow one to save preliminary additions when applying the method to matrices $B = A^\intercal$: since then operations on A and $A^\intercal$ will be the same. The constraints of type H allow to save multiplications especially when dealing with a block-matrix product: in fact, if some matrix products are transpose of another, only one of the pair needs to be computed as shown in Section 3.

We are thus looking for the largest possible sets J and H. By exhaustive search, we conclude that the cardinality of H is at most 2 and then the cardinality of J is at most 3. For example, choosing the sets $J=\{1,2,5\}$ and $H=\{(3,6),(4,7)\}$ we obtain for these solutions the following parameterization expressed with a primitive element $z = v_{11}-v_{21}$:

$$\begin{aligned}
v_{11} &= z+v_{21},\\
v_{22} &= \big(2\,v_{21}(v_{21}+z)-1\big)v_{21}+z^3,\\
v_{12} &= -\big({v_{21}}^2+(v_{21}+z^2)^2+1\big)v_{21}-z,\\
u_{11} &= -\big((z+v_{21})^2+{v_{21}}^2\big)(w_{21}+w_{22}),\\
u_{21} &= -\big((z+v_{21})^2+{v_{21}}^2\big)(w_{11}+w_{12}),\\
u_{12} &= -\big((z+v_{21})^2+{v_{21}}^2\big)w_{22},\\
u_{22} &= \big((z+v_{21})^2+{v_{21}}^2\big)w_{12},\\
&\big((z+v_{21})^2+{v_{21}}^2\big)^2+1=0,\ w_{11}w_{22}-w_{12}w_{21}=1.
\end{aligned}\tag{16}$$

Remark 2.3. *As $\big((z+v_{21})^2+{v_{21}}^2\big)^2+1=0$ occurs in this parameterization, field extension could not be avoided in these algorithms if the field does not have—at least—a square root of -1. We show in Section 3 that we can avoid these extensions with block-matrix products and use our algorithm directly in any field of prime characteristic.*

2.2.2 Supplementary constraint on the number of additions. As done in Example 1, we could also try to reduce the additive complexity and use 4 pre-additions on A (resp. B) [6, Lemma 9] and 7 post-additions on the products to form C [6, Lemma 2]. In the current situation, if the operations on B are exactly the transpose of that of A, then we have the following lower bound:

Lemma 2.1. *Over a non-commutative domain, 11 additive operations are necessary to multiply a 2×2 matrix by its transpose with a bilinear algorithm that uses 7 multiplications.*

Indeed, over a commutative domain, the lower left and upper right parts of the product are transpose of one another and one can save also multiplications. Differently, over non-commutative domains, $A\cdot A^\intercal$ is not symmetric in general (say $ac+bd\neq ca+db$) and all four coefficients need to be computed. But one can still save 4 additions, since there are algorithms where pre-additions are the same on A and $A^\intercal$. Now, to reach that minimum, the constraints (15) must be combined with the minimal number 4 of pre-additions for A. Those can be attained only if 3 of the T_{i1} factors do not require any addition [6, Lemma 8]. Hence, those factors involve only one of the four elements of A and they are just permutations of e_{11}. We thus add these constraints to the system for a subset K of $\{1,\dots,7\}$:

$$|K|=3 \text{ and } T_{i1} \text{ is in } \left\{\left(\begin{smallmatrix}1&0\\0&0\end{smallmatrix}\right),\left(\begin{smallmatrix}0&1\\0&0\end{smallmatrix}\right),\left(\begin{smallmatrix}0&0\\1&0\end{smallmatrix}\right),\left(\begin{smallmatrix}0&0\\0&1\end{smallmatrix}\right)\right\} \text{ and } i \text{ in } K.\tag{17}$$

2.2.3 Selected solution. We choose $K=\{1,2,3\}$ similar to (10) and obtain the following isotropy that sends Strassen tensor to an algorithm computing the symmetric product more efficiently:

$$\mathsf{a} = \left(\begin{smallmatrix}z^2&0\\0&z^2\end{smallmatrix}\right)\times\left(\begin{smallmatrix}z&-z\\0&z^3\end{smallmatrix}\right)\times\left(\begin{smallmatrix}-1&1\\1&0\end{smallmatrix}\right),\quad z^4=-1.\tag{18}$$

We remark that a is equal to $\mathsf{d}\circ\mathsf{w}$ with w the isotropy (11) that sends Strassen tensor to Winograd tensor and with:

$$\mathsf{d} = D_1\otimes D_2\otimes D_3 = \left(\begin{smallmatrix}z^2&0\\0&z^2\end{smallmatrix}\right)\times\left(\begin{smallmatrix}z&0\\0&-z^3\end{smallmatrix}\right)\times\left(\begin{smallmatrix}1&0\\0&1\end{smallmatrix}\right),\ z^4=-1.\tag{19}$$

Hence, the induced algorithm can benefit from the scheduling and additive complexity of the classical Winograd algorithm. In fact, our choice $\mathsf{a}\diamond\mathcal{S}$ is equal to $(\mathsf{d}\circ\mathsf{w})\diamond\mathcal{S}$ and thus, according to remark (2.1) the resulting algorithm expressed as the total contraction

$$\{(\mathsf{d}\circ\mathsf{w})\diamond\mathcal{S},(A\otimes A^\intercal\otimes C)\} = \{\mathsf{w}\diamond\mathcal{S},d^\intercal\diamond(A\otimes A^\intercal\otimes C)\}\tag{20}$$

could be written as a slight modification of Algorithm 1 inputs.

Precisely, as d's components are diagonal, the relation $\mathsf{d}^\intercal=\mathsf{d}$ holds; hence, we could express input modification as:

$$\Big(D_1{}^{-1}\cdot A\cdot D_2\Big)\otimes\Big(D_2{}^{-1}\cdot A^\intercal\cdot D_3\Big)\otimes\Big(D_3{}^{-1}\cdot C\cdot D_1\Big).\tag{21}$$

The above expression is trilinear and the matrices D_i are scalings of the identity for i in $\{1,3\}$, hence our modifications are just:

$$\left(\frac{1}{z^2}A\cdot D_2\right)\otimes\Big(D_2{}^{-1}\cdot A^\intercal\Big)\otimes z^2C.\tag{22}$$

Using notations of Algorithm 1, this is $C=\mathrm{W}\big(A\cdot D_2, D_2{}^{-1}\cdot A^\intercal\big)$.

Allowing our isotropies to have determinant different from 1, we rescale D_2 by a factor $1/z$ to avoid useless 4th root as follows:

$$Q=\frac{D_2}{z}=\left(\begin{smallmatrix}1&0\\0&-z^2\end{smallmatrix}\right)=\left(\begin{smallmatrix}1&0\\0&-y\end{smallmatrix}\right),\quad z^4=-1\tag{23}$$

where y designates the expression z^2 that is a root of -1. Hence, our algorithm to compute the symmetric product is:

$$C = \mathrm{W}\left(A \cdot \frac{D_2}{z}, \left(\frac{D_2}{z}\right)^{-1} \cdot A^{\mathsf{T}}\right) = \mathrm{W}\left(A \cdot Q, \left(A \cdot (Q^{-1})^{\mathsf{T}}\right)^{\mathsf{T}}\right). \quad (24)$$

In the next sections, we describe and extend this algorithm to higher-dimensional symmetric products $A \cdot A^{\mathsf{T}}$ with a $2^\ell m \times 2^\ell m$ matrix A.

3 FAST 2×2-BLOCK RECURSIVE SYRK

The algorithm presented in the previous section is non-commutative and thus we can extend it to higher-dimensional matrix product by a divide and conquer approach. To do so, we use in the sequel upper case letters for coefficients in our algorithms instead of lower case previously (since these coefficients now represent matrices). Thus, new properties and results are induced by this shift of perspective. For example, the coefficient Y introduced in (23) could now be transposed in (24); that leads to the following definition:

Definition 3.1. *An invertible matrix is* skew-orthogonal *if the following relation $Y^{\mathsf{T}} = -Y^{-1}$ holds.*

If Y is skew-orthogonal, then of the 7 recursive matrix products involved in expression (24): 1 can be avoided (P_6) since we do not need the upper right coefficient anymore, 1 can be avoided since it is the transposition of another product ($P_7 = P_4{}^{\mathsf{T}}$) and 3 are recursive calls to SYRK. This results in Algorithm 2.

Algorithm 2 syrk: symmetric matrix product

Input: $A = \left(\begin{smallmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{smallmatrix}\right)$; a skew-orthogonal matrix Y.

Output: The lower left triangular part of $C = A \cdot A^{\mathsf{T}} = \left(\begin{smallmatrix} C_{11} & C_{21}{}^{\mathsf{T}} \\ C_{21} & C_{22} \end{smallmatrix}\right)$.

▷ *4 additions and 2 multiplications by Y:*

$S_1 \leftarrow (A_{21} - A_{11}) \cdot Y$, $S_2 \leftarrow A_{22} - A_{21} \cdot Y$,
$S_3 \leftarrow S_1 - A_{22}$, $S_4 \leftarrow S_3 + A_{12}$.

▷ *3 recursive SYRK (P_1, P_2, P_5) and 2 generic (P_3, P_4) products:*

$P_1 \leftarrow A_{11} \cdot A_{11}{}^{\mathsf{T}}$, $P_2 \leftarrow A_{12} \cdot A_{12}{}^{\mathsf{T}}$,
$P_3 \leftarrow A_{22} \cdot S_4{}^{\mathsf{T}}$, $P_4 \leftarrow S_1 \cdot S_2{}^{\mathsf{T}}$, $P_5 \leftarrow S_3 \cdot S_3{}^{\mathsf{T}}$.

▷ *2 symmetric additions (half additions):*

$\mathrm{Low}(U_1) \leftarrow \mathrm{Low}(P_1) + \mathrm{Low}(P_5)$, ▷ *$U_1, P_1, P_5$ are symm.*
$\mathrm{Low}(U_3) \leftarrow \mathrm{Low}(P_1) + \mathrm{Low}(P_2)$, ▷ *$U_3, P_1, P_2$ are symm.*

▷ *2 complete additions (P_4 and P_3 are not symmetric):*

$\mathrm{Up}(U_1) \leftarrow \mathrm{Low}(U_1)^{\mathsf{T}}$, $U_2 \leftarrow U_1 + P_4$, $U_4 \leftarrow U_2 + P_3$,

▷ *1 half addition ($U_5 = U_1 + P_4 + P_4{}^{\mathsf{T}}$ is symmetric):*

$\mathrm{Low}(U_5) \leftarrow \mathrm{Low}(U_2) + \mathrm{Low}(P_4{}^{\mathsf{T}})$.

return $\left(\begin{smallmatrix} \mathrm{Low}(U_3) & \\ U_4 & \mathrm{Low}(U_5) \end{smallmatrix}\right)$.

3.1 Skew orthogonal matrices

Algorithm 2 requires a skew-orthogonal matrix. Unfortunately there are no skew-orthogonal matrices over $\mathbb{R}$, nor $\mathbb{Q}$. Hence, we report no improvement in these cases. In other domains, the simplest skew-orthogonal matrices just use a square root of -1.

3.1.1 Over the complex field. Therefore Algorithm 2 is directly usable over $\mathbb{C}^{n\times n}$ with $Y = i\,\mathrm{I}_n \in \mathbb{C}^{n\times n}$. Further, usually, complex numbers are emulated by a pair of floats so then the multiplications by $Y = i\,\mathrm{I}_n$ are essentially free since they just exchange the real and imaginary parts, with one sign flipping. Even though over the complex the product ZHERK of a matrix by its *conjugate* transpose is more widely used, ZSYRK has some applications, see for instance [1].

3.1.2 Negative one is a square. Over some fields with prime characteristic, square roots of -1 can be elements of the base field, denoted i in $\mathbb{F}$ again. There, Algorithm 2 only requires some pre-multiplications by this square root (with also $Y = i\,\mathrm{I}_n \in \mathbb{F}^{n\times n}$), but *within the field*. Proposition 3.1 thereafter characterizes these fields.

Proposition 3.1. *Fields with characteristic two, or with an odd characteristic $p \equiv 1 \bmod 4$, or finite fields that are an even extension, contain a square root of -1.*

Proof. If $p = 2$, then $1 = 1^2 = -1$. If $p \equiv 1 \bmod 4$, then half of the non-zero elements x in the base field of size p satisfy $x^{\frac{p-1}{4}} \neq \pm 1$ and then the square of the latter must be -1. If the finite field $\mathbb{F}$ is of cardinality p^{2k}, then, similarly, there exists elements $x^{\frac{p^k-1}{2}}\frac{p^k+1}{2}$ different from ± 1 and then the square of the latter must be -1. □

3.1.3 Any field with prime characteristic. Finally, we show that Algorithm 2 can also be run without any field extension, even when -1 is not a square: form the skew-orthogonal matrices constructed in Proposition 3.2, thereafter, and use them directly as long as the dimension of Y is even. Whenever this dimension is odd, it is always possible to pad with zeroes so that $A \cdot A^{\mathsf{T}} = \left(\begin{smallmatrix} A & 0 \end{smallmatrix}\right) \cdot \left(\begin{smallmatrix} A^{\mathsf{T}} \\ 0 \end{smallmatrix}\right)$.

Proposition 3.2. *Let $\mathbb{F}$ be a field of characteristic p, there exists (a, b) in $\mathbb{F}^2$ such that the matrix:*

$$\begin{pmatrix} a & b \\ -b & a \end{pmatrix} \otimes \mathrm{I}_n = \begin{pmatrix} a\,\mathrm{I}_n & b\,\mathrm{I}_n \\ -b\,\mathrm{I}_n & a\,\mathrm{I}_n \end{pmatrix} \quad \text{in} \quad \mathbb{F}^{2n\times 2n} \quad (25)$$

is skew-orthogonal.

Proof. Using the relation

$$\begin{pmatrix} a\,\mathrm{I}_n & b\,\mathrm{I}_n \\ -b\,\mathrm{I}_n & a\,\mathrm{I}_n \end{pmatrix} \begin{pmatrix} a\,\mathrm{I}_n & b\,\mathrm{I}_n \\ -b\,\mathrm{I}_n & a\,\mathrm{I}_n \end{pmatrix}^{\mathsf{T}} = (a^2 + b^2)\,\mathrm{I}_{2n}, \quad (26)$$

it suffices to prove that there exist a, b such that $a^2 + b^2 = -1$. In characteristic 2, $a = 1, b = 0$ is a solution as $1^2 + 0^2 = -1$. In odd characteristic, there are $\frac{p+1}{2}$ distinct square elements ${x_i}^2$ in the base prime field. Therefore, there are $\frac{p+1}{2}$ distinct elements $-1 - {x_i}^2$. But there are only p distinct elements in the base field, thus there exists a couple (i, j) such that $-1 - {x_i}^2 = {x_j}^2$ [19, Lemma 6]. □

Proposition 3.2 shows that skew-orthogonal matrices do exist for any field with prime characteristic. For Algorithm 2, we need to build them mostly for $p \equiv 3 \bmod 4$ (otherwise use Proposition 3.1).

For this, without the extended Riemann hypothesis (ERH), it is possible to use the decomposition of primes into squares:

(1) Compute first a prime $r = 4pk + (3-1)p - 1$, then the relations $r \equiv 1 \bmod 4$ and $r \equiv -1 \bmod p$ hold;
(2) Thus, results of [5] allow one to decompose primes into squares and give a couple (a, b) in $\mathbb{Z}^2$ such that $a^2 + b^2 = r$. Finally, we get $a^2 + b^2 \equiv -1 \bmod p$.

By the prime number theorem the first step is polynomial in $\log(p)$, as is the second step (square root modulo a prime, denoted `sqrt`, has a cost close to exponentiation and then the rest of Brillhart's

algorithm is GCD-like). In practice, though, it is faster to use the following Algorithm 3, even though the latter has a better asymptotic complexity bound only if the ERH is true.

Algorithm 3 SoS: Sum of squares decomposition over a finite field

Input: $p \in \mathbb{P}\backslash\{2\}, k \in \mathbb{Z}$.
Output: $(a, b) \in \mathbb{Z}^2$, s.t. $a^2 + b^2 \equiv k \bmod p$.

1: **if** $\left(\frac{k}{p}\right) = 1$ **then** ▷ *k is a square mod p*
2: **return** $(\texttt{sqrt}(k), 0)$.
3: **else** ▷ *Find smallest quadratic non-residue*
4: $s \leftarrow 2$; **while** $\left(\frac{s}{p}\right) == 1$ **do** $s \leftarrow s + 1$
5: $c \leftarrow \texttt{sqrt}(s-1)$ ▷ *$s-1$ must be a square*
6: $r \leftarrow ks^{-1} \bmod p$
7: $a \leftarrow \texttt{sqrt}(r)$ ▷ *Now $k \equiv a^2 s \equiv a^2(1+c^2) \bmod p$*
8: **return** $(a, ac \bmod p)$

Proposition 3.3. *Algorithm 3 is correct and, under the ERH, runs in expected time $\widetilde{O}(\log^3(p))$.*

Proof. If k is square then the square of one of its square roots added to the square of zero is a solution. Otherwise, the lowest quadratic non-residue (LQNR) modulo p is one plus a square b^2 (1 is always a square so the LQNR is larger than 2). For any generator of $\mathbb{Z}_p$, quadratic non-residues, as well as their inverses (s is invertible as it is non-zero and p is prime), have an odd discrete logarithm. Therefore the multiplication of k and the inverse of the LQNR must be a square a^2. This means that the relation $k = a^2(1 + b^2) = a^2 + (ab)^2$ holds. Now for the running time, under ERH, the LQNR should be lower than $3\log^2(p)/2 - 44\log(p)/5 + 13$ [21, Theorem 6.35]. The expected number of Legendre symbol computations is $O(\log^2(p))$ and this dominates the modular square root computations. □

Remark 3.1. *Another possibility is to use randomization: instead of using the lowest quadratic non-residue (LQNR), randomly select a non-residue s, and then decrement it until $s-1$ is a quadratic residue (1 is a square so this will terminate)*[1]*. Also, when computing t sum of squares modulo the same prime, one can compute the LQNR* only once *to get all the sum of squares with an expected cost bounded by $\widetilde{O}(\log^3(p) + t\log^2(p))$.*

Remark 3.2. *Except in characteristic 2 or in algebraic closures, where every element is a square anyway, Algorithm 3 is easily extended over any finite field: compute the LQNR in the base prime field, then use Tonelli-Shanks or Cipolla-Lehmer algorithm to compute square roots in the extension field.*

Denote by $\texttt{SoS}(q, k)$ this algorithm decomposing k as a sum of squares within any finite field $\mathbb{F}_q$. This is not always possible over infinite fields, but there Algorithm 3 still works anyway for the special case $k = -1$: just run it in the prime subfield.

[1] In practice, the running time seems very close to that of Algorithm 3 anyway, see, e.g. the implementation in Givaro rev. 7bdefe6, https://github.com/linbox-team/givaro.

3.2 Conjugate transpose

Note that Algorithm 2 remains valid if transposition is replaced by *conjugate transposition*, provided that there exists a matrix Y such that $Y \cdot \overline{Y}^{\mathsf{T}} = -\mathrm{I}$. This is not possible anymore over the complex field, but works for any even extension field, thanks to Algorithm 3: if -1 is a square in $\mathbb{F}_q$, then $Y = \sqrt{-1} \cdot \mathrm{I}_n$ still works; otherwise there exists a square root i of -1 in $\mathbb{F}_{q^2}$, from Proposition 3.1. In the latter case, thus build (a, b), both in $\mathbb{F}_q$, such that $a^2 + b^2 = -1$. Now $Y = (a + ib) \cdot \mathrm{I}_n$ in $\mathbb{F}_{q^2}^{n\times n}$ is appropriate: indeed, since $q \equiv 3 \bmod 4$, we have that $\overline{a + ib} = (a + ib)^q = a - ib$.

4 ANALYSIS AND IMPLEMENTATION

4.1 Complexity bounds

Theorem 4.1. *Algorithm 2 requires $\frac{2}{2^\omega - 3} C_\omega n^\omega + o(n^\omega)$ field operations, over $\mathbb{C}$ or over any field with prime characteristic.*

Proof. Algorithm 2 is applied recursively to compute three products P_1, P_2 and P_7, while P_4 and P_5 are computed in $\mathrm{MM}_\omega(n) = C_\omega n^\omega + o(n^\omega)$ using a general matrix multiplication algorithm. We will show that applying the skew-orthogonal matrix Y to a $n \times n$ matrix costs yn^2 for some constant y depending on the base field. Then applying Remark 4.1 thereafter, the cost $T(n)$ of Algorithm 2 satisfies:

$$T(n) \le 3T(n/2) + 2C_\omega(n/2)^\omega + (7.5 + 2y)(n/2)^2 + o\left(n^2\right) \quad (27)$$

and $T(4)$ is a constant. Thus, by the master Theorem:

$$T(n) \le \frac{2C_\omega}{2^\omega - 3} n^\omega + o\left(n^\omega\right) = \frac{2}{2^\omega - 3}\mathrm{MM}_\omega(n) + o\left(n^\omega\right). \quad (28)$$

If the field is $\mathbb{C}$ or satisfies the conditions of Proposition 3.1, there is a square root i of -1. Setting $Y = i\,\mathrm{I}_{n/2}$ yields $y = 1$. Otherwise, in characteristic $p \equiv 3 \bmod 4$, Proposition 3.2 produces Y equal to $\left(\begin{smallmatrix} a & b \\ -b & a \end{smallmatrix}\right) \otimes \mathrm{I}_{n/2}$ for which $y = 3$. As a subcase, the latter can be improved when $p \equiv 3 \bmod 8$: then -2 is a square (indeed, $\left(\frac{-2}{p}\right) = \left(\frac{-1}{p}\right)\left(\frac{2}{p}\right) = (-1)^{\frac{p-1}{2}}(-1)^{\frac{p^2-1}{8}} = (-1)(-1) = 1$). Therefore, in this case set $a = 1$ and $b \equiv \sqrt{-2} \bmod p$ such that the relation $a^2 + b^2 = -1$ yields $Y = \left(\begin{smallmatrix} 1 & \sqrt{-2} \\ -\sqrt{-2} & 1 \end{smallmatrix}\right) \otimes \mathrm{I}_{n/2}$ for which $y = 2$. □

To our knowledge, the best previously known result was with a $\frac{2}{2^\omega - 4}$ factor instead, see e.g. [11, § 6.3.1]. Table 1 summarizes the arithmetic complexity bound improvements.

Problem	Alg.	$O(n^3)$	$O\left(n^{\log_2(7)}\right)$	$O(n^\omega)$
$A \cdot A^{\mathsf{T}} \in \mathbb{F}^{n\times n}$	[11]	n^3	$\frac{2}{3}\mathrm{MM}_{\log_2(7)}(n)$	$\frac{2}{2^\omega - 4}\mathrm{MM}_\omega(n)$
	Alg. 2	$0.8n^3$	$\frac{1}{2}\mathrm{MM}_{\log_2(7)}(n)$	$\frac{2}{2^\omega - 3}\mathrm{MM}_\omega(n)$

Table 1: Arithmetic complexity bounds leading terms.

Alternatively, over $\mathbb{C}$, the $3M$ method (Karatsuba) for non-symmetric matrix multiplication reduces the number of multiplications of real matrices from 4 to 3 [15]: if $RR_\omega(n)$ is the cost of multiplying $n \times n$ matrices over $\mathbb{R}$, then the $3M$ method costs $3RR_\omega(n) + o(n^\omega)$ operations over $\mathbb{R}$. Adapting this approach to the symmetric case yields a $2M$ method to compute the product of a complex

matrix by its transpose, using only 2 real products: $H = A \cdot B^\intercal$ and $G = (A+B) \cdot (A^\intercal - B^\intercal)$. Combining those into $(G - H^\intercal + H) + i(H + H^\intercal)$, yields the product $(A+iB) \cdot (A^\intercal + iB^\intercal)$. This approach costs $2RR_\omega + o(n^\omega)$ operations in $\mathbb{R}$.

Classical algorithm [11, § 6.3.1] applies a divide and conquer approach directly on the complex field. This would use only the equivalent of $\frac{2}{2^\omega - 4}$ complex floating point $n \times n$ products. Using the $3M$ method for the complex products, this algorithm uses overall $\frac{6}{2^\omega - 4} RR_\omega + o(n^\omega)$ operations in $\mathbb{R}$. Finally, Algorithm 2 only costs $\frac{2}{2^\omega - 3}$ complex multiplications for a leading term bounded by $\frac{6}{2^\omega - 3}\mathrm{RR}_\omega$, better than $2\mathrm{RR}_\omega$ for $\omega > \log_2(6) \approx 2.585$. This is summarized in Table 2, replacing ω by 3 or $\log_2(7)$.

Problem	Alg.	$\mathrm{MM}_3(n)$	$\mathrm{MM}_{\log_2 7}(n)$	$\mathrm{MM}_\omega(n)$
$A \cdot B \in \mathbb{C}^{n\times n}$	naive	$8n^3$	$4\,\mathrm{RR}_{\log_2(7)}(n)$	$4\,\mathrm{RR}_\omega(n)$
	3M	$6n^3$	$3\,\mathrm{RR}_{\log_2(7)}(n)$	$3\,\mathrm{RR}_\omega(n)$
$A \cdot A^\intercal \in \mathbb{C}^{n\times n}$	2M	$4n^3$	$2\,\mathrm{RR}_{\log_2(7)}(n)$	$2\,\mathrm{RR}_\omega(n)$
	[11]	$3n^3$	$2\,\mathrm{RR}_{\log_2(7)}(n)$	$\frac{6}{2^\omega-4}\,\mathrm{RR}_\omega(n)$
	Alg. 2	$2.4n^3$	$\frac{3}{2}\,\mathrm{RR}_{\log_2(7)}(n)$	$\frac{6}{2^\omega-3}\,\mathrm{RR}_\omega(n)$

Table 2: Symmetric multiplication over $\mathbb{C}$: leading term of the cost in number of operations over $\mathbb{R}$.

Remark 4.1. *Each recursive level of Algorithm 2 is composed of 9 block additions. An exhaustive search on all symmetric algorithms derived from Strassen's showed that this number is minimal in this class of algorithms. Note also that* 3 *out of these* 9 *additions in Algorithm 2 involve symmetric matrices and are therefore only performed on the lower triangular part of the matrix. Overall, the number of scalar additions is* $6n^2 + 3/2n(n+1) = 15/2n^2 + 1.5n$*, nearly half of the optimal in the non-symmetric case [6, Theorem 1].*

To further reduce the number of additions, a promising approach is that undertaken in [2, 16]. This is however not clear to us how to adapt our strategy to their recursive transformation of basis.

4.2 Implementation and scheduling

This section reports on an implementation of Algorithm 2 over prime fields. We propose in Table 3 and Figure 1 a schedule for the operation $C \leftarrow A \cdot A^\intercal$ using no more extra storage than the unused upper triangular part of the result C.

#	operation	loc.	#	operation	loc.
1	$S_1 = (A_{21} - A_{11}) \cdot Y$	C_{21}	9	$U_1 = P_1 + P_5$	C_{12}
2	$S_2 = A_{22} - A_{21} \cdot Y$	C_{12}		$\mathrm{Up}(U_1) = \mathrm{Low}(U_1)^\intercal$	C_{12}
3	$P_4{}^\intercal = S_2 \cdot S_1{}^\intercal$	C_{22}	10	$U_2 = U_1 + P_4$	C_{12}
4	$S_3 = S_1 - A_{22}$	C_{21}	11	$U_4 = U_2 + P_3$	C_{21}
5	$P_5 = S_3 \cdot S_3{}^\intercal$	C_{12}	12	$U_5 = U_2 + P_4{}^\intercal$	C_{22}
6	$S_4 = S_3 + A_{12}$	C_{11}	13	$P_2 = A_{12} \cdot A_{12}{}^\intercal$	C_{12}
7	$P_3 = A_{22} \cdot S_4{}^\intercal$	C_{21}	14	$U_3 = P_1 + P_2$	C_{11}
8	$P_1 = A_{11} \cdot A_{11}{}^\intercal$	C_{11}			

Table 3: Memory placement and schedule of tasks to compute the lower triangular part of $C \leftarrow A \cdot A^\intercal$ when $k \le n$. The block C_{12} of the output matrix is the only temporary used.

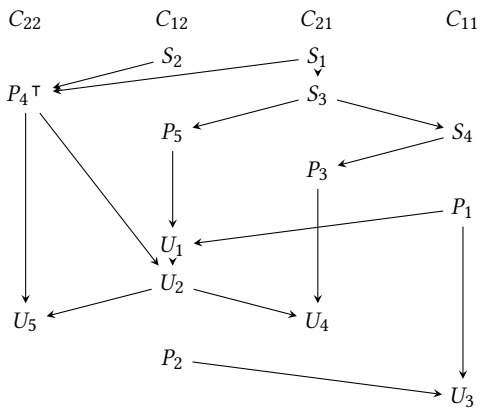


Figure 1: DAG of the tasks and their memory location for the computation of $C \leftarrow A \cdot A^\intercal$ presented in Table 3.

operation	loc.	operation	loc.
$S_1 = (A_{21} - A_{11}) \cdot Y$	tmp	$P_1 = \alpha A_{11} \cdot A_{11}{}^\intercal$	tmp
$S_2 = A_{22} - A_{21} \cdot Y$	C_{12}	$U_1 = P_1 + P_5$	C_{12}
$\mathrm{Up}(C_{11}) = \mathrm{Low}(C_{22})^\intercal$	C_{11}	$\mathrm{Up}(U_1) = \mathrm{Low}(U_1)^\intercal$	C_{12}
$P_4{}^\intercal = \alpha S_2 \cdot S_1{}^\intercal$	C_{22}	$U_2 = U_1 + P_4$	C_{12}
$S_3 = S_1 - A_{22}$	tmp	$U_4 = U_2 + P_3$	C_{21}
$P_5 = \alpha S_3 \cdot S_3{}^\intercal$	C_{12}	$U_5 = U_2 + P_4{}^\intercal + \beta \mathrm{Up}(C_{11})^\intercal$	C_{22}
$S_4 = S_3 + A_{12}$	tmp	$P_2 = \alpha A_{12} \cdot A_{12}{}^\intercal + \beta C_{11}$	C_{11}
$P_3 = \alpha A_{22} \cdot S_4{}^\intercal + \beta C_{21}$	C_{21}	$U_3 = P_1 + P_2$	C_{11}

Table 4: Memory placement and schedule of tasks to compute the lower triangular part of $C \leftarrow \alpha A \cdot A^\intercal + \beta C$ when $k \le n$. The block C_{12} of the output matrix as well as an $n/2 \times n/2$ block tmp are used as temporary storages.

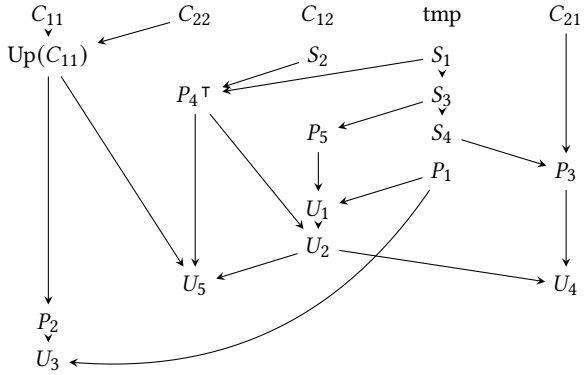


Figure 2: DAG of the tasks and their memory location for the computation of $C \leftarrow \alpha A \cdot A^\intercal + \beta C$ presented in Table 4.

For the more general operation $C \leftarrow \alpha A \cdot A^\intercal + \beta C$, Table 4 and Figure 2 propose a schedule requiring only an additional $n/2 \times n/2$ temporary storage. These algorithms have been implemented as the `fsyrk` routine in the `fflas-ffpack` library for dense linear algebra over a finite field [14, from commit 0a91d61e].

Figure 3 compares the computation speed in effective Gfops (defined as $n^3/(10^9 \times \text{time})$) of this implementation over $\mathbb{Z}/131071\mathbb{Z}$ with that of the double precision BLAS routines `dsyrk`, the classical

cubic-time routine over a finite field (calling `dsyrk` and performing modular reductions on the result), and the classical divide and conquer algorithm [11, § 6.3.1]. The `fflas-ffpack` library is linked

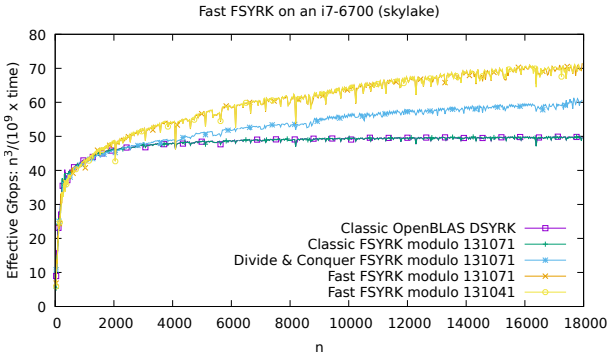


Figure 3: Speed of an implementation of Algorithm 2

with OpenBLAS [23, v0.3.6] and compiled with `gcc-9.2` on an Intel skylake i7-6700 running a Debian GNU/Linux system (v5.2.17).

The slight overhead of performing the modular reductions is quickly compensated by the speed-up of the sub-cubic algorithm (the threshold for a first recursive call is near $n = 2000$). The classical divide and conquer approach also speeds up the classical algorithm, but starting from a larger threshold, and hence at a slower pace. Lastly, the speed is merely identical modulo 131041, where square roots of -1 exist, thus showing the limited overhead of the preconditioning by the matrix Y.

5 SYRK WITH BLOCK DIAGONAL SCALING

Symmetric rank k updates are a key building block for symmetric triangular factorization algorithms, for their efficiency is one of the bottlenecks. In the most general setting (indefinite factorization), a block diagonal scaling by a matrix D, with 1 or 2 dimensional diagonal blocks, has to be inserted within the product, leading to the operation: $C \leftarrow C - A \cdot D \cdot A^\intercal$.

Handling the block diagonal structure over the course of the recursive algorithm may become tedious and quite expensive. For instance, a 2×2 diagonal block might have to be cut by a recursive split. We will see also in the following that non-squares in the diagonal need to be dealt with in pairs. In both cases it might be necessary to add a column to deal with these cases: this is potentially $O(\log_2(n))$ extra columns in a recursive setting.

Over a finite field, though, we will show in this section, how to factor the block-diagonal matrix D into $D = \Delta \cdot \Delta^\intercal$, *without needing any field extension*, and then compute instead $(A \cdot \Delta) \cdot (A \cdot \Delta)^\intercal$. Algorithm 6, deals with non-squares and 2×2 blocks only once beforehand, introducing no more than 2 extra-columns overall. Section 5.1 shows how to factor a diagonal matrix, without resorting to field extensions for non-squares. Then Sections 5.2.1 and 5.2.2 show how to deal with the 2×2 blocks depending on the characteristic.

5.1 Factoring non-squares within a finite field

First we give an algorithm handling pairs of non-quadratic residues.

Proposition 5.1. *Algorithm 4 is correct.*

Algorithm 4 : `nrsyf`: Sym. factorization. of a pair of non-residues

Input: $(\alpha, \beta) \in \mathbb{F}_q{}^2$, both being quadratic non-residues.
Output: $Y \in \mathbb{F}_q{}^{2\times 2}$, s.t. $Y \cdot Y^\intercal = \begin{pmatrix} \alpha & 0 \\ 0 & \beta \end{pmatrix}$.

1: $(a, b) \leftarrow \mathsf{SoS}(q, \alpha)$; ▷ $\alpha = a^2 + b^2$
2: $d \leftarrow a\,\mathtt{sqrt}(\beta\alpha^{-1})$; ▷ $d^2 = a^2\beta\alpha^{-1}$
3: $c \leftarrow -bda^{-1}$; ▷ $ac + bd = 0$
4: **return** $Y \leftarrow \begin{pmatrix} a & b \\ c & d \end{pmatrix}$.

Proof. Given α and β quadratic non-residues, the couple (a, b), such that $\alpha = a^2 + b^2$, is found by the algorithm of Remark 3.2. Second, as α and β are quadratic non-residues, over a finite field their quotient is a residue since: $\left(\beta\alpha^{-1}\right)^{\frac{q-1}{2}} = \frac{-1}{-1} = 1$. Third, if c denotes $-bda^{-1}$ then $c^2 + d^2$ is equal to $(-bd/a)^2 + d^2$ and thus to $(b^2/a^2 + 1)d^2$; this last quantity is equal to $(\alpha)d^2/a^2$ and then to $\alpha(a\sqrt{\beta/\alpha})^2/a^2 = \alpha(a^2\beta/\alpha)/a^2 = \beta$. Fourth, a (or w.l.o.g. b) is invertible. Indeed, α is not a square, therefore it is non-zero and thus one of a or b must be non-zero. Finally, we obtain the cancellation $ac + bd = a(-dba^{-1}) + bd = -db + bd = 0$ and the matrix product $Y \cdot Y^\intercal$ is $\begin{pmatrix} a & b \\ c & d \end{pmatrix}\begin{pmatrix} a & c \\ b & d \end{pmatrix} = \begin{pmatrix} a^2+b^2 & ac+bd \\ ac+bd & c^2+d^2 \end{pmatrix} = \begin{pmatrix} \alpha & 0 \\ 0 & \beta \end{pmatrix}$. □

Using Algorithm 4, one can then factor any diagonal matrix within a finite field as a symmetric product with a tridiagonal matrix. This can then be used to compute efficiently $A \cdot D \cdot A^\intercal$ with D a diagonal matrix: factor D with a tridiagonal matrix $D = \Delta \cdot \Delta^\intercal$, then pre-multiply A by this tridiagonal matrix and run a fast symmetric product on the resulting matrix. This is shown in Algorithm 5, where the overhead, compared to simple matrix multiplication, is only $O(n^2)$ (that is $O(n)$ square roots and $O(n)$ column scalings).

Algorithm 5 `syrkd`: sym. matrix product with diagonal scaling

Input: $A \in \mathbb{F}_q{}^{m\times n}$ and $D = \mathrm{Diag}(d_1, \ldots, d_n) \in \mathbb{F}_q{}^{n\times n}$
Output: $A \cdot D \cdot A^\intercal$ in $\mathbb{F}_q{}^{m\times m}$

1: **if** number of quadratic non-residues in $\{d_1, \ldots, d_n\}$ is odd **then**
  Let d_ℓ be one of the quadratic non-residues
2: $\bar{D} \leftarrow \mathrm{Diag}(d_1, \ldots, d_n, d_\ell) \in \mathbb{F}_q{}^{(n+1)\times(n+1)}$
3: $\bar{A} \leftarrow (\,A\ 0\,) \in \mathbb{F}_q{}^{m\times(n+1)}$ ▷ *Augment A with a zero column*
4: **else**
5: $\bar{D} \leftarrow \mathrm{Diag}(d_1, \ldots, d_n) \in \mathbb{F}_q{}^{n\times n}$
6: $\bar{A} \leftarrow A \in \mathbb{F}_q{}^{m\times n}$
7: **for all** quadratic residues d_j in $\bar{D}$ **do**
8: $\bar{A}_{*,j} \leftarrow \mathtt{sqrt}(d_j) \cdot \bar{A}_{*,j}$ ▷ *Scale col. j of $\bar{A}$ by a sq. root of d_j*
9: **for all** distinct pairs of quadratic non-residues (d_i, d_j) in $\bar{D}$ **do**
10: $\Delta \leftarrow \mathtt{nrsyf}(d_i, d_j)$ ▷ $\Delta \cdot \Delta^\intercal = \begin{pmatrix} d_i & 0 \\ 0 & d_j \end{pmatrix}$ *using Algorithm 4*
11: $(\,\bar{A}_{*,i}\ \bar{A}_{*,j}\,) \leftarrow (\,\bar{A}_{*,i}\ \bar{A}_{*,j}\,) \cdot \Delta$;
12: **return** $\mathtt{syrk}(\bar{A})$ ▷ $\bar{A} \cdot \bar{A}^\intercal$ *using Algorithm 2*

5.2 Antidiagonal and antitriangular blocks

In general, an $L \cdot D \cdot L^\intercal$ factorization may have antitriangular or antidiagonal blocks in D [12]. In order to reduce to a routine for fast symmetric multiplication with diagonal scaling, these blocks need to be processed once for all, which is what this section is about.

5.2.1 Antidiagonal blocks in odd characteristic. In odd characteristic, the 2-dimensional blocks in an $L \cdot D \cdot L^\mathsf{T}$ factorization are only of the form $\left(\begin{smallmatrix} 0 & \beta \\ \beta & 0 \end{smallmatrix}\right)$, and always have the symmetric factorization:

$$\left(\begin{smallmatrix} 1 & 1 \\ 1 & -1 \end{smallmatrix}\right) \left(\begin{smallmatrix} \frac{1}{2}\beta & 0 \\ 0 & -\frac{1}{2}\beta \end{smallmatrix}\right) \left(\begin{smallmatrix} 1 & 1 \\ 1 & -1 \end{smallmatrix}\right)^\mathsf{T} = \left(\begin{smallmatrix} 0 & \beta \\ \beta & 0 \end{smallmatrix}\right). \tag{29}$$

This shows the reduction to the diagonal case (note the requirement that 2 is invertible).

5.2.2 Antitriangular blocks in characteristic 2. In characteristic 2, some 2×2 blocks might not be reduced further than an antitriangular form: $\left(\begin{smallmatrix} 0 & \beta \\ \beta & \gamma \end{smallmatrix}\right)$, with $\gamma \neq 0$.

In characteristic 2 every element is a square, therefore those antitriangular blocks can be factored as shown in Eq. (30):

$$\left(\begin{smallmatrix} 0 & \beta \\ \beta & \gamma \end{smallmatrix}\right) = \left(\left(\begin{smallmatrix} \beta\gamma^{-1/2} & 0 \\ 0 & \gamma^{1/2} \end{smallmatrix}\right) \left(\begin{smallmatrix} 1 & 1 \\ 1 & 0 \end{smallmatrix}\right)\right) \left(\left(\begin{smallmatrix} \beta\gamma^{-1/2} & 0 \\ 0 & \gamma^{1/2} \end{smallmatrix}\right) \left(\begin{smallmatrix} 1 & 1 \\ 1 & 0 \end{smallmatrix}\right)\right)^\mathsf{T}. \tag{30}$$

Therefore the antitriangular blocks also reduce to the diagonal case.

5.2.3 Antidiagonal blocks in characteristic 2. The symmetric factorization in this case might require an extra row or column [18] as shown in Eq. (31):

$$\left(\begin{smallmatrix} 1 & 0 \\ 0 & \beta \end{smallmatrix}\right) \left(\begin{smallmatrix} 1 & 0 & 1 \\ 0 & 1 & 1 \end{smallmatrix}\right) \left(\left(\begin{smallmatrix} 1 & 0 \\ 0 & \beta \end{smallmatrix}\right) \left(\begin{smallmatrix} 1 & 0 & 1 \\ 0 & 1 & 1 \end{smallmatrix}\right)\right)^\mathsf{T} = \left(\begin{smallmatrix} 0 & \beta \\ \beta & 0 \end{smallmatrix}\right) \bmod 2. \tag{31}$$

A first option is to augment A by one column for each antidiagonal block, by applying the 2×3 factor in Eq. (31). However one can instead combine a diagonal element, say x, and an antidiagonal block as shown in Eq. (32).

$$\left(\begin{smallmatrix} \sqrt{x} & \sqrt{x} & \sqrt{x} \\ 1 & 0 & 1 \\ 0 & \beta & \beta \end{smallmatrix}\right) \left(\begin{smallmatrix} \sqrt{x} & \sqrt{x} & \sqrt{x} \\ 1 & 0 & 1 \\ 0 & \beta & \beta \end{smallmatrix}\right)^\mathsf{T} = \left(\begin{smallmatrix} x & 0 & 0 \\ 0 & 0 & \beta \\ 0 & \beta & 0 \end{smallmatrix}\right) \bmod 2. \tag{32}$$

Hence, any antidiagonal block can be combined with any 1×1 block to form a symmetric factorization.

There remains the case when there are no 1×1 blocks. Then, one can use Eq. (31) once, on the first antidiagonal block, and add column to A. This indeed extracts the antidiagonal elements and creates a 3×3 identity block in the middle. Any one of its three ones can then be used as x in a further combination with the next antidiagonal blocks. Algorithm 6 sums up the use of Eqs. (29) to (32).

Algorithm 6 : `syrkbd`: sym. matrix product with block diag. scaling

Input: $A \in \mathbb{F}_q^{m \times n}$; $B \in \mathbb{F}_q^{n \times n}$, block diagonal with scalar or 2-dimensional blocks of the form $\left(\begin{smallmatrix} 0 & \beta \\ \beta & \gamma \end{smallmatrix}\right)$ with $\beta \neq 0$

Output: $A \cdot B \cdot A^\mathsf{T} \in \mathbb{F}_q^{m \times m}$

$\bar{A} \leftarrow A \in \mathbb{F}_q^{m \times n}$; $\bar{D} \leftarrow \mathrm{I}_n$
for all scalar blocks in B at position j **do** $\bar{D}_j \leftarrow B_{j,j}$
if q is odd **then** ▷ *Use Eq. (29)*
  for all symmetric antidiagonal blocks in B at $(j, j+1)$ **do**
    $\beta \leftarrow B_{j,j+1} (= B_{j+1,j})$
    $\bar{D}_j \leftarrow \frac{1}{2}\beta$; $\bar{D}_{j+1} \leftarrow -\frac{1}{2}\beta$
    $\left(\begin{smallmatrix} \bar{A}_{*,i} & \bar{A}_{*,j} \end{smallmatrix}\right) \leftarrow \left(\begin{smallmatrix} \bar{A}_{*,i} & \bar{A}_{*,j} \end{smallmatrix}\right) \left(\begin{smallmatrix} 1 & 1 \\ 1 & -1 \end{smallmatrix}\right)$
else
  for all antitriangular blocks in B at position $(j, j+1)$ **do**
    $\beta \leftarrow B_{j,j+1} (= B_{j+1,j})$; $\delta \leftarrow \mathtt{sqrt}(B_{j+1,j+1})$;
    $\bar{A}_{*,j} \leftarrow \beta\delta^{-1} \cdot \bar{A}_{*,j}$ ▷ *Scale column j of $\bar{A}$*
    $\bar{A}_{*,j+1} \leftarrow \delta \cdot \bar{A}_{*,j+1}$ ▷ *Scale column $j+1$ of $\bar{A}$*
    $\bar{A}_{*,j+1} \leftarrow \bar{A}_{*,j+1} + \bar{A}_{*,j}$ ▷ *Use Eq. (30)*
    Swap columns j and $j+1$ of $\bar{A}$
  if there are $n/2$ antidiagonal blocks in B **then** ▷ *Use Eq. (31)*
    $\beta \leftarrow B_{1,2} (= B_{2,1})$
    $\bar{A}_{*,2} \leftarrow \beta \cdot \bar{A}_{*,2}$; $\bar{A} \leftarrow \left(\begin{smallmatrix} \bar{A} & \bar{A}_{*,1} + \bar{A}_{*,2} \end{smallmatrix}\right) \in \mathbb{F}_q^{m \times (n+1)}$
    $\ell \leftarrow 1$; $\delta \leftarrow 1$
  else
    $\delta \leftarrow \mathtt{sqrt}(\bar{D}_{\ell,\ell})$ where ℓ is s.t. $\bar{D}_{\ell,\ell}$ is a scalar block
  for all remaining antidiagonal blocks in B at $(j, j+1)$ **do**
    $\beta \leftarrow B_{j,j+1} (= B_{j+1,j})$ ▷ *Use Eq. (32)*
    $\bar{A}_{*,\ell} \leftarrow \delta \cdot \bar{A}_{*,\ell}$; $\bar{A}_{*,j+1} \leftarrow \beta \cdot \bar{A}_{*,j+1}$
    $\left(\begin{smallmatrix} \bar{A}_{*,\ell} & \bar{A}_{*,j} & \bar{A}_{*,j+1} \end{smallmatrix}\right) \leftarrow \left(\begin{smallmatrix} \bar{A}_{*,\ell} & \bar{A}_{*,j} & \bar{A}_{*,j+1} \end{smallmatrix}\right) \cdot \left(\begin{smallmatrix} 1 & 1 & 1 \\ 1 & 0 & 1 \\ 0 & 1 & 1 \end{smallmatrix}\right)$
    $\delta \leftarrow 1$
return $\mathtt{syrkd}(\bar{A}, \bar{D})$ ▷ $\bar{A} \cdot \bar{D} \cdot \bar{A}^\mathsf{T}$ *using Algorithm 5*

REFERENCES

[1] M. Baboulin, L. Giraud, and S. Gratton. A parallel distributed solver for large dense symmetric systems: Applications to geodesy and electromagnetism problems. *Int. J. of HPC Applications*, 19(4):353–363, 2005. doi:10.1177/1094342005056134.

[2] G. Beniamini and O. Schwartz. Faster matrix multiplication via sparse decomposition. In *Proc. SPAA'19*, pages 11–22, 2019. doi:10.1145/3323165.3323188.

[3] M. Bodrato. A Strassen-like matrix multiplication suited for squaring and higher power computation. In *Proc. ISSAC'10*, pages 273–280. ACM, 2010. doi:10.1145/1837934.1837987.

[4] R. P. Brent. Algorithms for matrix multiplication. Technical Report STAN-CS-70-157, C.S. Dpt. Standford University, Mar. 1970.

[5] J. Brillhart. Note on representing a prime as a sum of two squares. *Math. of Computation*, 26(120):1011–1013, 1972. doi:10.1090/S0025-5718-1972-0314745-6.

[6] N. H. Bshouty. On the additive complexity of 2×2 matrix multiplication. *Inf. Processing Letters*, 56(6):329–335, Dec. 1995. doi:10.1016/0020-0190(95)00176-X.

[7] Ph. Chatelin. On transformations of algorithms to multiply 2×2 matrices. *Inf. processing letters*, 22(1):1–5, Jan. 1986. doi:10.1016/0020-0190(86)90033-5.

[8] H. F. de Groot. On varieties of optimal algorithms for the computation of bilinear mappings I. The isotropy group of a bilinear mapping. *Theoretical Computer Science*, 7(2):1–24, 1978. doi:10.1016/0304-3975(78)90038-5.

[9] H. F. de Groot. On varieties of optimal algorithms for the computation of bilinear mappings II. Optimal algorithms for 2×2-matrix multiplication. *Theoretical Computer Science*, 7(2):127–148, 1978. doi:10.1016/0304-3975(78)90045-2.

[10] J. J. Dongarra, J. Du Croz, S. Hammarling, and I. S. Duff. A Set of Level 3 Basic Linear Algebra Subprograms. *ACM Trans. on Math. Soft.*, 16(1):1–17, Mar. 1990. doi:10.1145/77626.79170.

[11] J.-G. Dumas, P. Giorgi, and C. Pernet. Dense linear algebra over prime fields. *ACM Trans. on Math. Soft.*, 35(3):1–42, Nov. 2008. doi:10.1145/1391989.1391992.

[12] J.-G. Dumas and C. Pernet. Symmetric indefinite elimination revealing the rank profile matrix. In *Proc. ISSAC'18*, pages 151–158. ACM, 2018. doi:10.1145/3208976.3209019.

[13] J.-C. Faugère. FGb: A Library for Computing Gröbner Bases. In *Proc ICMS'10*, LNCS, 6327, pages 84–87, 2010. doi:10.1007/978-3-642-15582-6_17.

[14] The FFLAS-FFPACK group. *FFLAS-FFPACK: Finite Field Linear Algebra Subroutines / Package*, 2019. v2.4.1. URL: http://github.com/linbox-team/fflas-ffpack.

[15] N. J. Higham. Stability of a method for multiplying complex matrices with three real matrix multiplications. *SIMAX*, 13(3):681–687, 1992. doi:10.1137/0613043.

[16] E. Karstadt and O. Schwartz. Matrix multiplication, a little faster. In *Proc. SPAA'17*, pages 101–110. ACM, 2017. doi:10.1145/3087556.3087579.

[17] F. Le Gall. Powers of tensors and fast matrix multiplication. In *Proc ISSAC'14*, pages 296–303. ACM, 2014. doi:10.1145/2608628.2608664.

[18] A. Lempel. Matrix factorization over $GF(2)$ and trace-orthogonal bases of $GF(2^n)$. *SIAM J. on Computing*, 4(2):175–186, 1975. doi:10.1137/0204014.

[19] G. Seroussi and A. Lempel. Factorization of symmetric matrices and trace-orthogonal bases in finite fields. *SIAM J. on Computing*, 9(4):758–767, 1980. doi:10.1137/0209059.

[20] V. Strassen. Gaussian elimination is not optimal. *Numerische Mathematik*, 13:354–356, 1969. doi:10.1007/BF02165411.

[21] S. Wedeniwski. Primality tests on commutator curves. PhD U. Tübingen, 2001.

[22] S. Winograd. La complexité des calculs numériques. *La Recherche*, 8:956–963, 1977.

[23] Z. Xianyi, M. Kroeker, et al. *OpenBLAS, an Optimized BLAS library*, 2019. http://www.openblas.net/.

On the Bit Complexity of Finding Points in Connected Components of a Smooth Real Hypersurface

Jesse Elliott
Cheriton School of Computer Science
University of Waterloo
jakellio@uwaterloo.ca

Mark Giesbrecht
Cheriton School of Computer Science
University of Waterloo
mwg@uwaterloo.ca

Éric Schost
Cheriton School of Computer Science
University of Waterloo
eschost@uwaterloo.ca

Abstract

We present a full analysis of the bit complexity of an efficient algorithm for the computation of at least one point in each connected component of a smooth real hypersurface. This is a basic and important operation in semi-algebraic geometry: it gives an upper bound on the number of connected components of a real hypersurface, and is also used in many higher level algorithms.

Our starting point is an algorithm by Safey El Din and Schost (*Polar varieties and computation of one point in each connected component of a smooth real algebraic set*, ISSAC'03). This algorithm uses random changes of variables that are proved to generically ensure certain desirable geometric properties. The cost of the algorithm was given in an algebraic complexity model; the analysis of the bit complexity and the error probability were left for future work.

Our paper answers these questions. Our main contribution is a quantitative analysis of several genericity statements, such as Thom's weak transversality theorem or Noether normalization properties for polar varieties.

CCS Concepts

• **Computing methodologies → Algebraic algorithms**.

Keywords

Real algebraic geometry; weak transversality; Noether position; complexity

ACM Reference Format:
Jesse Elliott, Mark Giesbrecht, and Éric Schost. 2020. On the Bit Complexity of Finding Points in Connected Components of a Smooth Real Hypersurface. In *International Symposium on Symbolic and Algebraic Computation (ISSAC '20), July 20–23, 2020, Kalamata, Greece.* ACM, New York, NY, USA, 8 pages. https://doi.org/10.1145/3373207.3404058

1 Introduction

Background and problem statement. Computing one point in each connected component of a real algebraic set S is a basic subroutine in real algebraic and semi-algebraic geometry; it is also useful in its own right, since it allows one to decide if S is empty or not.

In this paper, we consider the case where S is given as $S = V \cap \mathbb{R}^n$, where $V = V(f) \subset \mathbb{C}^n$ is a complex hypersurface defined by a squarefree polynomial $f \in \mathbb{Z}[X_1, \ldots, X_n]$. Algorithms for this task have been known for decades, and their complexity is to some extent well understood. Suppose that f has degree d, and coefficients of bit-size h. Without making any assumption on f, the algorithm given in [7, Section 13.1] solves our problem using $d^{O(n)}$ operations in $\mathbb{Q}$; in addition, the output of the algorithm is represented by polynomials of degree $d^{O(n)}$, with coefficients of bit-size $hd^{O(n)}$. The key idea behind this algorithm goes back to [18]: sample points are found through the computation of critical points of well-chosen functions on $V(f)$.

The number of connected components of $V(f)$ admits the lower bound $d^{\Omega(n)}$, so up to polynomial factors this result is optimal. However, due to the generality of the algorithm, the constant hidden in the exponent $O(n)$ in its runtime turns out to be rather large: the algorithm relies on infinitesimal deformations, that affect runtime non-trivially.

In this paper, we will work under the additional assumption that $V = V(f)$ is a *smooth* complex hypersurface. We place ourselves in the continuation of the line of work initiated by [4]: that reference deals with cases where V is smooth and $V \cap \mathbb{R}^n$ is compact, pointing out how *polar varieties* (that were introduced in the 1930's in order to define characteristic classes [25, 34]) can play a role in effective real geometry. This paper was extended in several directions: to V being a smooth complete intersection, still with $V \cap \mathbb{R}^n$ compact [5], then without the compactness assumption [6, 28]; the smoothness assumption was then partly dropped in [2, 3].

Our starting point is the algorithm in [28]. In the hypersurface case, its runtime is $d^{(4+o(1))n}$ operations in $\mathbb{Q}$. As with many results in this vein, the algorithm is randomized: we need to assume that we are in generic coordinates; this is done by applying a random change of coordinates prior to all computations. In addition, the algorithm relies on procedures for solving systems of polynomial equations that are themselves randomized. Altogether, we choose $n^{O(1)}$ random vectors, each of them in an affine space of dimension $n^{O(1)}$; every time a choice is made, there exists a hypersurface of the parameter space that one has to avoid in order to guarantee success. In this paper, we revisit this algorithm and give a complete analysis of its probability of success and its bit complexity.

Data structures. The output of the algorithm is a finite set in $\overline{\mathbb{Q}}^n$. To represent it, we rely on a widely used data structure based on univariate polynomials [1, 13–16, 22, 23, 26]. For a zero-dimensional algebraic set $S \subset \mathbb{C}^n$ defined over $\mathbb{Q}$, a *zero-dimensional parameterization* $\mathscr{Q} = ((q, v_1, \ldots, v_n), \lambda)$ of S consists in polynomials $(q, v_1, \ldots, v_n)$, such that $q \in \mathbb{Q}[T]$ is monic and squarefree, all v_i's

ISSAC '20, July 20–23, 2020, Kalamata, Greece

ACM ISBN 978-1-4503-7100-1/20/07…$15.00
https://doi.org/10.1145/3373207.3404058

are in $\mathbb{Q}[T]$ and satisfy $\deg(v_i) < \deg(q)$, and in a $\mathbb{Q}$-linear form λ in variables $X_1, \ldots, X_n$, such that

- $\lambda(v_1, \ldots, v_n) = Tq' \bmod q$;
- we have the equality $S = \left\{ \left(\frac{v_1(\tau)}{q'(\tau)}, \ldots, \frac{v_n(\tau)}{q'(\tau)} \right) \mid q(\tau) = 0 \right\}$.

The constraint on λ says that the roots of q are the values taken by λ on S. The parameterization of the coordinates by rational functions having q' as a denominator goes back to [22, 23]: as pointed out in [1], it allows one to control precisely the size of the coefficients of $v_1, \ldots, v_n$.

Main result. To state our main result, we need to define the *height* of a rational number, and of a polynomial with rational coefficients.

The *height* of a non-zero $a = u/v \in \mathbb{Q}$ is the maximum of $\ln(|u|)$ and $\ln(v)$, where $u \in \mathbb{Z}$ and $v \in \mathbb{N}$ are coprime. For a polynomial f with rational coefficients, if $v \in \mathbb{N}$ is the minimal common denominator of all non-zero coefficients of f, then the *height* $\mathrm{ht}(f)$ of f is defined as the maximum of the logarithms of v and of the absolute values of the coefficients of vf.

Theorem 1.1. *Suppose that $f \in \mathbb{Z}[X_1 \ldots, X_n]$ is squarefree, satisfies $\deg(f) \le d$ and $\mathrm{ht}(f) \le b$, and that $V(f) \subset \mathbb{C}^n$ is smooth. Also suppose that $0 < \epsilon < 1$.*

There exists a randomized algorithm that takes f and ϵ as input and produces n zero-dimensional parameterizations, the union of whose zeros includes at least one point in each connected component of $V(f) \cap \mathbb{R}^n$, with probability at least $1 - \epsilon$. Otherwise, the algorithm either returns a proper subset of the points, or FAIL. In any case, the algorithm uses

$$O^{\sim}(d^{3n+1}(\log 1/\epsilon)(b + \log 1/\epsilon))$$

bit operations. The polynomials in the output have degree at most d^n, and height

$$O^{\sim}(d^{n+1}(b + \log 1/\epsilon)).$$

Here we assume that f is given as a dense polynomial. Following references such as [4, 14–16, 28], it would be possible to refine the runtime estimate by assuming that f is given by a *straight-line program* (that is, a sequence of operations $+, -, \times$ that takes as input $X_1, \ldots, X_n$ and evaluates f). Any polynomial of degree d in n variables can be computed by a straight-line program that does $O(d^n)$ operations: evaluate all monomials of degree up to d in n variables, multiply them by their respective coefficients and sum the results. However, some inputs may be given by shorter straight-line program, and the algorithm would actually be able to benefit from this.

The algorithm itself is rather simple. To describe it, we need to define *polar varieties*, which will play a crucial role in this paper. Let $V = V(f)$, for f as in the theorem. For $i \in \{1, \ldots, n-1\}$, denote by $\pi_i : \mathbb{C}^n \to \mathbb{C}^i$ the projection $(x_1, \ldots, x_n) \mapsto (x_1, \ldots, x_i)$. The i-th *polar variety*

$$W(\pi_i, V) := \{\boldsymbol{x} \in V \mid \dim \pi_i(T_{\boldsymbol{x}}V) < i\}$$

is the set of critical points of π_i on V. It is thus defined by the vanishing of

$$f, \frac{\partial f}{\partial X_{i+1}}, \ldots, \frac{\partial f}{\partial X_n}.$$

In general, we cannot say much about the geometry of $W(\pi_i, V)$, but if we apply a generic change of coordinates $\boldsymbol{A}$ to f, then $W(\pi_i, V)$ is known to be equidimensional of dimension $(i-1)$ or empty [4], and to be in so-called *Noether position* [28] (background notions in algebraic geometry are in [12, 24, 33]; we will recall key definitions). If this is the case, it suffices to choose arbitrary $\sigma_1, \ldots, \sigma_{n-1}$ in $\mathbb{Q}$, and solve the systems defined by

$$X_1 - \sigma_1, \ldots, X_{i-1} - \sigma_{i-1}, f, \frac{\partial f}{\partial X_{i+1}}, \ldots, \frac{\partial f}{\partial X_n}, \tag{1}$$

for $i = 1, \ldots, n$. They all admit finitely many solutions, and Theorem 2 in [28] proves that the union of their solution sets contains one point on each connected component of $V \cap \mathbb{R}^n$.

Our main contribution is to analyze precisely what conditions on our change of coordinates $\boldsymbol{A}$ guarantee success. This is done by revisiting the key ingredients in the proofs given in [4] and [28], and giving quantitative versions of these results, bounding the degree of the hypersurfaces we have to avoid. To solve the equations (1), we use the algorithm in [31], for which a complete bit complexity analysis is available.

This work should be seen as a first step toward the analysis of further randomized algorithms in real algebraic geometry. An immediate follow-up question would be to handle the case of algebraic sets defined by *regular sequences*: the algorithm in [28] still applies, but the modifications needed are beyond the scope of this publication. Further still, randomized algorithms for deciding *connectivity queries* on smooth, compact algebraic sets have been developed in a series of papers [29, 32], and could be revisited using the techniques introduced here.

2 Genericity properties

Consider $f \in \mathbb{Z}[X_1, \ldots, X_n]$ with total degree d, and assume that f is squarefree and that $V(f) \subset \mathbb{C}^n$ is smooth. The key to the proof of Theorem 1.1 is the following quantitative version of facts we stated above, namely that in generic coordinates, polar varieties are smooth, equidimensional, and in Noether position (or empty).

We recall that an equidimensional algebraic set $X \subset \mathbb{C}^n$ of dimension d is in *Noether position* for the projection π_d when the extension $\mathbb{C}[X_1, \ldots, X_d] \to \mathbb{C}[X_1, \ldots, X_n]/I(X)$ is integral; here, $I(X) \subset \mathbb{C}[X_1, \ldots, X_n]$ is the defining ideal of X. In this case, for any $\boldsymbol{x} \in \mathbb{C}^d$, the fiber $X \cap \pi_d^{-1}(\boldsymbol{x})$ has dimension zero (so it is finite and not empty).

For i in $\{1, \ldots, n\}$ and f as above, we will let $\mathfrak{J}(i, f)$ denote the sequence of $n - (i-1)$ polynomials $(f, \partial f/\partial X_{i+1}, \ldots, \partial f/\partial X_n)$. As pointed out in the introduction, their zero-set is the i-th polar variety $W(\pi_i, V(f))$. Then, we say that f satisfies $\mathbf{H}_i$ if

(1) For any $\boldsymbol{x}$ in $W(\pi_i, V(f))$, the Jacobian matrix $\mathbf{jac}_{\boldsymbol{x}}(\mathfrak{J}(i, f))$ has full rank $n - (i-1)$ at $\boldsymbol{x}$.
By the Jacobian Criterion [12, Corollary 16.20], this implies that $W(\pi_i, V(f))$ is either empty or $(i-1)$-equidimensional, and that $\mathfrak{J}(i, f)$ defines a radical ideal.

(2) $W(\pi_i, V(f))$ is either empty or in Noether position for π_{i-1}.

Given $\boldsymbol{\sigma} = (\sigma_1, \ldots, \sigma_{i-1})$ in $\mathbb{C}^{i-1}$, we further say that f and $\boldsymbol{\sigma}$ satisfy $\mathbf{H}'_i$ if

(1) For any root $\boldsymbol{x}$ of

$$(X_1 - \sigma_1, \ldots, X_{i-1} - \sigma_{i-1}, f, \partial f/\partial X_{i+1}, \ldots, \partial f/\partial X_n),$$

the Jacobian matrix of these equations at $\boldsymbol{x}$ has full rank n. *By the Jacobian Criterion [12, Corollary 16.20], this implies that there are finitly many solutions to these equations.*

Even if f does not initially satisfy $\mathbf{H}_i$, it does after applying a generic change of variables. The precise statement is as follows, for which we use the following notation. For a matrix $\boldsymbol{A}$ in $\mathbb{C}^{n\times n}$ and g in $\mathbb{C}[X_1,\dots,X_n]$ we write $g^{\boldsymbol{A}} := g(\boldsymbol{AX}) \in \mathbb{C}[X_1,\dots,X_n]$, where $\boldsymbol{X}$ is the column vector with entries $X_1,\dots,X_n$.

Note that for a variety $Y \subset \mathbb{C}^n$, we can define $Y^{\boldsymbol{A}}$ as the image of Y by the map $\phi_{\boldsymbol{A}} : \boldsymbol{x} \mapsto \boldsymbol{A}^{-1}\boldsymbol{x}$. Note that $W(\pi_i, V(f^{\boldsymbol{A}}))$ may not equal $W(\pi_i, V(f))^{\boldsymbol{A}}$, as, for instance, their dimensions may vary.

We will also have to consider matrices with generic entries. For this, we introduce n^2 new indeterminates $(\mathfrak{A}_{j,k})_{1\le j,k\le n}$. Then, $\mathfrak{A}$ will denote the matrix with entries $(\mathfrak{A}_{j,k})_{1\le j,k\le n}$, $\mathbb{C}(\mathfrak{A})$ will denote the rational function field $\mathbb{C}((\mathfrak{A}_{j,k})_{1\le j,k\le n})$ and $\mathbb{C}[\mathfrak{A}]$ the polynomial ring $\mathbb{C}[(\mathfrak{A}_{j,k})_{1\le j,k\le n}]$. For f as above, we will then define the polynomial $f^{\mathfrak{A}} := f(\mathfrak{A}\boldsymbol{X})$, which we may consider in either $\mathbb{C}(\mathfrak{A})[X_1,\dots,X_n]$ or $\mathbb{C}[\mathfrak{A},X_1,\dots,X_n]$.

This being said, our two key results are the following.

Theorem 2.1. *For $i = 1,\dots,n$, there exists a non-zero polynomial $\Delta_i \in \mathbb{C}[\mathfrak{A}]$ of degree at most $5n^2(2d)^{2n}$ such that if $\boldsymbol{A} \in \mathbb{C}^{n\times n}$ does not cancel Δ_i, then $\boldsymbol{A}$ is invertible and $f^{\boldsymbol{A}}$ satisfies $\mathbf{H}_i$.*

Theorem 2.2. *For $i = 1,\dots,n$, suppose that f satisfies $\mathbf{H}_i$, then there exists a non-zero polynomial $\Xi_i \in \mathbb{C}[S_1,\dots,S_{i-1}]$ of degree at most d^{2n} such that if $\boldsymbol{\sigma} \in \mathbb{C}^{i-1}$ does not cancel Ξ_i, then f and $\boldsymbol{\sigma}$ satisfy $\mathbf{H}'_i$.*

The proof of these theorems occupies the next two sections. Some related results appear in the literature; for instance, Lemma 5 in [20] or Proposition 4.5 in [21] are quantitative Noether position statements. However, Theorem 2.1 does not follow from these previous results. Indeed, those references would allow us to quantify when $W(\pi_i, V(f))^{\boldsymbol{A}}$ is in Noether position, whereas we need to understand when $W(\pi_i, V(f^{\boldsymbol{A}}))$ is. As we pointed out before, these two sets are in general different.

3 Weak transversality and applications

Sard's lemma states that the set of critical values of a smooth function $\mathbb{R}^n \to \mathbb{R}^m$ has measure zero. One can give "algebraic" versions of it, for semi-algebraic mappings $\mathbb{R}^n \to \mathbb{R}^m$ as in [9, Chapter 9], or polynomial mappings $\mathbb{C}^n \to \mathbb{C}^m$ as in [24, Chapter 3], for which the sets of critical values are contained in strict semi-algebraic, resp. algebraic sets in the codomain. Thom's weak transversality lemma, as given for instance in [11], generalizes Sard's lemma. In this section, we consider a particular case of this result (transversality to a point), and establish a quantitative version of it; this will allow us to establish the first item in property $\mathbf{H}_i$, as well as property $\mathbf{H}'_i$.

3.1 Weak transversality

Transversality to a point can be rephrased entirely in terms of critical and regular values. Recall that if Ψ is a mapping from a smooth algebraic set Y to $\mathbb{C}^t$, with $t \le \dim(Y)$, a *critical point* of Ψ is a point $\boldsymbol{y} \in Y$ such that the image of the tangent space $T_{\boldsymbol{y}}Y$ by the differential $d_{\boldsymbol{y}}\Psi$ has dimension less than t. When for instance $Y = \mathbb{C}^v$, we have $T_{\boldsymbol{y}}Y = \mathbb{C}^v$ and this condition is equivalent to the Jacobian of Ψ having rank less than t at $\boldsymbol{y}$. *Critical values* are the images by Ψ of critical points; the complement of this set are the *regular values* (so a regular value is not necessarily in the image of Ψ).

Let then n, s, and m be positive integers, with $m \le n$, and denote by $\boldsymbol{\Phi} : \mathbb{C}^n \times \mathbb{C}^s \to \mathbb{C}^m$ a mapping defined by polynomials in $\mathbb{C}[\boldsymbol{X},\boldsymbol{\Theta}]$, where $\boldsymbol{X}$, resp. $\boldsymbol{\Theta}$, is a set of n, resp. s, indeterminates. For $\boldsymbol{\vartheta}$ in $\mathbb{C}^s$, let $\boldsymbol{\Phi}_{\boldsymbol{\vartheta}} : \mathbb{C}^n \to \mathbb{C}^m$ be the induced mapping $\boldsymbol{x} \mapsto \boldsymbol{\Phi}(\boldsymbol{x},\boldsymbol{\vartheta})$. The transversality result we will need is the following.

Proposition 3.1 (Weak transversality). *Suppose that $\mathbf{0}$ is a regular value of $\boldsymbol{\Phi}$. Then there exists a non-zero polynomial $\Gamma \in \mathbb{C}[\boldsymbol{\Theta}]$ of degree at most d^{m+n} such that for $\boldsymbol{\vartheta}$ in $\mathbb{C}^s$, if $\Gamma(\boldsymbol{\vartheta}) \neq 0$, then $\mathbf{0}$ is a regular value of $\boldsymbol{\Phi}_{\boldsymbol{\vartheta}}$.*

The following simple example shows this result at work. Consider a squarefree f in $\mathbb{C}[X_1,X_2]$, such that $V(f)$ is a smooth curve in $\mathbb{C}^2$, and let the mapping $\boldsymbol{\Phi} : \mathbb{C}^2 \times \mathbb{C} \to \mathbb{C}^2$ be defined by $\boldsymbol{\Phi}(X_1,X_2,\Theta) = (f(X_1,X_2), X_1 - \Theta)$. One checks that the Jacobian of $\boldsymbol{\Phi}$ with respect to (X_1,X_2,Θ) has rank two at any point in $\boldsymbol{\Phi}^{-1}(\mathbf{0})$, so the assumptions of the proposition apply. We deduce that for a generic ϑ in $\mathbb{C}$, that is, for all ϑ in $\mathbb{C}$ except a finite number, the ideal $(f(X_1,X_2), X_1 - \vartheta)$ is radical in $\mathbb{C}[X_1,X_2]$; equivalently, $f(\vartheta,X_2)$ is squarefree. We will revisit this example in Section 3.3.

The rest of the subsection is devoted to the proof of the proposition. The proof of [30, Theorem B.3] already shows the existence of Γ; it is essentially the classical proof for smooth mappings [11, Section 3.7], written in an algebraic context. In what follows, we revisit this proof, establishing a bound on the degree of Γ.

Put $V := \boldsymbol{\Phi}^{-1}(\mathbf{0})$. If V is empty, there is nothing to do, since all values $\boldsymbol{\vartheta}$ in $\mathbb{C}^s$ satisfy the conclusion of the proposition. Thus, we assume that V is not empty. Then, the Jacobian criterion shows that V is smooth and $(n+s-m)$-equidimensional.

We will reuse the following fact, proved in [30]. Consider the projection $\pi : (\boldsymbol{x},\boldsymbol{\vartheta}) \in \mathbb{C}^n \times \mathbb{C}^s \mapsto \boldsymbol{\vartheta} \in \mathbb{C}^s$. Let Z be the set of critical points of $\pi_{|V}$, and consider its projection $\pi(Z)$ in $\mathbb{C}^s$. This is the set of critical values of $\pi_{|V}$; hence, by the algebraic form of Sard's lemma (see [24, Theorem 3.7] for irreducible V and [30, Proposition B.2] for general V), its Zariski closure $\overline{\pi(Z)}$ is a strict closed subset of $\mathbb{C}^s$. As we will see below, if $\boldsymbol{\vartheta} \in \mathbb{C}^s$ is not in $\overline{\pi(Z)}$, then $\mathbf{0}$ is a regular value of $\boldsymbol{\Phi}_{\boldsymbol{\vartheta}}$.

To describe the set Z of critical points of $\pi_{|V}$, let $\boldsymbol{M}$ denote the $(s+m)\times(s+n)$ Jacobian matrix with entries in $\mathbb{C}[\boldsymbol{X},\boldsymbol{\Theta}]$ given by $\boldsymbol{M} := \mathbf{jac}_{\boldsymbol{X},\boldsymbol{\Theta}}(\pi,\boldsymbol{\Phi})$, that is,

$$\boldsymbol{M} = \begin{bmatrix} \mathbf{jac}_{\boldsymbol{X},\boldsymbol{\Theta}}(\pi) \\ \mathbf{jac}_{\boldsymbol{X},\boldsymbol{\Theta}}(\boldsymbol{\Phi}) \end{bmatrix} = \begin{bmatrix} \mathbf{0}_{s\times n} & \mathbf{I}_s \\ \mathbf{jac}_{\boldsymbol{X},\boldsymbol{\Theta}}(\boldsymbol{\Phi}) \end{bmatrix}.$$

Lemma 3.2. *For $(\boldsymbol{x},\boldsymbol{\vartheta})$ in V, $(\boldsymbol{x},\boldsymbol{\vartheta})$ is in Z if and only if the matrix $\boldsymbol{M}$ has rank less than $s+m$ at $(\boldsymbol{x},\boldsymbol{\vartheta})$.*

Proof. Take $(\boldsymbol{x},\boldsymbol{\vartheta})$ on V, and let $\boldsymbol{K}(\boldsymbol{x},\boldsymbol{\vartheta})$ be the Jacobian matrix $\mathbf{jac}_{\boldsymbol{X},\boldsymbol{\Theta}}(\boldsymbol{\Phi})$ taken at $(\boldsymbol{x},\boldsymbol{\vartheta})$. Then, the rank of $\boldsymbol{M}(\boldsymbol{x},\boldsymbol{\vartheta})$ can be written as $\operatorname{rank}(\boldsymbol{K}(\boldsymbol{x},\boldsymbol{\vartheta})) + \operatorname{rank}([\mathbf{0}_{s\times n}\ \mathbf{I}_s] \mid \ker \boldsymbol{K}(\boldsymbol{x},\boldsymbol{\vartheta}))$, where the latter is the rank of the restriction of $[\mathbf{0}_{s\times n}\ \mathbf{I}_s]$ to the nullspace of $\boldsymbol{K}(\boldsymbol{x},\boldsymbol{\vartheta})$.

Since V is smooth, $\boldsymbol{K}(\boldsymbol{x},\boldsymbol{\vartheta})$ has full rank $\operatorname{codim}(V) = m$. On the other hand, the nullspace of $\boldsymbol{K}(\boldsymbol{x},\boldsymbol{\vartheta})$ is the tangent space $T_{\boldsymbol{x},\boldsymbol{\vartheta}}V$, and $\operatorname{rank}([\mathbf{0}_{s\times n}\ \mathbf{I}_s] \mid \ker \boldsymbol{K}(\boldsymbol{x},\boldsymbol{\vartheta}))$ is the dimension of $\pi(T_{\boldsymbol{x},\boldsymbol{\vartheta}}V)$. In

other words, the rank of $\boldsymbol{M}(\boldsymbol{x}, \boldsymbol{\vartheta})$ is equal to $m + \dim(\pi(T_{\boldsymbol{x},\boldsymbol{\vartheta}}V))$; this implies the claim in the lemma. □

Therefore, we can characterize the set Z of critical points of $\pi_{|V}$ as those points satisfying $\boldsymbol{\Phi}(\boldsymbol{x}, \boldsymbol{\vartheta}) = \boldsymbol{0}$ and where all minors of $\boldsymbol{M}$ of order $s+m$ vanish. We can actually describe this set using a smaller matrix, by discarding certain minors that are identically zero. Let indeed $\boldsymbol{J}$ denote the $m \times n$ submatrix of the Jacobian of $\boldsymbol{\Phi}$ consisting of the first n columns. This is the Jacobian matrix of $\boldsymbol{\Phi}$ with respect to $\boldsymbol{X}$.

Lemma 3.3. *For $(\boldsymbol{x}, \boldsymbol{\vartheta})$ in V, $(\boldsymbol{x}, \boldsymbol{\vartheta})$ is in Z if and only if $\boldsymbol{J}(\boldsymbol{x}, \boldsymbol{\vartheta})$ has rank less than m.*

Proof. Notice

$$M(\boldsymbol{x}, \boldsymbol{\vartheta}) = \begin{bmatrix} \boldsymbol{0}_{s\times n} & \mathbf{I}_s \\ \boldsymbol{J}(\boldsymbol{x}, \boldsymbol{\vartheta}) & \boldsymbol{J}'(\boldsymbol{x}, \boldsymbol{\vartheta}) \end{bmatrix},$$

where $\boldsymbol{J}'$ consists of the remaining columns of the Jacobian matrix of $\boldsymbol{\Phi}$. Then, the rank of the former matrix is equal to the rank of

$$M(\boldsymbol{x}, \boldsymbol{\vartheta}) = \begin{bmatrix} \boldsymbol{0}_{s\times n} & \mathbf{I}_s \\ \boldsymbol{J}(\boldsymbol{x}, \boldsymbol{\vartheta}) & \boldsymbol{0}_{m\times s} \end{bmatrix},$$

and the conclusion follows. □

In particular, take $\boldsymbol{\vartheta}$ in $\mathbb{C}^s - \overline{\pi(Z)}$. Then for all $\boldsymbol{x}$ in $\boldsymbol{\Phi}_{\boldsymbol{\vartheta}}^{-1}(\boldsymbol{0})$, $(\boldsymbol{x}, \boldsymbol{\vartheta})$ is in V, so it is not in Z. The previous lemma then implies that the Jacobian matrix $\boldsymbol{J}$ of $\boldsymbol{\Phi}_{\boldsymbol{\vartheta}}$ has full rank m at $(\boldsymbol{x}, \boldsymbol{\vartheta})$. In other words, $\boldsymbol{0}$ is a regular value of $\boldsymbol{\Phi}_{\boldsymbol{\vartheta}}$, as claimed.

Our next step is to bound the degree of Z. In that, we use the definition of degree given in [19]: the degree of an irreducible algebraic set is the number of intersection points it has with a generic hyperplane of complementary dimension, and the degree of an arbitrary algebraic set is the sum of the degrees of its irreducible components. To obtain an estimate on the degree of Z, rather than considering minors of $\boldsymbol{J}$, we will rewrite the condition that $\boldsymbol{J}(\boldsymbol{x}, \boldsymbol{\vartheta})$ has rank less than m as the existence of a non-trivial left kernel element.

For this, we let $\boldsymbol{L} = [L_1, \ldots, L_m]$ be new variables, thought of as Lagrange multipliers, and consider the "Lagrange polynomials" $\mathscr{L}_1, \ldots, \mathscr{L}_n$, with

$$[\mathscr{L}_1 \cdots \mathscr{L}_n] := \boldsymbol{L} \cdot \boldsymbol{J}(\boldsymbol{x}, \boldsymbol{\vartheta}).$$

Denote by $\mathfrak{Z} \subset \mathbb{C}^{n+s+m}$ the algebraic set defined by the vanishing of $\mathscr{L}_1, \ldots, \mathscr{L}_n$, and $\boldsymbol{\Phi}$, and by $\mathfrak{Z}'$ the algebraic set

$$\mathfrak{Z}' := \overline{\mathfrak{Z} - \{(\boldsymbol{x}, \boldsymbol{\vartheta}, 0, \ldots, 0) \in \mathbb{C}^{n+s+m} \mid (\boldsymbol{x}, \boldsymbol{\vartheta}, 0 \ldots, 0) \in \mathfrak{Z}\}},$$

where the bar denotes Zariski closure (we have to remove such points, since $L_1 = \cdots = L_m = 0$ is always a trivial solution to the Lagrange equations). Finally, consider the projection

$$\begin{aligned} \mu : \mathbb{C}^{n+s+m} &\to \mathbb{C}^{n+s} \\ (\boldsymbol{x}, \boldsymbol{\vartheta}, \boldsymbol{\ell}) &\mapsto (\boldsymbol{x}, \boldsymbol{\vartheta}). \end{aligned}$$

Lemma 3.4. *The algebraic set Z is equal to the projection $\mu(\mathfrak{Z}')$.*

Proof. Take $(\boldsymbol{x}, \boldsymbol{\vartheta})$ in Z. Then, $(\boldsymbol{x}, \boldsymbol{\vartheta})$ cancels all polynomials $\boldsymbol{\Phi}$, and there exists $\boldsymbol{\ell} = (\ell_1, \ldots, \ell_m)$, not identically zero, such that $(\boldsymbol{x}, \boldsymbol{\vartheta}, \boldsymbol{\ell})$ cancels the Lagrange polynomials. This implies that $(\boldsymbol{x}, \boldsymbol{\vartheta}, \boldsymbol{\ell})$ is in $\mathfrak{Z} - \{(\boldsymbol{x}', \boldsymbol{\vartheta}', 0, \ldots, 0) \in \mathbb{C}^{n+s+m} \mid (\boldsymbol{x}', \boldsymbol{\vartheta}', 0 \ldots, 0) \in \mathfrak{Z}\}$, and thus in $\mathfrak{Z}'$. This proves the inclusion $Z \subset \mu(\mathfrak{Z}')$.

Conversely, take an irreducible component Y of $\mathfrak{Z}'$. We prove that $\mu(Y)$ is contained in Z. By construction, there exists an open dense subset $Y^o \subset Y$ such that for any $(\boldsymbol{x}, \boldsymbol{\vartheta}, \boldsymbol{\ell})$ in Y^o, $\boldsymbol{\ell}$ is not identically zero. As a result, $(\boldsymbol{x}, \boldsymbol{\vartheta})$ is in Z, that is, $\mu(Y^o)$ is in Z. This implies that its Zariski closure $\overline{\mu(Y^o)}$ is in Z. Since $\mu(Y)$ is contained in $\overline{\mu(Y^o)}$, we deduce $\mu(Y) \subset Z$. Taking the union over all Y, we get $\mu(\mathfrak{Z}') \subset Z$, as claimed. □

Corollary 3.5. *The degree of Z is at most d^{m+n}.*

Proof. The algebraic set $\mathfrak{Z}$ is defined by $m+n$ equations, all of them having degree at most d. It follows from Bézout's Theorem [19] that $\deg(\mathfrak{Z}) \le d^{m+n}$, and the same upper bound holds for $\deg(\mathfrak{Z}')$, since it consists of certain irreducible components of $\mathfrak{Z}$. Applying the projection μ yields the result, since degree cannot increase through projection. □

In particular, we obtain the same degree bound for $\overline{\pi(Z)}$. It then suffices to take for Γ any non-zero polynomial of degree at most d^{m+n} that vanishes on $\overline{\pi(Z)}$; this proves Proposition 3.1.

3.2 Application: property $\mathbf{H}_i(1)$

Let $f \in \mathbb{Z}[X_1, \ldots, X_n]$ have total degree d, with $V(f) \subset \mathbb{C}^n$ smooth. In what follows, we fix i in $1, \ldots, n$, and we prove the following: *there exists a non-zero polynomial $\Delta_{i,1} \in \mathbb{C}[\mathfrak{A}]$ of degree at most $2nd^{2n}$ such that if $\boldsymbol{A} \in \mathbb{C}^{n\times n}$ does not cancel $\Delta_{i,1}$, then $\boldsymbol{A}$ is invertible and $f^{\boldsymbol{A}}$ satisfies $\boldsymbol{H}_i(1)$.*

The following construction is already in [4]; our contribution is the degree estimate. We let $\boldsymbol{\Phi} : \mathbb{C}^n \times \mathbb{C}^{n\times n} \to \mathbb{C}^{n-i+1}$ be the mapping defined by the polynomials

$$\big(f, \mathbf{grad}(f) \cdot \mathfrak{A}_{i+1}, \ldots, \mathbf{grad}(f) \cdot \mathfrak{A}_n\big),$$

where $\mathfrak{A}_1, \ldots, \mathfrak{A}_n$ denote the columns of $\mathfrak{A}$ and $\cdot$ is the dot-product.

Lemma 3.6. *$\boldsymbol{0}$ is a regular value of $\boldsymbol{\Phi}$.*

Proof. Let $(\boldsymbol{x}, \boldsymbol{A}) \in \mathbb{C}^n \times \mathbb{C}^{n\times n}$ be a zero of $\boldsymbol{\Phi}$. We have to show that the Jacobian matrix of the equations defining $\boldsymbol{\Phi}$, taken with respect to $\boldsymbol{X}$ and $\mathfrak{A}$, has full rank $n-i+1$ at $(\boldsymbol{x}, \boldsymbol{A})$. If we set

$$F_j = \frac{\partial f}{\partial X_1} A_{i+j,1} + \ldots + \frac{\partial f}{\partial X_n} A_{i+j,n}, \ 1 \le j \le n-i,$$

this Jacobian matrix is equal to

$$\begin{bmatrix} \frac{\partial f}{\partial X_1} \cdots \frac{\partial f}{\partial X_n} & \cdots & 0 \ldots 0 & \cdots & 0 \ldots 0 \\ \frac{\partial F_1}{\partial X_1} \cdots \frac{\partial F_1}{\partial X_n} & \cdots & \frac{\partial f}{\partial X_1} \cdots \frac{\partial f}{\partial X_n} & \cdots & 0 \ldots 0 \\ \ddots & \ddots & \ddots & \ddots & \ddots \\ \frac{\partial F_{n-i}}{\partial X_1} \cdots \frac{\partial F_{n-i}}{\partial X_n} & \cdots & 0 \ldots 0 & \cdots & \frac{\partial f}{\partial X_1} \cdots \frac{\partial f}{\partial X_n} \end{bmatrix},$$

where the first columns are indexed by $X_1, \ldots, X_n$ and the further ones by $\mathfrak{A}_{1,i+1}, \ldots, \mathfrak{A}_{n,i+1}, \ldots, \mathfrak{A}_{1,n}, \ldots, \mathfrak{A}_{n,n}$. Since $f(\boldsymbol{x}) = 0$, our assumption on f implies that at least one of its partial derivatives is non-zero at $\boldsymbol{x}$, and the conclusion follows. □

Since all equations defining $\boldsymbol{\Phi}$ have degree at most d, it follows by Proposition 3.1 that there exists a non-zero polynomial $\Gamma_i \in \mathbb{C}[\mathfrak{A}]$ of degree at most $d^{2n-i+1} \le d^{2n}$, with the property that, if $\boldsymbol{A} \in \mathbb{C}^{n\times n}$ does not cancel Γ_i, then the Jacobian matrix of

$$\boldsymbol{\Phi}_{\boldsymbol{A}} = \big(f, \mathbf{grad}(f) \cdot \boldsymbol{A}_{i+1}, \ldots, \mathbf{grad}(f) \cdot \boldsymbol{A}_n\big),$$

taken with respect to $\boldsymbol{X}$, has full rank $n-i+1$ at all $\boldsymbol{x}$ that cancels equations. We then define $\Delta_{i,1} := \Gamma_i \det(\mathfrak{A})$; this is a non-zero polynomial of degree at most $d^{2n} + n \leq 2nd^{2n}$.

Let us verify that $\Delta_{i,1}$ satisfies the claim in the preamble. Take $\boldsymbol{A}$ in $\mathbb{C}^{n\times n}$, such that $\Delta_{i,1}(\boldsymbol{A})$ is non-zero. Clearly, $\boldsymbol{A}$ is invertible; it remains to check that $f^{\boldsymbol{A}}$ satisfies $\mathbf{H}_i(1)$. Thus, we take $\boldsymbol{x}$ that cancels $(f^{\boldsymbol{A}}, \partial f^{\boldsymbol{A}}/\partial X_{i+1}, \ldots, \partial f^{\boldsymbol{A}}/\partial X_n)$, and we prove that the Jacobian matrix of these equations, taken with respect to $\boldsymbol{X}$, has full rank $n-i+1$ at $\boldsymbol{x}$. Using the chain rule, the equations above can be rewritten as $\boldsymbol{\Phi}_{\boldsymbol{A}}(\boldsymbol{A}\boldsymbol{x})$, so their Jacobian matrix at $\boldsymbol{x}$ has the same rank as that of $\boldsymbol{\Phi}_{\boldsymbol{A}}$ at $\boldsymbol{A}\boldsymbol{x}$, that is, $n-i+1$. Our claim is proved.

In Section 4, we will need the following by-product of this result: if we consider $f^{\mathfrak{A}} \in \mathbb{C}(\mathfrak{A}_{j,k})[X_1, \ldots, X_n]$ as defined in Section 2, this polynomial satisfies the rank property $\mathbf{H}_i(1)$.

3.3 Application: property $\mathbf{H}_i'$

Let $f \in \mathbb{Z}[X_1, \ldots, X_n]$ and i be as before. We now assume that f satisfies $\mathbf{H}_i(1)$, and we prove the following: *there exists a non-zero polynomial $\Xi_i \in \mathbb{C}[S_1, \ldots, S_{i-1}]$ of degree at most d^{2n} such that if $\boldsymbol{\sigma} = (\sigma_1, \ldots, \sigma_{i-1}) \in \mathbb{C}^{i-1}$ does not cancel Ξ_i, then for any root $\boldsymbol{x}$ of*

$$(X_1 - \sigma_1, \ldots, X_{i-1} - \sigma_{i-1}, f, \partial f/\partial X_{i+1}, \ldots, \partial f/\partial X_n),$$

the Jacobian matrix of these equations at $\boldsymbol{x}$ has full rank n.

Let $\boldsymbol{\Psi} : \mathbb{C}^n \times \mathbb{C}^{i-1} \to \mathbb{C}^n$ be the mapping defined by the polynomials

$$(X_1 - S_1, \ldots, X_{i-1} - S_{i-1}, f, \partial f/\partial X_{i+1}, \ldots, \partial f/\partial X_n)\,.$$

Lemma 3.7. *$\boldsymbol{0}$ is a regular value of $\boldsymbol{\Psi}$.*

Proof. At all zeros $(\boldsymbol{x}, \boldsymbol{\sigma})$ of $\boldsymbol{\Psi}$, the Jacobian matrix of $\boldsymbol{\Psi}$ has full rank n. Indeed, indexing columns by $X_1, \ldots, X_n, S_1, \ldots, S_{i-1}$, this matrix is equal to

$$\begin{bmatrix} \mathbf{I}_{i-1} & \boldsymbol{0}_{(i-1)\times(n-i+1)} & -\mathbf{I}_{i-1} \\ \mathbf{jac}_{\boldsymbol{x}}\left(f, \frac{\partial f}{\partial X_{i+1}}, \ldots, \frac{\partial f}{\partial X_n}\right) & & \boldsymbol{0}_{(n-i+1)\times(i-1)} \end{bmatrix}.$$

Since the Jacobian of $f, \partial f/\partial X_{i+1}, \ldots, \partial f/\partial X_n$ at $\boldsymbol{x}$ is non-zero (by $\mathbf{H}_i$), the entire matrix must have full rank n. Thus, $\boldsymbol{0}$ is a regular value of $\boldsymbol{\Psi}$. □

Since all polynomials defining $\boldsymbol{\Psi}$ have degree at most d, it follows by Proposition 3.1 that there exists a non-zero polynomial Ξ_i in $\mathbb{C}[S_1, \ldots, S_{i-1}]$ of degree at most d^{2n}, with the following property: if $\Xi_i(\boldsymbol{\sigma}) \neq 0$ then at any root $\boldsymbol{x}$ of

$$(X_1 - \sigma_1, \ldots, X_{i-1} - \sigma_{i-1}, f, \partial f/\partial X_{i+1}, \ldots, \partial f/\partial X_n),$$

the Jacobian matrix of these equations has full rank n. Theorem 2.2 is proved.

4 Property $\mathrm{H}_i(2)$: Noether position

Throughout this section, f and $i \in \{1, \ldots, n\}$ are fixed. We prove that there exists a non-zero polynomial Δ_i in n^2 variables and of degree at most $5n^2(2d)^{2n}$ such that if $\boldsymbol{A}$ does not cancel Δ_i, then $\boldsymbol{A}$ is invertible and satisfies both conditions in $\mathbf{H}_i$.

Consider again the matrix of indeterminates $\mathfrak{A} = (\mathfrak{A}_{j,k})_{1\leq j,k\leq n}$ and the field $\mathbb{C}(\mathfrak{A})$, and define $f^{\mathfrak{A}} \in \mathbb{C}(\mathfrak{A})[X_1, \ldots, X_n]$. Since i is fixed, to simplify notation, let $\mathfrak{I}^{\mathfrak{A}}$ denote the following polynomials in $\mathbb{C}(\mathfrak{A})[X_1, \ldots, X_n]$:

$$\mathfrak{I}(i, f^{\mathfrak{A}}) = \left(f^{\mathfrak{A}}, \partial f^{\mathfrak{A}}/\partial X_{i+1}, \ldots, \partial f^{\mathfrak{A}}/\partial X_n\right),$$

and let $W^{\mathfrak{A}}$ denote their zero-set, that is, $W(\pi_i, V(f^{\mathfrak{A}}))$. In Section 3.2, we saw that $f^{\mathfrak{A}}$ satisfies $\mathbf{H}_i(1)$, so that $\mathfrak{I}^{\mathfrak{A}}$ defines a radical ideal, and $W^{\mathfrak{A}}$ is equidimensional of dimension $i-1$. We now point out that $f^{\mathfrak{A}}$ also satisfies $\mathbf{H}_i(2)$.

Lemma 4.1. *The extension*

$$\mathbb{C}(\mathfrak{A})[X_1, \ldots, X_{i-1}] \to \mathbb{C}(\mathfrak{A})[X_1, \ldots, X_n]/\mathfrak{I}^{\mathfrak{A}}$$

is integral.

Proof. Let $(\mathfrak{P}_\ell)_{1\leq\ell\leq L}$ be the prime components of the radical ideal $\mathfrak{I}^{\mathfrak{A}}$. By [28, Proposition 1], for all ℓ,

$$\mathbb{C}(\mathfrak{A})[X_1, \ldots, X_{i-1}] \to \mathbb{C}(\mathfrak{A})[X_1, \ldots, X_n]/\mathfrak{P}_\ell$$

is integral. Therefore polynomials $q_{\ell,j} \in \mathbb{C}(\mathfrak{A})[X_1, \ldots, X_{i-1}, X_j]$ exist, all monic in X_j, with $q_{\ell,j}(X_j) \in \mathfrak{P}_\ell$ for each j in $\{i, \ldots, n\}$. Thence, $Q_j := \prod_{1\leq\ell\leq L} q_{\ell,j}$ is monic in X_j and satisfies $Q_j \in \mathfrak{I}^{\mathfrak{A}}$, for each $j \in \{i, \ldots, n\}$. This proves our claim. □

If P is any polynomial in $\mathbb{C}(\mathfrak{A})[X_1, \ldots, X_n]$, we will let $D \in \mathbb{C}[\mathfrak{A}]$ be the minimal common denominator of all its coefficients, and we will write $\overline{P} := DP$, so that $\overline{P}$ is in $\mathbb{C}[\mathfrak{A}, X_1, \ldots, X_n]$.

Lemma 4.2. *For $j = i, \ldots, n$, there exists a polynomial P_j in $\mathbb{C}(\mathfrak{A})[X_1, \ldots, X_{i-1}, X_j]$, monic in X_j, with $\overline{P_j}$ in $\mathfrak{I}^{\mathfrak{A}}$, and such that* $\deg(\overline{P_j}) \leq (2d)^n$.

Proof. We let $\mathfrak{L}^{\mathfrak{A}}$ denote the extension of $\mathfrak{I}^{\mathfrak{A}}$ given by $\mathfrak{L}^{\mathfrak{A}} := \mathfrak{I}^{\mathfrak{A}} \cdot \mathbb{C}(\mathfrak{A}, X_1, \ldots, X_{i-1})[X_i, \ldots, X_n]$. Then,

$$\mathbb{C}(\mathfrak{A}, X_1, \ldots, X_{i-1}) \to \mathbb{C}(\mathfrak{A}, X_1, \ldots, X_{i-1})[X_i, \ldots, X_n]/\mathfrak{L}^{\mathfrak{A}} \quad (2)$$

is an algebraic extension. On the other hand, the previous lemma states that

$$\mathbb{C}(\mathfrak{A})[X_1, \ldots, X_{i-1}] \to \mathbb{C}(\mathfrak{A})[X_1, \ldots, X_n]/\mathfrak{I}^{\mathfrak{A}} \quad (3)$$

is integral; from this, Proposition 3.3.1 in [17] implies that it is actually a free module. Any basis of the latter is also a basis of (2); as a consequence, for j in $i, \ldots, n$, the characteristic polynomials of X_j in (2) or (3) are the same. Let P_j be the minimal polynomial of X_j in (2). The previous discussion implies that the characteristic polynomial χ_j of X_j in (2), and thus also P_j, are in $\mathbb{C}(\mathfrak{A})[X_1, \ldots, X_{i-1}, X_j]$ and monic in X_j.

By definition, χ_j is in $\mathfrak{I}^{\mathfrak{A}}$ and since there exists an integer k such that χ_j divides ${P_j}^k$ in $\mathbb{C}(\mathfrak{A})[X_1, \ldots, X_{i-1}][X_j]$, ${P_j}^k$ is in $\mathfrak{I}^{\mathfrak{A}}$. Since the latter ideal is radical, we conclude that P_j is in $\mathfrak{I}^{\mathfrak{A}}$. This implies that $\overline{P_j}$ is in $\mathfrak{I}^{\mathfrak{A}}$ as well.

Now, consider the polynomials $f^{\mathfrak{A}}, \partial f^{\mathfrak{A}}/\partial X_{i+1}, \ldots, \partial f^{\mathfrak{A}}/\partial X_n$ in $\mathbb{C}[\mathfrak{A}, X_1, \ldots, X_n]$, let $\mathfrak{W}$ be their zero-set, and let $\deg(\mathfrak{W})$ be its degree, in the sense of [19]. Proposition 1 in [27] implies that $\overline{P_j}$ has degree at most $\deg(\mathfrak{W})$. Since all polynomials defining $\mathfrak{W}$, seen in $\mathbb{C}[\mathfrak{A}, X_1, \ldots, X_n]$, have degree at most $2d$, the Bézout inequality of [19] gives $\deg(\overline{P_j}) \leq (2d)^{n-i+1} \leq (2d)^n$. □

Our next step is to give degree bounds on the coefficients appearing in the membership equality $\overline{P_j} \in \mathfrak{I}^{\mathfrak{A}}$. This is done using Rabinovicz's trick. Let T be a new variable; applying the Nullstellensatz in $\mathbb{C}(\mathfrak{A})[X_1, \ldots, X_n, T]$, and clearing denominators, we obtain the existence of α_j in $\mathbb{C}[\mathfrak{A}] - \{0\}$ and $C_{j,\ell}, B_j$ in $\mathbb{C}[\mathfrak{A}][X_1, \ldots, X_n][T]$, such that

$$\alpha_j = \sum_{\ell=1}^{n-i+1} C_{j,\ell} G_\ell + B_j(1 - \overline{P_j}T), \quad G_\ell \in \left\{ f^{\mathfrak{A}}, \frac{\partial f^{\mathfrak{A}}}{\partial X_{i+1}}, \ldots, \frac{\partial f^{\mathfrak{A}}}{\partial X_n} \right\}. \tag{4}$$

Let us then define

$$\Delta_i := \Delta_{i,1} \alpha_i \cdots \alpha_n D_i \cdots D_n,$$

where $\Delta_{i,1}$ was defined in Section 3.2 and for all j, α_j is as above and D_j is the leading coefficient of $\overline{P_j}$ with respect to X_j. Thus, Δ_i is a non-zero polynomial in $\mathbb{C}[\mathfrak{A}]$; we will estimate its degree below.

Lemma 4.3. *Suppose that $\boldsymbol{A} \in \mathbb{C}^{n \times n}$ does not cancel Δ_i. Then $f^{\boldsymbol{A}}$ satisfies $\mathbf{H}_i$.*

Proof. By assumption, $\Delta_{i,1}(\boldsymbol{A})$ is non-zero, so that $\boldsymbol{A}$ is invertible and $f^{\boldsymbol{A}}$ satisfies $\mathbf{H}_i(1)$. In particular, the ideal $\mathfrak{I}(i, f^{\boldsymbol{A}})$ is radical, and its zero-set $W(\pi_i, V(f^{\boldsymbol{A}}))$ is either empty or $(i-1)$-equidimensional. If it is empty, we are done.

Otherwise, for $j = i, \ldots, n$, evaluate all indeterminates in $\mathfrak{A}$ at the corresponding entries of $\boldsymbol{A}$ in (4). This gives us an equality in $\mathbb{C}[X_1, \ldots, X_n, T]$ of the form

$$a_j = \sum_{\ell=1}^{n-i+1} c_{j,\ell} g_\ell + b_j(1 - p_j T), \quad g_\ell \in \left\{ f^{\boldsymbol{A}}, \frac{\partial f^{\boldsymbol{A}}}{\partial X_{i+1}}, \ldots, \frac{\partial f^{\boldsymbol{A}}}{\partial X_n} \right\},$$

for a_j in $\mathbb{C}$, polynomials $c_{j,\ell}$ and b_j in $\mathbb{C}[X_1, \ldots, X_n, T]$ and p_j in $\mathbb{C}[X_1, \ldots, X_{i-1}, X_j]$. Since neither α_j nor D_j vanish at $\boldsymbol{A}$, a_j is non-zero and the leading coefficient of p_j in X_j is a non-zero constant.

The conclusion is now routine. Replace T by $1/p_j$ in the previous equality; after clearing denominators, this gives a membership equality of the form ${p_j}^k \in \mathfrak{I}(i, f^{\boldsymbol{A}})$, for some integer $k \geq 1$ (we cannot have $k = 0$, since we assumed that $W(\pi_i, V(f^{\boldsymbol{A}}))$ is not empty). Since $\mathfrak{I}(i, f^{\boldsymbol{A}})$ is radical, p_j is in $\mathfrak{I}(i, f^{\boldsymbol{A}})$. Repeating this for all j proves that $\mathbb{C}[X_1, \ldots, X_{i-1}] \to \mathbb{C}[X_1, \ldots, X_n]/\mathfrak{I}(i, f^{\boldsymbol{A}})$ is integral. □

To estimate the degree of Δ_i, what remains is to give an upper bound on the degree of $\alpha_i, \ldots, \alpha_n$. This will come as an application of the effective Nullstellensatz given in [10], for which we first need to determine degree bounds, separately in $\boldsymbol{X}, T$ and $\mathfrak{A}$, of the polynomials in the membership relationship:

$$\deg_{\boldsymbol{X},T} \left\{ f^{\mathfrak{A}}, \frac{\partial f^{\mathfrak{A}}}{\partial X_{i+1}}, \ldots, \frac{\partial f^{\mathfrak{A}}}{\partial X_n} \right\} \leq d;$$

$$\deg_{\mathfrak{A}} \left\{ f^{\mathfrak{A}}, \frac{\partial f^{\mathfrak{A}}}{\partial X_{i+1}}, \ldots, \frac{\partial f^{\mathfrak{A}}}{\partial X_n} \right\} \leq d;$$

$$\deg_{\boldsymbol{X},T}(1 - T\overline{P_j}) \leq (2d)^n + 1;$$

$$\deg_{\mathfrak{A}}(1 - T\overline{P_j}) \leq (2d)^n.$$

For each $j \in \{i, \ldots, n\}$, a direct application of [10, Theorem 0.5], gives $\deg(\alpha_j) \leq (n+1)d^n((2d)^n + 1)$; we will use the slightly less precise bound $\deg(\alpha_j) \leq 2n(2d)^{2n}$.

We saw in Section 3.2 that $\Delta_{i,1}$ has degree at most $2nd^{2n}$, and all D_j's have degree at most $(2d)^n$. This gives the upper bound

$$\deg(\Delta_i) \leq 2nd^{2n} + 2n^2(2d)^{2n} + n(2d)^n \leq 5n^2(2d)^{2n}.$$

This completes the proof of Theorem 2.1.

5 Proof of the main result

The following is our main algorithm; it expands on the sketch given in the introduction, by quantifying the various random choices.

In step 4, we use [31, Algorithm 2] to solve a square system. This subroutine is randomized; in order to guarantee a higher probability of success, we repeat the calculation k times, for a well-chosen parameter k.

This subroutine also requires that the input system be given by a straight-line program. We build it (at Step 3) in the straightforward manner already suggested in the introduction: given f, we can build a straight-line program that evaluates f in $O(d^n)$ operations, by computing all monomials of degree up to d, multiplying them by the corresponding coefficients in f, and adding results. To obtain a straight-line program for $f^{\boldsymbol{A}}$, we add $O(n^2)$ steps corresponding to the application of the change of variables $\boldsymbol{A}$. From this, we can compute the required partial derivatives of $f^{\boldsymbol{A}}$ for the same asymptotic cost [8]. Finally, we add the linear equations $X_1 - \sigma_1, \ldots, X_{i-1} - \sigma_{i-1}$; this gives Γ_i.

Algorithm 1: Main Algorithm

Input: $f \in \mathbb{Z}[X_1, \ldots, X_n]$ of degree at most d and height at most b, and $0 < \epsilon < 1$

Output: n zero-dimensional parameterizations, the union of whose zeros includes at least one point in each connected component of $V(f) \cap \mathbb{R}^n$, with probability of success at least $1 - \epsilon$.

1 Construct $S := \{1, 2, \ldots, \lceil 3\epsilon^{-1} 5n^3 (2d)^{2n} \rceil\}$ and $T := \{1, 2, \ldots, \lceil 3\epsilon^{-1} nd^{2n} \rceil\}$, and randomly choose $\boldsymbol{A} \in S^{n^2}$, and $\boldsymbol{\sigma} \in T^{n-1}$;

2 **for** $i \leftarrow 1$ **to** n **do**

3 Build a straight-line program Γ_i that computes the equations $\left\{ X_1 - \sigma_1, \ldots, X_{i-1} - \sigma_{i-1}, f^{\boldsymbol{A}}, \frac{\partial f^{\boldsymbol{A}}}{\partial X_{i+1}}, \ldots, \frac{\partial f^{\boldsymbol{A}}}{\partial X_n} \right\}$;

4 Run [31, Algorithm 2] $k \geq \lg(3n/\epsilon)$ times with input Γ_i;

5 Let $\mathscr{Q}_i$ be the highest cardinality zero-dimensional parameterization returned in step 4 ;

6 **return** $[\mathscr{Q}_1, \ldots, \mathscr{Q}_n]$.

If $f^{\boldsymbol{A}}$ satisfies $\mathbf{H}_i$, and $f^{\boldsymbol{A}}$ and $(\sigma_1, \ldots, \sigma_{i-1})$ satisfy $\mathbf{H}'_i$ for all i, then Theorem 2 in [28] establishes correctness.

Bit operation cost. The following lists the costs for each step of Algorithm 1:

(1) We defined $S := \{1, 2, \ldots, \lceil 3\epsilon^{-1} 5n^3 (2d)^{2n} \rceil\}$ and therefore the height of any $a_{i,j} \in S$ is at most

$$\log 3/\epsilon + \log(5n^3(2d)^{2n}) \in O^{\sim}(\log 1/\epsilon + n \log d).$$

Since $|T| < |S|$, the height of any $\sigma_j \in T$ is at most the same.

(3) After computing the partial derivatives, the height grows by at most another factor of $\log d$. Thus, all polynomials in the system considered at Step 3 have height $O^{\sim}(b + d\log 1/\epsilon + dn)$. All integer coefficients appearing in Γ_i satisfy the same bound.

(4) As a result, after applying [31, Algorithm 2] k times for each index i, with $k = O(\log n + \log 1/\epsilon)$, the total boolean cost of the algorithm is

$$O^{\sim}(d^{3n+1}(\log 1/\epsilon)(b + \log 1/\epsilon))$$

where the polynomials in the output have degree at most d^n, and height at most

$$O^{\sim}(d^{n+1}(b + \log 1/\epsilon)).$$

This proves the runtime estimate, as well as our bounds on the height of the output.

Probability of success. Let $\Delta_i \in \mathbb{C}[\mathfrak{U}]$ be the polynomials from Theorem 2.1. Denote by $\Delta := \prod_{i=1}^{n} \Delta_i$, and note that

$$\deg \Delta \leq \sum_{i=1}^{n} \deg \Delta_i \leq 5n^3(2d)^{2n}. \tag{5}$$

If $\boldsymbol{A} \in \mathbb{C}^{n\times n}$ does not cancel Δ, then $\boldsymbol{A}$ is invertible and $f^{\boldsymbol{A}}$ satisfies $\mathbf{H}_i$ for all $i \in \{1, \ldots, n\}$. Now, assuming that $\boldsymbol{A}$ is such a matrix, let $\Xi_i \in \mathbb{C}[S_1, \ldots, S_{i-1}]$ be the polynomials from Theorem 2.2 applied to $f^{\boldsymbol{A}}$. Denote by $\Xi := \prod_{i=1}^{n} \Xi_i$, and note that

$$\deg \Xi \leq \sum_{i=1}^{n} \deg \Xi_i \leq nd^{2n}. \tag{6}$$

If $\boldsymbol{\sigma} \in \mathbb{C}^{i-1}$ does not cancel Ξ, then $f^{\boldsymbol{A}}$ and $\boldsymbol{\sigma}$ satisfy $\mathbf{H}_i'$ for all $i \in \{1, \ldots, n\}$. As we argued above, the algorithm is guaranteed to succeed, as long as our call to Algorithm 2 in [31] succeeds. That latter reference establishes that by repeating the calculation k times, and keeping the output of highest degree among those k results, we succeed with probability at least $1 - (1/2)^k$. When Algorithm 2 does not succeed, it either returns a proper subset of the solutions, or FAIL. Note that Algorithm 2 is shown to succeed in a single run with probability at least $1 - 11/32$, and we bound the probability of success with $1 - 1/2$ for simplicity. Now, by construction of

$$S := \{1, 2, \ldots, \lceil 3\epsilon^{-1}5n^3(2d)^{2n}\rceil\}$$

and

$$T := \{1, 2, \ldots, \lceil 3\epsilon^{-1}nd^{2n}\rceil\},$$

where $\boldsymbol{A} \in S^{n^2}$ and $\boldsymbol{\sigma} \in T^{n-1}$ are randomly chosen, we have

$$\mathbb{P}[\Delta(\boldsymbol{A}) = 0] \leq \frac{\deg \Delta}{|S|} = \epsilon/3$$

and

$$\mathbb{P}[\Xi(\boldsymbol{\sigma}) = 0] \leq \frac{\deg \Xi}{|T|} = \epsilon/3.$$

Let $\mathscr{E}$ be the event that the parameterizations $[\mathscr{Q}_1, \ldots, \mathscr{Q}_n]$ returned in step 6 of Algorithm 1 are correct. Then, the probability of success is equal to

$$\mathbb{P}[\Delta(\boldsymbol{A}) \neq 0] \times \mathbb{P}[\Xi(\boldsymbol{\sigma}) \neq 0 \mid \Delta(\boldsymbol{A}) \neq 0] \times \mathbb{P}[\mathscr{E} \mid \Delta(\boldsymbol{A})\Xi(\boldsymbol{\sigma}) \neq 0].$$

Set $k = \lg(3n/\epsilon)$ so that

$$(1 - 2^{-k})^n = (1 - \epsilon/(3n))^n \geq 1 - \epsilon/3,$$

by Bernoulli's inequality. Therefore,

$$\begin{aligned}\mathbb{P}[\text{success}] &\geq (1 - \epsilon/3)(1 - \epsilon/3)\mathbb{P}[\mathscr{E} \mid \Delta(\boldsymbol{A})\Xi(\boldsymbol{\sigma}) \neq 0] \\ &\geq (1 - \epsilon/3)(1 - \epsilon/3)(1 - 2^{-k})^n \\ &\geq (1 - \epsilon/3)(1 - \epsilon/3)(1 - \epsilon/3) \\ &\geq 1 - \epsilon.\end{aligned}$$

This finishes the proof of our main theorem.

References

[1] M. Alonso, E. Becker, M.-F. Roy, and T. Wörmann. 1996. Zeroes, multiplicities and idempotents for zerodimensional systems. In *Algorithms in algebraic geometry and applications. Proceedings of MEGA'94 (Progress in Mathematics)*, Vol. 142. Birkhaüser, 1–15.

[2] B. Bank, M. Giusti, and J. Heintz. 2014. Point searching in real singular complete intersection varieties: Algorithms of intrinsic complexity. *Math. Comp.* 83 (2014), 873–897.

[3] B. Bank, M. Giusti, J. Heintz, L. Lehmann, and L.-M. Pardo. 2012. Algorithms of Intrinsic Complexity for Point Searching in Compact Real Singular Hypersurfaces. *Foundations of Computational Mathematics* 12 (2012), 75–122.

[4] B. Bank, M. Giusti, J. Heintz, and G. Mbakop. 1997. Polar Varieties and Efficient Real Equation Solving: The Hypersurface Case. *Journal of Complexity* 13, 1 (1997), 5–27.

[5] B. Bank, M. Giusti, J. Heintz, and G.-M. Mbakop. 2001. Polar varieties and efficient real elimination. *Mathematische Zeitschrift* 238, 1 (2001), 115–144.

[6] B. Bank, M. Giusti, J. Heintz, and L.-M. Pardo. 2005. Generalized polar varieties: geometry and algorithms. *Journal of Complexity* 21, 4 (2005), 377–412.

[7] S. Basu, R. Pollack, and M.-F. Roy. 2003. *Algorithms in Real Algebraic Geometry.* Algorithms and computation in mathematics, Vol. 10. Springer-Verlag.

[8] W. Baur and V. Strassen. 1983. The complexity of partial derivatives. *Theoret. Comput. Sci.* 22, 3 (1983), 317–330.

[9] J. Bochnak, M. Coste, and M.-F. Roy. 1998. *Real algebraic geometry.* Springer-Verlag.

[10] C. D'Andrea, T. Krick, and M. Sombra. 2013. Heights of varieties in multiprojective spaces and arithmetic Nullstellensatz. *Annales scientifiques de l'École Normale Supérieure* 46, 4 (Aug 2013), 549–627.

[11] M. Demazure. 2000. *Bifurcations and catastrophes: geometry of solutions to nonlinear problems.* Springer.

[12] D. Eisenbud. 1995. *Commutative Algebra with a View Toward Algebraic Geometry* (1st. ed.). Graduate Texts in Mathematics, Vol. 150. Springer-Verlag, New York.

[13] P. Gianni and T. Mora. 1989. Algebraic solution of systems of polynomial equations using Groebner bases. In *AAECC (LNCS)*, Vol. 356. Springer, 247–257.

[14] M. Giusti, K. Hägele, J. Heintz, J.-E. Morais, J.-L. Montaña, and L.-M. Pardo. 1997. Lower bounds for diophantine approximation. *J. of Pure and Applied Algebra* 117/118 (1997), 277–317.

[15] M. Giusti, J. Heintz, J.-E. Morais, J. Morgenstern, and L.-M. Pardo. 1998. Straight-line programs in geometric elimination theory. *Journal of Pure and Applied Algebra* 124 (1998), 101–146.

[16] M. Giusti, J. Heintz, J.-E. Morais, and L.-M. Pardo. 1995. When polynomial equation systems can be solved fast?. In *AAECC-11 (LNCS)*, Vol. 948. Springer, 205–231.

[17] M. Giusti, J. Heintz, and J. Sabia. 1993. On the efficiency of effective Nullstellensätze. *Computational Complexity* 3 (1993), 56–95.

[18] D. Grigoriev and N. Vorobjov. 1988. Solving Systems of Polynomial Inequalities in Subexponential Time. *J. Symbolic Comput.* 5 (1988), 37–64.

[19] J. Heintz. 1983. Definability and fast quantifier elimination in algebraically closed fields. *Theoretical Computer Science* 24, 3 (May 1983), 239–277.

[20] G. Jeronimo and J. Sabia. 2002. Effective equidimensional decomposition of affine varieties. *Journal of Pure and Applied Algebra* 169 (2002), 229–248.

[21] T. Krick, L.-M. Pardo, and M. Sombra. 2001. Sharp estimates for the arithmetic Nullstellensatz. *Duke Mathematical Journal* 109, 3 (2001), 521–598.

[22] L. Kronecker. 1882. Grundzüge einer arithmetischen Theorie der algebraischen Grössen. *Journal für die reine und angewandte Mathematik* 92 (1882), 1–122.

[23] F. Macaulay. 1916. *The Algebraic Theory of Modular Systems.* Cambridge University Press.

[24] D. Mumford. 1976. *Algebraic Geometry 1 : complex algebraic varieties.* Springer.

[25] R. Piene. 1978. Polar classes of singular varieties. *Annales Scientifiques de l'École Normale Supérieure* 11, 2 (1978), 247–276.

[26] F. Rouillier. 1999. Solving zero-dimensional systems through the Rational Univariate Representation. *Applicable Algebra in Engineering, Communication and Computing* 9, 5 (1999), 433–461.
[27] É. Schost. 2003. Computing Parametric Geometric Resolutions. *Applicable Algebra in Engineering, Communication and Computing* 5 (2003), 349–393.
[28] É. Schost and M. Safey El Din. 2003. Polar Varieties and Computation of one Point in each Connected Component of a Smooth Real Algebraic Set. In *ISSAC'03*. ACM, 224–231.
[29] É. Schost and M. Safey El Din. 2011. A baby steps/giant steps probabilistic algorithm for computing roadmaps in smooth bounded real hypersurface. *Discrete and Computational Geometry* 5 (2011), 181–220.
[30] É. Schost and M. Safey El Din. 2017. A nearly optimal algorithm for deciding connectivity queries in smooth and bounded real algebraic sets. *J. ACM* 63, 6 (Feb. 2017), 1–48.
[31] É. Schost and M. Safey El Din. 2018. Bit complexity for multi-homogeneous system solving. Application to polynomial minimization. *Journal of Symbolic Computation* 87 (May 2018), 176–206.
[32] É. Schost, B. Saugata, M-F Roy, and M. Safey El Din. 2014. A baby step-giant step roadmap algorithm for general algebraic sets,. *Foundations of Computational Mathematics* 14 (2014), 1117–1172.
[33] I. Shafarevich. 1977. *Basic Algebraic Geometry 1.* Springer Verlag.
[34] B. Teissier. 1988. Quelques points de l'histoire des variétés polaires, de Poncelet à nos jours. In *Sém. Annales Univ. Blaise Pascal*, Vol. 4.

The Fundamental Theorem of Tropical Partial Differential Algebraic Geometry

Sebastian Falkensteiner
Johannes Kepler University (RISC Hagenberg)
Linz, Austria

Cristhian Garay-López
Centro de Investigación en Matemáticas, A.C. (CIMAT)
Guanajuato, México

Mercedes Haiech
Université de Rennes 1, UMR 6625 (IRMAR)
Rennes, France

Marc Paul Noordman
Bernoulli Institute, University of Groningen
Groningen, The Netherlands

Zeinab Toghani
School of Mathematical Sciences, Queen Mary University of London
London, United Kingdom

François Boulier
Univ. Lille, CNRS, Centrale Lille, Inria, UMR 9189 - CRIStAL
Lille, France

ABSTRACT

Tropical Differential Algebraic Geometry considers difficult or even intractable problems in Differential Equations and tries to extract information on their solutions from a restricted structure of the input. The Fundamental Theorem of Tropical Differential Algebraic Geometry states that the support of solutions of systems of ordinary differential equations with formal power series coefficients over an uncountable algebraically closed field of characteristic zero can be obtained by solving a so-called tropicalized differential system. Tropicalized differential equations work on a completely different algebraic structure which may help in theoretical and computational questions. We show that the Fundamental Theorem can be extended to the case of systems of partial differential equations by introducing vertex sets of Newton polytopes.

CCS CONCEPTS

• **Computing methodologies** → *Symbolic and algebraic manipulation.*

KEYWORDS

Differential Algebra, Tropical Differential Algebraic Geometry, Power Series Solutions, Newton Polytope, Arc Spaces

1 INTRODUCTION

Given an algebraically closed field of characteristic zero K, we consider the partial differential ring $(R_{m,n}, D)$, where

$$R_{m,n} = K[[t_1, \ldots, t_m]]\{x_1, \ldots, x_n\}$$

and $D = (\frac{\partial}{\partial t_k} \,:\, k = 1, \ldots, m)$ for $n, m \geq 1$ (see Section 2 for definitions). Up to now, tropical differential algebra has been limited to the study of the relation between the set of solutions $\mathrm{Sol}(G) \subseteq K[[t]]^n$ of differential ideals G in $R_{1,n}$ and their corresponding *tropicalizations*, which are certain polynomials p with coefficients in a tropical semiring $\mathbb{T}_1 = (\mathbb{Z}_{\geq 0} \cup \{\infty\}, +, \min)$ and with a set of solutions $\mathrm{Sol}(p) \subseteq \mathcal{P}(\mathbb{Z}_{\geq 0})^n$, see [7] and [1]. These elements $S \in \mathrm{Sol}(p)$ can be found by looking at *evaluations* $p(S) \in \mathbb{T}_1$ where the usual tropical vanishing condition holds.

In this paper, we consider the case $m > 1$. On this account, we work with elements in $\mathbb{Z}^m_{\geq 0}$, which requires new techniques. We show that considering the Newton polytopes and their vertex sets is the appropriate method for formulating and proving our generalization of the Fundamental Theorem of Tropical Differential Algebraic Geometry. We remark that in the case of $m = 1$ the definitions and properties presented here coincide with the corresponding ones in [1] and therefore, this work can indeed be seen as a generalization.

The problem of finding power series solutions of systems of partial differential equations has been extensively studied in the literature, but is very limited in the general case. In fact, we know from [5, Theorem 4.11] that there is already no algorithm for deciding whether a given linear partial differential equation with polynomial coefficients has a solution or not. The Fundamental Theorem, as it is stated in here, helps to find necessary conditions for the support of possible solutions.

The structure of the paper is as follows. In Section 2 we cover the necessary material from partial differential algebra. In Section 3 we introduce the semiring of supports $\mathcal{P}(\mathbb{Z}^m_{\geq 0})$, the semiring of vertex sets $\mathbb{T}_m$ and the vertex homomorphism $\mathrm{Vert}\colon \mathcal{P}(\mathbb{Z}^m_{\geq 0}) \longrightarrow \mathbb{T}_m$. In Section 4 we introduce the support and the tropicalization maps. In Section 5 we define the set of tropical differential polynomials $\mathbb{T}_{m,n}$, the notion of tropical solutions for them, and the tropicalization morphism $\mathrm{trop}\colon R_{m,n} \to \mathbb{T}_{m,n}$. The main result is Theorem 6.1, which is proven in Section 6. The proof we give here differs essentially from the one in [1] for the case of $m = 1$. In Section 7 we give some examples to illustrate our results.

In the following we will use the conventions that for a set S we denote by $\mathcal{P}(S)$ its power set, and by K we denote an algebraically closed field of characteristic zero.

2 PARTIAL DIFFERENTIAL ALGEBRA

Here we recall the preliminaries for partial differential algebraic geometry. The reference book for differential algebra is [8].

A **partial differential ring** is a pair (R, D) consisting of a commutative ring R with unit and a set $D = \{\delta_1, \ldots, \delta_m\}$ of $m > 1$ **derivations** which act on R and are pairwise commutative. We

ISSAC '20, July 20–23, 2020, Kalamata, Greece

ACM ISBN 978-1-4503-7100-1/20/07…$15.00
https://doi.org/10.1145/3373207.3404040

denote by Θ the free commutative monoid generated by D. If $J = (j_1, \ldots, j_m)$ is an element of the monoid $\mathbb{Z}_{\geq 0}^m = (\mathbb{Z}_{\geq 0}^m, +, 0)$, we denote $\Theta(J) = \delta_1^{j_1} \cdots \delta_m^{j_m}$ the **derivative operator** defined by J. If φ is any element of R, then $\Theta(J)\varphi$ is the element of R obtained by application of the derivative operator $\Theta(J)$ on φ.

Let (R, D) be a partial differential ring with $R \supseteq \mathbb{Q}$ and $x_1, \ldots, x_n$ be n **differential indeterminates**. The monoid Θ acts on the differential indeterminates, giving the infinite set of the **derivatives** which are denoted by $x_{i,J}$ with $1 \leq i \leq n$ and $J \in \mathbb{Z}_{\geq 0}^m$. Given any $1 \leq k \leq m$ and any derivative $x_{i,J}$, the action of δ_k on $x_{i,J}$ is defined by $\delta_k(x_{i,J}) = x_{i,J+e_k}$ where e_k is the m-dimensional vector whose k-th coordinate is 1 and all other coordinates are zero. One denotes $R\{x_1, \ldots, x_n\}$ the ring of the polynomials, with coefficients in R, the indeterminates of which are the derivatives. More formally, $R\{x_1, \ldots, x_n\}$ consists of all R-linear combinations of differential monomials, where a differential monomial in n independent variables of order less than or equal to r is an expression of the form

$$E_M := \prod_{\substack{1 \leq i \leq n \\ ||J||_\infty \leq r}} x_{i,J}^{M_{i,J}} \tag{1}$$

where $J = (j_1, \ldots, j_m) \in \mathbb{Z}_{\geq 0}^m$, $||J||_\infty := \max_i\{j_i\} = \max(J)$ and $M = (M_{i,J}) \in (\mathbb{Z}_{\geq 0})^{n \times (r+1)^m}$.

The pair $(R\{x_1, \ldots, x_n\}, D)$ then constitutes a **differential polynomial ring**. A differential polynomial $P \in R\{x_1, \ldots, x_n\}$ induces an evaluation map from R^n to R given by

$$P \colon R^n \to R, \quad (\varphi_1, \ldots, \varphi_n) \mapsto P|_{x_{i,J} = \Theta(J)\varphi_i},$$

where $P|_{x_{i,J} = \Theta(J)\varphi_i}$ is the element of R obtained by substituting $\Theta(J)\varphi_i$ for $x_{i,J}$.

A **zero** or **solution** of $P \in R\{x_1, \ldots, x_n\}$ is an n-tuple $\varphi = (\varphi_1, \ldots, \varphi_n) \in R^n$ such that $P(\varphi) = 0$. An n-tuple $\varphi \in R^n$ is a solution of a system of differential polynomials $\Sigma \subseteq R\{x_1, \ldots, x_n\}$ if it is a solution of every element of Σ. We denote by $\mathrm{Sol}(\Sigma)$ the solution set of the system Σ.

A **differential ideal** of $R\{x_1, \ldots, x_n\}$ is an ideal of that ring which is stable under the action of Θ. A differential ideal is said to be **perfect** if it is equal to its radical. If $\Sigma \subseteq R\{x_1, \ldots, x_n\}$, one denotes by $[\Sigma]$ the **differential ideal generated by** Σ and by $\{\Sigma\}$ the **perfect differential ideal generated by** Σ, which is defined as the intersection of all perfect differential ideals containing Σ.

For $m, n \geq 1$, we will denote by R_m the partial differential ring

$$(K[[t_1, \ldots, t_m]], D)$$

where $D = \{\frac{\partial}{\partial t_1}, \ldots, \frac{\partial}{\partial t_m}\}$, and by $R_{m,n}$ the partial differential ring $(R_m\{x_1, \ldots, x_n\}, D)$. The proof of the following proposition can be found in [3].

Proposition 2.1. *For any $\Sigma \subseteq R_{m,n}$, there exists a finite subset Φ of Σ such that $\mathrm{Sol}(\Sigma) = \mathrm{Sol}(\Phi)$.*

3 THE SEMIRINGS OF SUPPORTS AND VERTEX SETS

In this part we introduce and give some properties on our main idempotent semirings, namely the semiring of supports $\mathcal{P}(\mathbb{Z}_{\geq 0}^m)$, the semiring of vertex sets $\mathbb{T}_m$ and the map $\mathrm{Vert} \colon \mathcal{P}(\mathbb{Z}_{\geq 0}^m) \to \mathbb{T}_m$ which is a homomorphism of semirings.

Recall that a commutative semiring S is a tuple $(S, +, \times, 0, 1)$ such that $(S, +, 0)$ and $(S, \times, 1)$ are commutative monoids and additionally, for all $a, b, c \in S$ it holds that

(1) $a \times (b + c) = a \times b + a \times c$;
(2) $0 \times a = 0$.

A semiring is called **idempotent** if $a + a = a$ for all $a \in S$. A map $f \colon S_1 \longrightarrow S_2$ between semirings is a morphism if it induces morphisms at the level of monoids.

For $m \geq 1$, we denote by $\mathcal{P}(\mathbb{Z}_{\geq 0}^m)$ the idempotent semiring whose elements are the subsets of $\mathbb{Z}_{\geq 0}^m$ equipped with the union $X \cup Y$ as sum and the Minkowski sum $X + Y = \{x + y : x \in X, y \in Y\}$ as product. We call it the **semiring of supports**. For $n \in \mathbb{Z}_{\geq 1}$ and $X \in \mathcal{P}(\mathbb{Z}_{\geq 0}^m)$, the notation nX will indicate $\underbrace{X + \cdots + X}_{n \text{ times}}$. By convention we set $0X = \{(0, \ldots, 0)\}$.

We define the **Newton polytope** $\mathcal{N}(X) \subseteq \mathbb{R}_{\geq 0}^m$ of $X \in \mathcal{P}(\mathbb{Z}_{\geq 0}^m)$ as the convex hull of $X + \mathbb{Z}_{\geq 0}^m$. We call $x \in X$ a **vertex** if $x \notin \mathcal{N}(X \setminus \{x\})$, and we denote by $\mathrm{Vert}\, X$ the set of vertices of X.

Lemma 3.1. *Let $S, T \in \mathcal{P}(\mathbb{Z}_{\geq 0}^m)$ such that $\mathcal{N}(S) = \mathcal{N}(T)$. Then* $\mathrm{Vert}\, S = \mathrm{Vert}\, T$.

Proof. Let $s \in \mathrm{Vert}\, S$ and we assume that $s \in \mathcal{N}(T \setminus \{s\})$. Then there are $t_i \in T \setminus \{s\}$, $w_i \in \mathbb{Z}_{\geq 0}^m$ and positive $\lambda_i \in \mathbb{R}$ adding up to 1 such that

$$s = \sum_i \lambda_i (t_i + w_i).$$

Since $t_i \in \mathcal{N}(S)$, we can write the t_i as

$$t_i = \sum_j \mu_{i,j}(s_{i,j} + z_{i,j}),$$

where $s_{i,j} \in S$, $z_{i,j} \in \mathbb{Z}_{\geq 0}^m$ and $\mu_{i,j} \in \mathbb{R}$ are positive and adding up to 1. Thus,

$$s = \sum_{i,j} \lambda_i \mu_{i,j}(s_{i,j} + z_{i,j} + w_i) = \sum_{i,j} \lambda_i \mu_{i,j} s_{i,j} + v,$$

where v is a vector with non-negative coordinates. By excluding in the sum those summands $s_{i,j}$ which are equal to s, we obtain

$$s = cs + \sum_{\substack{i,j \\ s_{i,j} \neq s}} \lambda_i \mu_{i,j} s_{i,j} + v$$

where $c = \sum_{i,j : s_{i,j} = s} \lambda_i \mu_{i,j} \in [0, 1]$. If $c < 1$ we can solve the equation above for s to get

$$s = \sum_{\substack{i,j \\ s_{i,j} \neq s}} \frac{\lambda_i \mu_{i,j}}{1 - c} s_{i,j} + \frac{v}{1 - c}.$$

The coefficients for the $s_{i,j}$ are positive and sum to 1, so the summation in the right hand side gives an element of $\mathcal{N}(S \setminus \{s\})$. Since $\mathcal{N}(S \setminus \{s\})$ is closed under adding elements of $\mathbb{R}_{\geq 0}^m$, and the coordinates of $v/(1 - c)$ are non-negative, we then find that $s \in \mathcal{N}(S \setminus \{s\})$ in contradicting to the assumption that s is a vertex of S. If $c = 1$, then all $s_{i,j}$ are equal to s and we get $s = s + v$. Therefore, $v = 0$

and $t_i = s$ for each i, and in particular $s \in T \setminus \{s\}$, which is a contradiction. So we conclude that $s \notin \mathcal{N}(T \setminus \{s\})$ and s is a vertex of T. □

Lemma 3.2. *Let $X \in \mathcal{P}(\mathbb{Z}^m_{\geq 0})$. Then $\mathcal{N}(\mathrm{Vert}\, X) = \mathcal{N}(X)$.*

Proof. By Dickson's lemma [4, chap. 2, Thm 5], there is a finite subset $S \subseteq X$ with $X \subseteq S + \mathbb{Z}^m_{\geq 0}$. For such S, it holds that $\mathcal{N}(X) = \mathcal{N}(S)$ and by Lemma 3.1, we get $\mathrm{Vert}\, X = \mathrm{Vert}\, S$. Therefore, replacing X by S, we may assume that X is finite.

We proceed by induction on $\#X$. Indeed, if $X = \emptyset$, the statement is obvious. Let X be an arbitrary finite set. If every element of X is a vertex of X, then $\mathcal{N}(X) = \mathcal{N}(\mathrm{Vert}\, X)$ is trivially true. Else, take $x \in X \setminus \mathrm{Vert}\, X$ and let $Y = X \setminus \{x\}$. Then $\mathcal{N}(X) = \mathcal{N}(Y)$ by definition, so applying Lemma 3.1 again we obtain $\mathrm{Vert}\, X = \mathrm{Vert}\, Y$. Since $\#Y < \#X$, we may apply the induction hypothesis to Y, and get that $\mathcal{N}(X) = \mathcal{N}(Y) = \mathcal{N}(\mathrm{Vert}\, Y) = \mathcal{N}(\mathrm{Vert}\, X)$. □

Corollary 3.3. *For $X, Y \in \mathcal{P}(\mathbb{Z}^m_{\geq 0})$ we have $\mathrm{Vert}\, X = \mathrm{Vert}\, Y$ if and only if $\mathcal{N}(X) = \mathcal{N}(Y)$.*

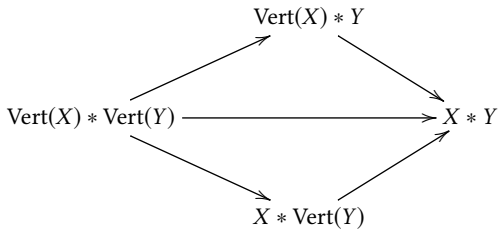
Lemma 3.4. *For $X, Y \in \mathcal{P}(\mathbb{Z}^m_{\geq 0})$, we have*

$$\begin{aligned} \mathrm{Vert}(\mathrm{Vert}(X) \cup \mathrm{Vert}(Y)) &= \mathrm{Vert}(\mathrm{Vert}(X) \cup Y) \\ &= \mathrm{Vert}(X \cup \mathrm{Vert}(Y)) \\ &= \mathrm{Vert}(X \cup Y) \end{aligned}$$

and

$$\begin{aligned} \mathrm{Vert}(\mathrm{Vert}(X) + \mathrm{Vert}(Y)) &= \mathrm{Vert}(\mathrm{Vert}(X) + Y) \\ &= \mathrm{Vert}(X + \mathrm{Vert}(Y)) \\ &= \mathrm{Vert}(X + Y). \end{aligned}$$

Proof. Let $*$ be either $\cup$ or $+$. We have the following diagram of inclusions



We show that these four sets generate the same Newton polytope. For this, it is enough to show that $X * Y \subseteq \mathcal{N}(\mathrm{Vert}(X) * \mathrm{Vert}(Y))$.

For $* = \cup$, we have $X \subseteq \mathcal{N}(\mathrm{Vert}\, X) \subseteq \mathcal{N}(\mathrm{Vert}(X) \cup \mathrm{Vert}(Y))$ and similarly $Y \subseteq \mathcal{N}(\mathrm{Vert}(X) \cup \mathrm{Vert}(Y))$. Hence, $X \cup Y \subseteq \mathcal{N}(\mathrm{Vert}(X) \cup \mathrm{Vert}(Y))$

Now suppose that $* = +$. Let $t \in X + Y$, and write $t = x + y$ with $x \in X$ and $y \in Y$. Using the inclusions $X \subseteq \mathcal{N}(\mathrm{Vert}\, X)$ and $Y \subseteq \mathcal{N}(\mathrm{Vert}\, Y)$, there are $x_i \in \mathrm{Vert}(X)$, $y_j \in \mathrm{Vert}(Y)$, $u_i, v_j \in \mathbb{Z}^m_{\geq 0}$ and $\alpha_i, \beta_j \in \mathbb{R}_{\geq 0}$ satisfying $\sum_i \alpha_i = 1$ and $\sum_j \beta_j = 1$ such that

$$t = \sum_i \alpha_i (x_i + u_i) + \sum_j \beta_j (y_j + v_j).$$

Rewriting this gives

$$t = \sum_{i,j} \alpha_i \beta_j (x_i + y_j + u_i + v_j).$$

For each pair i, j, the expression between parentheses is an element of $\mathrm{Vert}(X) + \mathrm{Vert}(Y) + \mathbb{Z}_{\geq 0}$ and the coefficients are non-negative and sum up to 1. This shows that $t \in \mathcal{N}(\mathrm{Vert}(X) + \mathrm{Vert}(Y))$, which ends the proof of the inclusions. □

Example 3.5. An element $X \in \mathcal{P}(\mathbb{Z}^m_{\geq 0})$ generates a monomial ideal which contains a unique minimal basis $B(X)$ (see e.g. [4]). In general, $\mathrm{Vert}(X) \subset B(X)$ and this inclusion may be strict. Consider the set $X = \{A_1 = (1,4), A_2 = (2,3), A_3 = (3,3), A_4 = (4,1)\} \subseteq \mathbb{Z}^2_{\geq 0}$. The Newton polytope $\mathcal{N}(X)$ can be visualized as in Figure 1 and $\mathrm{Vert}(X) = \{A_1, A_4\}$ which is a strict subset of $B(X) = \{A_1, A_2, A_4\}$.

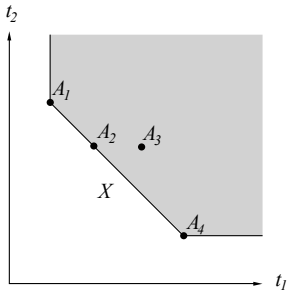


Figure 1: The Newton polytope of X. The vertex set of X is $\{A_1, A_4\}$.

We deduce from Corollary 3.3 that the map $\mathrm{Vert}\colon \mathcal{P}(\mathbb{Z}^m_{\geq 0}) \longrightarrow \mathcal{P}(\mathbb{Z}^m_{\geq 0})$ is a projection operator in the sense that $\mathrm{Vert}^2 = \mathrm{Vert}$.

Definition 3.6. We denote by $\mathbb{T}_m$ the image of the operator Vert, and call its elements **vertex sets**. For $S, T \in \mathbb{T}_m$, we define

$$S \oplus T = \mathrm{Vert}(S \cup T) \quad \text{and} \quad S \odot T = \mathrm{Vert}(S + T).$$

Corollary 3.7. *The set $(\mathbb{T}_m, \oplus, \odot)$ is a commutative idempotent semiring, with the zero element $\emptyset$ and the unit element $\{(0, \ldots, 0)\}$.*

Proof. The only things to check are associativity of $\oplus$, associativity of $\odot$ and the distributive property. The associativity of $\oplus$ and $\odot$ follows from the equalities

$$S \oplus (T \oplus U) = \mathrm{Vert}(S \cup T \cup U) = (S \oplus T) \oplus U$$

and

$$S \odot (T \odot U) = \mathrm{Vert}(S + T + U) = (S \odot T) \odot U$$

which are consequences of Lemma 3.4. The distributivity follows from

$$S \odot (T \oplus U) = \mathrm{Vert}((S+T) \cup U) = \mathrm{Vert}((S+T) \cup (S+U)) = (S \odot T) \oplus (S \odot U).$$

□

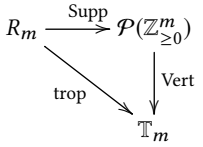
Corollary 3.8. *The map* Vert *is a homomorphism of semirings.*

Proof. Follows directly from Lemma 3.4 and Corollary 3.7. □

4 THE SUPPORT MAP AND THE TROPICALIZATION MAP

We consider the differential ring R_m from Section 2, and the semirings $\mathcal{P}(\mathbb{Z}^m_{\geq 0})$, $\mathbb{T}_m$ from Section 3. In this part we introduce the support and the tropicalization maps, which are related by the following

commutative diagram



If $J = (j_1, \ldots, j_m)$ is an element of $\mathbb{Z}^m_{\geq 0}$, we will denote by t^J the monomial $t_1^{j_1} \cdots t_m^{j_m}$. An element of R_m is of the form $\varphi = \sum_{J \in \mathbb{Z}^m_{\geq 0}} a_J t^J$ with $a_J \in K$.

Definition 4.1. The **support** of $\varphi = \sum a_J t^J \in R_m$ is defined as

$$\mathrm{Supp}(\varphi) = \{J \in \mathbb{Z}^m_{\geq 0} \mid a_J \neq 0\}.$$

For a fixed integer n, the map which sends $\varphi = (\varphi_1, \ldots, \varphi_n) \in R_m^n$ to $\mathrm{Supp}(\varphi) = (\mathrm{Supp}(\varphi_1), \ldots, \mathrm{Supp}(\varphi_n)) \in \mathcal{P}(\mathbb{Z}^m_{\geq 0})^n$ will also be denoted by Supp. The **set of supports** of a subset $T \subseteq R_m^n$ is its image under the map Supp:

$$\mathrm{Supp}(T) = \{\mathrm{Supp}(\varphi) \mid \varphi \in T\} \subseteq \mathcal{P}(\mathbb{Z}^m_{\geq 0})^n$$

Definition 4.2. The map that sends each series in R_m to the vertex set of its support is called the **tropicalization** map

$$\begin{array}{rccc} \mathrm{trop}\colon & R_m & \to & \mathbb{T}_m \\ & \varphi & \mapsto & \mathrm{Vert}(\mathrm{Supp}(\varphi)) \end{array}$$

Lemma 4.3. *The tropicalization map is a non-degenerate valuation in the sense of [6, Definition 2.5.1]. This is, it satisfies*

(1) $\mathrm{trop}(0) = \emptyset$, $\mathrm{trop}(\pm 1) = \{(0, \ldots, 0)\}$,
(2) $\mathrm{trop}(\varphi \cdot \psi) = \mathrm{trop}(\varphi) \odot \mathrm{trop}(\psi)$,
(3) $\mathrm{trop}(\varphi + \psi) \oplus \mathrm{trop}(\varphi) \oplus \mathrm{trop}(\psi) = \mathrm{trop}(\varphi) \oplus \mathrm{trop}(\psi)$,
(4) $\mathrm{trop}(\varphi) = \emptyset$ *implies that* $\varphi = 0$.

Proof. The first point is clear. For the second point, note that the Newton polytope has the well-known homomorphism-type property (see [9, Lemma 2.2])

$$\mathcal{N}(\mathrm{Supp}(\varphi \cdot \psi)) = \mathcal{N}(\mathrm{Supp}(\varphi)) + \mathcal{N}(\mathrm{Supp}(\psi)) = \mathcal{N}(\mathrm{Supp}(\varphi) + \mathrm{Supp}(\psi)).$$

Hence, the vertices of the left hand side coincide with the vertices of the right hand side. This gives $\mathrm{trop}(\varphi \cdot \psi) = \mathrm{Vert}(\mathcal{N}(\mathrm{Supp}(\varphi) + \mathrm{Supp}(\psi)))$. That this is equal to $\mathrm{trop}(\varphi) \odot \mathrm{trop}(\psi)$ follows from Lemma 3.4. The third point follows from the observation that $\mathrm{Supp}(\varphi + \psi) \subseteq \mathrm{Supp}(\varphi) \cup \mathrm{Supp}(\psi)$ and Corollary 3.8. The last point follows from the fact that the empty set is the only set with empty Newton polytope. □

Definition 4.4. For $J = (j_1, \ldots, j_m) \in \mathbb{Z}^m_{\geq 0}$, we define the **tropical derivative operator** $\Theta_{\mathrm{trop}}(J)\colon \mathcal{P}(\mathbb{Z}^m_{\geq 0}) \to \mathcal{P}(\mathbb{Z}^m_{\geq 0})$ as

$$\Theta_{\mathrm{trop}}(J)T := \left\{ (t_1 - j_1, \ldots, t_m - j_m) \;\middle|\; \begin{array}{c} (t_1, \ldots, t_m) \in T, \\ t_i - j_i \geq 0 \text{ for all } i \end{array} \right\}.$$

For example, if T is the grey part in Figure 2 left and $J = (1, 2)$, then informally $\Theta_{\mathrm{trop}}(J)T$ is a translation of T by the vector $-J$ and then keeping only the non-negative part. It is represented by the grey part in Figure 2 right.

Since K is of characteristic zero, for all $\varphi \in R_m$ and $J \in \mathbb{Z}^m_{\geq 0}$, we have

$$\mathrm{Supp}(\Theta(J)\varphi) = \Theta_{\mathrm{trop}}(J)\mathrm{Supp}(\varphi). \tag{2}$$

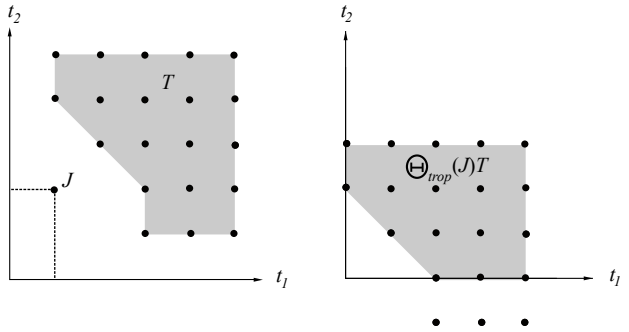


Figure 2: The operator $\Theta_{\mathrm{trop}}(J)$ for $J = (1, 2)$ applied to T.

Consider a differential monomial E_M as in (1) and $S = (S_1, \ldots, S_n) \in \mathcal{P}(\mathbb{Z}^m_{\geq 0})^n$. We can now define the evaluation of E_M at S as

$$E_M(S) = \sum_{\substack{1 \leq i \leq n \\ ||J||_\infty \leq r}} M_{i,J} \Theta_{\mathrm{trop}}(J) S_i \in \mathcal{P}(\mathbb{Z}^m_{\geq 0}). \tag{3}$$

Lemma 4.5. *Given $\varphi = (\varphi_1, \ldots, \varphi_n) \in R_m^n$ and a differential monomial E_M, we have* $\mathrm{trop}(E_M(\varphi)) = \mathrm{Vert}(E_M(\mathrm{Supp}(\varphi)))$

Proof. By applying Vert to equation (2), we have

$$\mathrm{trop}(\Theta(J)\varphi_i) = \mathrm{Vert}(\Theta_{\mathrm{trop}}(J)\mathrm{Supp}(\varphi_i)). \tag{4}$$

Using the multiplicativity of trop, equation (4) and Corollary 3.8, we obtain

$$\begin{aligned} \mathrm{trop}(E_M(\varphi)) &= \bigodot_{i,J} \mathrm{trop}(\Theta(J)\varphi_i)^{\odot M_{i,J}} \\ &= \bigodot_{i,J} \mathrm{Vert}(\Theta_{\mathrm{trop}}(J)\mathrm{Supp}(\varphi_i))^{\odot M_{i,J}} \\ &= \mathrm{Vert}(E_M(\mathrm{Supp}(\varphi))). \end{aligned}$$ □

Remark 4.6. If $P = \sum_M \alpha_M E_M \in R_{m,n}$ and $\varphi = (\varphi_1, \ldots, \varphi_n) \in R_m^n$, then we can consider the upper support $US(P, \varphi)$ of P at φ as

$$US(P, \varphi) = \bigcup_M (\mathrm{Supp}(\alpha_M) + \mathrm{Supp}(E_M(\varphi))) \in \mathcal{P}(\mathbb{Z}^m_{\geq 0}).$$

We now compute the vertex set of $US(P, \varphi)$ by applying the operation Vert and Corollary 3.8 to the above expression to find

$$\begin{aligned} \mathrm{Vert}\Big(US(P, \varphi)\Big) &= \bigoplus_M \mathrm{trop}(\alpha_M) \odot \mathrm{trop}(E_M(\varphi)) \\ &= \bigoplus_M \mathrm{trop}(\alpha_M) \odot \mathrm{Vert}(E_M(\mathrm{Supp}(\varphi))), \end{aligned}$$

since $\mathrm{trop}(E_M(\varphi)) = \mathrm{Vert}(E_M(\mathrm{Supp}(\varphi)))$ by Lemma 4.5. This motivates the definition of tropical differential polynomials in the next section.

5 TROPICAL DIFFERENTIAL POLYNOMIALS

In this section we define the set of tropical differential polynomials $\mathbb{T}_{m,n}$ and the corresponding tropicalization morphism $\mathrm{trop}\colon R_{m,n} \to \mathbb{T}_{m,n}$. Let us remark that in the case of $m = 1$ the definitions and properties presented here coincide with the corresponding ones in [1]. Moreover, later in Section 7 we illustrate in Example 7.2 the reason for the particular definitions given here.

Definition 5.1. For a set $S \in \mathcal{P}(\mathbb{Z}^m_{\geq 0})$ and a multi-index $J \in \mathbb{Z}^m_{\geq 0}$ we define

$$\mathrm{Val}_J(S) = \mathrm{Vert}(\Theta_{\mathrm{trop}}(J)S).$$

Note that for $\varphi \in R_m$ and any multi-index J this means that

$$\mathrm{Val}_J(\mathrm{Supp}(\varphi)) = \mathrm{trop}(\Theta(J)\varphi).$$

In particular, $\mathrm{Val}_J(\mathrm{Supp}(\varphi)) = \emptyset$ if and only if $\Theta(J)\varphi = 0$. It follows from Corollary 3.8 that

$$\mathrm{Vert}(E_M(S)) = \bigodot_{\substack{1\leq i\leq n\\ ||J||_\infty \leq r}} \mathrm{Val}_J(S_i)^{\odot M_{i,J}}.$$

Definition 5.2. A **tropical differential monomial** in the variables $x_1, \ldots, x_n$ of order less or equal to r is an expression of the form

$$\epsilon_M = \bigodot_{\substack{1\leq i\leq n\\ ||J||_\infty \leq r}} x_{i,J}^{\odot M_{i,J}}$$

where $M = (M_{i,J}) \in (\mathbb{Z}_{\geq 0})^{n\times(r+1)^m}$.

A tropical differential monomial ϵ_M induces an evaluation map from $\mathcal{P}(\mathbb{Z}^m_{\geq 0})^n$ to $\mathbb{T}_m$ by

$$\epsilon_M(S_1, \ldots, S_n) = \mathrm{Vert}(E_M(S)) = \bigodot_{i,J} \mathrm{Val}_J(S_i)^{\odot M_{i,J}}$$

where $\mathrm{Val}_J(S_i)$ is given in Definition 5.1 and $E_M(S)$ as in (3). Let us recall that, by Corollary 3.7, we can also write

$$\epsilon_M(S_1, \ldots, S_n) = \mathrm{Vert}\left(\sum_{i,J} \mathrm{Val}_J(S_i)^{\odot M_{i,J}}\right).$$

Definition 5.3. A **tropical differential polynomial** in the variables $x_1, \ldots, x_n$ of order less or equal to r is an expression of the form

$$p = p(x_1, \ldots, x_n) = \bigoplus_{M\in\Delta} a_M \odot \epsilon_M$$

where $a_M \in \mathbb{T}_m$, $a_M \neq \emptyset$ and Δ is a finite subset of $(\mathbb{Z}_{\geq 0})^{n\times(r+1)^m}$. We denote by $\mathbb{T}_{m,n} = \mathbb{T}_m\{x_1, \ldots, x_n\}$ the set of tropical differential polynomials.

A tropical differential polynomial p as in Definition 5.3 induces a map from $\mathcal{P}(\mathbb{Z}^m_{\geq 0})^n$ to $\mathbb{T}_m$ by

$$p(S) = \bigoplus_{M\in\Delta} a_M \odot \epsilon_M(S) = \mathrm{Vert}\Big(\bigcup_{M\in\Delta} (a_M + \epsilon_M(S))\Big)$$

The second equality follows again from Corollary 3.8. A differential polynomial $P \in R_{m,n}$ of order at most r is of the form

$$P = \sum_{M\in\Delta} \alpha_M E_M$$

where Δ is a finite subset of $(\mathbb{Z}_{\geq 0})^{n\times(r+1)^m}$, $\alpha_M \in K[[t_1, \ldots, t_m]]$ and E_M is a differential monomial as in (1). Then the **tropicalization** of P is defined as

$$\mathrm{trop}(P) = \bigoplus_{M\in\Delta} \mathrm{trop}(\alpha_M) \odot \epsilon_M \in \mathbb{T}_{m,n}$$

where ϵ_M is the tropical differential monomial corresponding to E_M.

Definition 5.4. Let $G \subseteq R_{m,n}$ be a differential ideal. Its **tropicalization** $\mathrm{trop}(G)$ is the set of tropical differential polynomials $\{\mathrm{trop}(P) \mid P \in G\} \subseteq \mathbb{T}_{m,n}$.

Lemma 5.5. *Given a differential monomial E_M and $\varphi = (\varphi_1, \ldots, \varphi_n) \in K[[t_1, \ldots, t_m]]^n$, we have that*

$$\mathrm{trop}(E_M(\varphi)) = \epsilon_M(\mathrm{Supp}(\varphi)).$$

Proof. Follows from notations and Lemma 4.5. □

The following tropical vanishing condition is a natural generalization of the case $m = 1$, but now the evaluation $p(S)$ consists of a vertex set instead of a single minimum.

Definition 5.6. Let $p = \bigoplus_{M\in\Delta} a_M \odot \epsilon_M$ be a tropical differential polynomial. An n-tuple $S \in \mathcal{P}(\mathbb{Z}^m_{\geq 0})^n$ is said to be a **solution** of p if for every $J \in p(S)$ there exists $M_1, M_2 \in \Delta$ with $M_1 \neq M_2$ such that $J \in a_{M_1} \odot \epsilon_{M_1}(S)$ and $J \in a_{M_2} \odot \epsilon_{M_2}(S)$. Note that in the particular case of $p(S) = \emptyset$, S is a solution of p.

For a family of differential polynomials $H \subseteq \mathbb{T}_{m,n}$, S is called a **solution** of H if and only if S is a solution of every tropical polynomial in H. The set of solutions of H will be denoted by $\mathrm{Sol}(H)$.

Proposition 5.7. *Let G be a differential ideal in the ring of differential polynomials $R_{m,n}$. If $\varphi \in \mathrm{Sol}(G)$, then $\mathrm{Supp}(\varphi) \in \mathrm{Sol}(\mathrm{trop}(G))$.*

Proof. Let φ be a solution of G and $S = \mathrm{Supp}(\varphi)$. Let $P = \sum_{M\in\Delta} \alpha_M E_M \in G$ and $p = \mathrm{trop}(P) = \bigoplus_{M\in\Delta} a_M \odot \epsilon_M$, where $a_M = \mathrm{trop}(\alpha_M)$. We need to show that S is a solution of p. Let $J \in p(S)$ be arbitrary. By the definition of $\oplus$, there is an index M_1 such that

$$J \in a_{M_1} \odot \epsilon_{M_1}(S).$$

Hence, by Lemma 5.5, and multiplicative property of trop Lemma 4.3

$$J \in \mathrm{Vert}(\mathrm{Supp}(\alpha_{M_1} E_{M_1}(\varphi))).$$

Since $P(\varphi) = 0$, there is another index $M_2 \neq M_1$ such that

$$J \in \mathrm{Supp}(\alpha_{M_2} E_{M_2}(\varphi)),$$

because otherwise there would not be cancellation. Since J is a vertex of $p(S)$, it follows that J is a vertex of every subset of $\mathcal{N}(p(S))$ containing J and in particular of $\mathcal{N}(\mathrm{Supp}(\alpha_{M_2} E_{M_2}(\varphi)))$. Therefore,

$$J \in a_{M_2} \odot \epsilon_{M_2}(S)$$

and because J and P were chosen arbitrary, S is a solution of G. □

6 THE FUNDAMENTAL THEOREM

Let $G \subset R_{m,n}$ be a differential ideal. Then Proposition 5.7 implies that $\mathrm{Supp}(\mathrm{Sol}(G)) \subseteq \mathrm{Sol}(\mathrm{trop}(G))$. The main result of this paper is to show that the reverse inclusion holds as well if the base field K is uncountable.

Theorem 6.1 (Fundamental Theorem)**.** *Let K be an uncountable, algebraically closed field of characteristic zero. Let G be a differential ideal in the ring $R_{m,n}$. Then*

$$\mathrm{Supp}(\mathrm{Sol}(G)) = \mathrm{Sol}(\mathrm{trop}(G)).$$

The proof of the Fundamental Theorem will take the rest of the section and is split into several parts. First let us introduce some notations. If $J = (j_1, \ldots, j_m)$ is an element of $\mathbb{Z}^m_{\geq 0}$, we define by $J!$ the component-wise product $j_1! \cdots j_m!$. The bijection between $K^{\mathbb{Z}^m_{\geq 0}}$ and R_m given by

$$\begin{array}{rccc} \psi: & K^{\mathbb{Z}^m_{\geq 0}} & \to & R_m \\ & \underline{a} = (a_J)_{J\in\mathbb{Z}^m_{\geq 0}} & \mapsto & \displaystyle\sum_{J\in\mathbb{Z}^m_{\geq 0}} \frac{1}{J!} a_J t^J \end{array}$$

allows us to identify points of R_m with points of $K^{\mathbb{Z}^m_{\geq 0}}$. Moreover, if $I \in \mathbb{Z}^m_{\geq 0}$, the mapping ψ has the following property:

$$\Theta(I)\psi(\underline{a}) = \sum_{J \in \mathbb{Z}^m_{\geq 0}} \frac{1}{J!} a_{I+J} t^J$$

which implies

$$\underline{a} = (\Theta(I)\psi(\underline{a})|_{t=0})_{I \in \mathbb{Z}^m_{\geq 0}}.$$

Fix for the rest of the section a finite set of differential polynomials $\Sigma = \{P_1, \ldots, P_s\} \subseteq G$ such that Σ has the same solution set as G (this is possible by Proposition 2.1). For all $\ell \in \{1, \ldots, s\}$ and $I \in \mathbb{Z}^m_{\geq 0}$ we define

$$F_{\ell,I} = (\Theta(I)P_\ell)|_{t_1=\cdots=t_m=0} \in K\left[x_{i,J} : 1 \leq i \leq n, J \in \mathbb{Z}^m_{\geq 0}\right]$$

and

$$A_\infty = \{(a_{i,J}) \in K^{n \times (\mathbb{Z}^m_{\geq 0})} : F_{\ell,I}(a_{i,J}) = 0 \text{ for all } 1 \leq \ell \leq s, I \in \mathbb{Z}^m_{\geq 0}\}.$$

The set A_∞ corresponds to the formal power series solutions of the differential system $\Sigma = 0$ as the following lemma shows.

Lemma 6.2. *Let $\varphi \in K[[t_1, \ldots, t_m]]^n$ with $\varphi = (\varphi_1, \ldots, \varphi_n)$, where*

$$\varphi_i = \sum_{J \in \mathbb{Z}^m_{\geq 0}} \frac{a_{i,J}}{J!} t^J.$$

Then φ is a solution of $\Sigma = 0$ if and only if $(a_{i,J}) \in A_\infty$.

Proof. This statement follows from formula

$$P_\ell(\varphi_1, \ldots, \varphi_n) = \sum_{I \in \mathbb{Z}^m_{\geq 0}} \frac{F_{\ell,I}((a_{i,J})_{i,J})}{I!} t^I,$$

which is commonly known as Taylor formula for multivariate formal power series. To prove this formula, first notice that for arbitrary $P \in R_{m,n}$ we have $P(\varphi)|_{t=0} = (P|_{t=0})((a_{i,J})_{i,J})$. Applying this to $P = \Theta(I)(P_\ell)$ for fixed I and ℓ, we find that

$$\Theta(I)(P_\ell(\varphi))\big|_{t=0} = (\Theta(I)(P_\ell)\big|_{t=0})((a_{i,J})_{i,J}) = F_{\ell,I}((a_{i,J})_{i,J}).$$

Therefore the coefficient of t^I in $P_\ell(\varphi)$ is $F_{\ell,I}((a_{i,J})_{i,J})/I!$, and this gives the formula above. □

For any $S = (S_1, \ldots, S_n) \in \mathcal{P}(\mathbb{Z}^m_{\geq 0})^n$ we define

$$A_{\infty,S} = \{(a_{i,J}) \in A_\infty : a_{i,J} = 0 \text{ if and only if } J \notin S_i\}.$$

This set corresponds to power series solutions of the system $\Sigma = 0$ which have support exactly S. In particular, $S \in \mathrm{Supp}(\mathrm{Sol}(G))$ if and only if $A_{\infty,S} \neq \emptyset$.

The sets A_∞ and $A_{\infty,S}$ refer to infinitely many coefficients. We want to work with a finite approximation of these sets. For this purpose, we make the following definitions. For each integer $k \geq 0$, choose $N_k \geq 0$ minimal such that for every $\ell \in \{1, \ldots, s\}$ and $||I||_\infty \leq k$ it holds that

$$F_{\ell,I} \in K[x_{i,J} : 1 \leq i \leq n, ||J||_\infty \leq N_k].$$

Note that for $k_1 \leq k_2$ it follows that $N_{k_1} \leq N_{k_2}$. Then we define

$$A_k = \{(a_{i,J}) \in K^{n \times \{1,\ldots,N_k\}^m} : F_{\ell,I}(a_{i,J}) = 0 \text{ for all } 1 \leq \ell \leq s, ||I||_\infty \leq k\}$$

and

$$A_{k,S} = \{(a_{i,J}) \in A_k : a_{i,J} = 0 \text{ if and only if } J \notin S_i\}.$$

Proposition 6.3. *Let $S \in \mathcal{P}(\mathbb{Z}^m_{\geq 0})^n$ and K be an uncountable algebraically closed field of characteristic zero. If $A_{\infty,S} = \emptyset$, then there exists $k \geq 0$ such that $A_{k,S} = \emptyset$.*

Proof. Assume that $A_{k,S} \neq \emptyset$ for every $k \geq 0$; we show that this implies $A_{\infty,S} \neq \emptyset$. We follow the strategy of the proof of [5, Theorem 2.10]: first we use the ultrapower construction to construct a larger field $\mathbb{K}$ over which a power series solution with support S exists, and then we show that this implies the existence of a solution with the same support and with coefficients in K. For more information on ultrafilters and ultraproducts, the reader may consult [2]. For each integer $k \geq 0$, choose an element $(a^{(k)}_{i,J})_{1 \leq i \leq n, ||J||_\infty \leq N_k} \in A_{k,S}$. Fix a non-principal ultrafilter $\mathcal{U}$ on the natural numbers $\mathbb{N}$ and consider the ultrapower $\mathbb{K}$ of K along $\mathcal{U}$. In other words, $\mathbb{K} = (\prod_{r \in \mathbb{N}} K)/\sim$ where $x \sim y$ for $x = (x_r)_{r \in \mathbb{N}}$ and $y = (y_r)_{r \in \mathbb{N}}$ if and only if the set $\{r \in \mathbb{N} : x_r = y_r\}$ is in $\mathcal{U}$. We will denote the equivalence class of a sequence (x_r) by $[(x_r)]$. We consider $\mathbb{K}$ as a K-algebra via the diagonal map $K \to \mathbb{K}$. Now for each i and J, we may define $a_{i,J} \in \mathbb{K}$ as

$$a_{i,J} = [(a^{(k)}_{i,J} : k \in \mathbb{N})]$$

where we set $a^{(k)}_{i,J} = 0$ for the finitely many values of k with $||J||_\infty > N_k$. For all ℓ and I, we have that $F_{\ell,I}((a^{(k)}_{i,J})_{i,J}) = 0$ for k large enough, and so $F_{\ell,I}((a_{i,J})_{i,J}) = 0$ in $\mathbb{K}$, because the set of k such that $F_{\ell,I}((a^{(k)}_{i,J})_{i,J}) \neq 0$ is finite. Moreover, for $J \in S_i$ we have, by hypothesis, $a^{(k)}_{i,J} \neq 0$ for all sufficiently large k, so $a_{i,J} \neq 0$ in $\mathbb{K}$. On the other hand, for $J \notin S_i$ we have $a^{(k)}_{i,J} = 0$ for all k, so also $a_{i,J} = 0$.

Now we will use that K is uncountable. Consider the ring

$$R = K\left[\begin{array}{c} x_{i,J} : 1 \leq i \leq n, J \in \mathbb{Z}^m_{\geq 0} \\ x^{-1}_{i,J} : 1 \leq i \leq n, J \in S_i \end{array}\right] \Big/ \left(\begin{array}{c} F_{\ell,I} : 1 \leq \ell \leq s, I \in \mathbb{Z}^m_{\geq 0} \\ x_{i,J} : 1 \leq i \leq n, J \notin S_i \end{array}\right)$$

The paragraph above shows that the map $R \to \mathbb{K}$ defined by sending $x_{i,J}$ to $a_{i,J}$ is a well-defined ring map. In particular, R is not the zero ring. Let $\mathfrak{m}$ be a maximal ideal of R. We claim that $K = R/\mathfrak{m}$ in the sense that the map $K \to R/\mathfrak{m}$ induced by the composition of the inclusion and the projection $K \to R \to R/\mathfrak{m}$ is an isomorphism. Indeed, $R/\mathfrak{m}$ is a field, and as a K-algebra it is countably generated, since R is. Therefore, it is of countable dimension as K-vector space (it is generated as K-vector space by the products of some set of generators as a K-algebra). If $t \in R/\mathfrak{m}$ were transcendental over K, then by the theory of partial fraction decomposition, the elements $1/(t - \alpha)$ for $\alpha \in K$ would form an uncountable, K-linearly independent subset of $R/\mathfrak{m}$. This is not possible, so $R/\mathfrak{m}$ is algebraic over K. Since K is algebraically closed, we conclude that $K = R/\mathfrak{m}$.

Now let $b_{i,J} \in K$ be the image of $x_{i,J}$ in $R/\mathfrak{m} = K$. Then by construction, the set $(b_{i,J})$ satisfies the conditions $F_{\ell,I}((b_{i,J})) = 0$ for all ℓ and I, and $b_{i,J} = 0$ if and only if $J \notin S_i$. So $(b_{i,J})$ is an element of $A_{\infty,S}$, and in particular $A_{\infty,S} \neq \emptyset$. □

Proof of Theorem 6.1. We now prove the remaining direction of the Fundamental Theorem by contraposition. Let $S = (S_1, \ldots, S_n)$ in $\mathcal{P}(\mathbb{Z}^m_{\geq 0})^n$ be such that $A_{\infty,S} = \emptyset$, i.e. there is no power series solution of $\Sigma = 0$ in $K[[t_1, \ldots, t_m]]^n$ with S as the support. Then by Proposition 6.3 there exists $k \geq 0$ such that $A_{k,S} = \emptyset$. Equivalently,

the relation

$$V\left(\begin{array}{l} F_{\ell,I} : 1 \le \ell \le s, ||I||_\infty \le k \\ x_{i,J} : 1 \le i \le n, J \notin S_i, ||J||_\infty \le N_k \end{array}\right) \subseteq V\left(\prod_{\substack{1\le i\le n \\ J\in S_i \\ ||J||_\infty \le N_k}} x_{i,J}\right)$$

holds, where V denotes the implicitly defined algebraic set. By Hilbert's Nullstellensatz, there is an integer $M \ge 1$ such that

$$E := \left(\prod_{\substack{1\le i\le n \\ J\in S_i \\ ||J||_\infty \le N_k}} x_{i,J}\right)^M \in \left\langle \begin{array}{l} F_{\ell,I} : 1 \le \ell \le s, ||I||_\infty \le k \\ x_{i,J} : 1 \le i \le n, J \notin S_i, ||J||_\infty \le N_k \end{array}\right\rangle.$$

Therefore, there exist $G_{\ell,I}$ and $H_{i,J}$ in $K[x_{i,J} : 1 \le i \le n, ||J||_\infty \le N_k]$ such that

$$E = \sum_{\substack{1\le \ell\le s \\ ||I||_\infty \le k}} G_{\ell,I} F_{\ell,I} + \sum_{\substack{1\le i\le n \\ J\notin S_i \\ ||J||_\infty \le N_k}} H_{i,J} x_{i,J}.$$

Define the differential polynomial P by

$$P = \sum_{\substack{1\le \ell\le s \\ ||I||_\infty \le k}} G_{\ell,I}\Theta(I)(P_\ell).$$

Then P is an element of the differential ideal generated by $P_1, \ldots, P_s$, so in particular $P \in G$. Since $F_{\ell,I} = \Theta(I)(P_\ell)|_{t=0}$, there exist $h_i \in R_{m,n}$ such that

$$P = E - \sum_{\substack{1\le i\le n \\ J\notin S_i \\ ||J||_\infty \le N_k}} H_{i,J} x_{i,J} \; + \; t_1 h_1 + \ldots + t_m h_m.$$

Notice that the monomial E occurs effectively in P, since it cannot cancel with other terms in the sum above. By construction we have $\mathrm{trop}(E)(S) = \{(0,\ldots,0)\}$. However, we have $(0,\ldots,0) \notin \mathrm{trop}(H_{i,J}x_{i,J})(S)$ because $J \notin S_i$, and we have $(0,\ldots,0) \notin \mathrm{trop}(t_i h_i)(S)$ because the factor t_i forces the ith coefficient of each element of $\mathrm{trop}(t_i h_i)(S)$ to be at least 1. Hence, the vertex $(0,\ldots,0)$ in $\mathrm{trop}(P)(S)$ is attained exactly once, in the monomial E, and therefore, S is not a solution of $\mathrm{trop}(P)$. Since $P \in G$, it follows that $S \notin \mathrm{Sol}(\mathrm{trop}(G))$, which proves the statement. □

7 EXAMPLES AND REMARKS ON THE FUNDAMENTAL THEOREM

In this section we give an example to illustrate the results obtained in the previous sections. Moreover, we show that some straightforward generalizations of the Fundamental Theorem from [1] and our version, Theorem 6.1, do not hold. Also we give more directions for further developments.

Example 7.1. Let us consider in $R_{2,2}$ the system

$$\Sigma = \{P_1 = x_{1,(1,0)}^2 - 4\,x_{1,(0,0)},\; P_2 = x_{1,(1,1)}\,x_{2,(0,1)} - x_{1,(0,0)} + 1,\; P_3 = x_{2,(2,0)} - x_{1,(1,0)}\}.$$

By means of elimination methods in differential algebra such as the ones implemented in the MAPLE `DifferentialAlgebra` package, it can be proven that

$$\begin{aligned}\mathrm{Sol}(\Sigma) = \{&\varphi_1(t_1,t_2) = 2\,c_0\,t_1 + c_0^2 + \sqrt{2}\,c_0\,t_2 + t_1^2 + \sqrt{2}\,t_1\,t_2 + \frac{1}{2}\,t_2^2,\\ &\varphi_2(t_1,t_2) = c_2\,t_1 + c_1 + \frac{1}{2}\,\sqrt{2}\,(c_0^2-1)\,t_2 + c_0\,t_1^2\\ &\quad + \sqrt{2}\,c_0\,t_1\,t_2 + \frac{1}{2}\,c_0\,t_2^2\\ &\quad + \frac{1}{3}\,t_1^3 + \frac{1}{2}\,\sqrt{2}\,t_1^2\,t_2 + \frac{1}{2}\,t_1\,t_2^2 + \frac{1}{12}\,\sqrt{2}\,t_2^3\},\end{aligned}$$

where $c_0, c_1, c_2 \in K$ are arbitrary constants. By setting $c_0 = c_2 = 0, c_1 \neq 0$, we obtain for example that

$$(\{(2,0),(1,1),(0,2)\},\{(0,0),(0,1),(3,0),(2,1),(1,1),(0,3)\})$$

is in $\mathrm{Supp}(\mathrm{Sol}(\Sigma))$.

Now we illustrate that by our results necessary conditions and relations on the support can be found. Let $(S_1,S_2) \in \mathcal{P}(\mathbb{Z}^2_{\ge 0})^2$ be a solution of $\mathrm{trop}([\Sigma])$. Let us first consider

$$\mathrm{trop}(P_1)(S_1,S_2) = \mathrm{Vert}(2\cdot\Theta_{\mathrm{trop}}(1,0)S_1 \,\cup\, S_1).$$

If we assume that $(0,0) \in S_1$, then $(0,0)$ is a vertex of S_1. By the definition of a solution of a tropical differential polynomial, $(0,0)$ must be a vertex of the term $2\cdot\Theta_{\mathrm{trop}}(1,0)S_1$ as well, so we then know that $(1,0) \in S_1$. Conversely, if $(1,0) \in S_1$, then $(0,0) \in S_1$ follows. This is what we expect since the corresponding monomials in φ_1 vanish if and only if $c_0 = 0$.
Now consider

$$\begin{aligned}&\mathrm{trop}(\Theta(1,0)P_1)(S_1,S_2) =\\ &\mathrm{Vert}(\Theta_{\mathrm{trop}}(1,0)S_1 + \Theta_{\mathrm{trop}}(2,0)S_1 \,\cup\, \Theta_{\mathrm{trop}}(1,0)S_1).\end{aligned}$$

If we assume that $(0,0)$ is not a vertex of this expression, which implies that $(1,0) \notin S_1$, and $(k,0)$ is a vertex in $\Theta_{\mathrm{trop}}(1,0)S_1$ for some $k \ge 1$, then we obtain from the two tropical differential monomials that necessarily $(k,0) = (2k-1,0)$. This is fulfilled only for $k = 1$ and hence, $(2,0) \in S_1$.

Another natural way for defining $\odot$ and $\oplus$ in Section 3 would be to simply take the minimal basis of the monomial ideal generated by the support of the series rather than the (possibly smaller) vertex set, as we do. If we do this, then some intermediate results (and in particular Proposition 5.7) do not hold anymore as the following example shows.

Example 7.2. Let $\{e_1,\ldots,e_4\}$ be the standard basis for $\mathbb{Z}^4_{\ge 0}$. We consider the differential ideal in $R_{4,1} = K[[t_1,\ldots,t_4]]\{x\}$ generated by

$$P = x_{e_3}x_{e_4} + (-t_1^2 + t_2^2)x_{e_1+e_3} = \frac{\partial x}{\partial t_3}\cdot\frac{\partial x}{\partial t_4} + (-t_1^2+t_2^2)\frac{\partial^2 x}{\partial t_1 \partial t_3}$$

and the solution $\varphi = (t_1+t_2)t_3 + (t_1-t_2)t_4$. Then

$$\mathrm{Supp}(\varphi) = \{e_1+e_3, e_2+e_3, e_1+e_4, e_2+e_4\}.$$

On the other hand, for $S \in \mathcal{P}(\mathbb{Z}^4_{\ge 0})$ we obtain

$$\begin{aligned}\mathrm{trop}(P)(S) = \mathrm{Vert}(\,&\mathrm{Vert}(\Theta_{\mathrm{trop}}(e_3)S + \Theta_{\mathrm{trop}}(e_4)S)\\ &\cup \mathrm{Vert}(2e_1 + \Theta_{\mathrm{trop}}(e_1+e_3)S)\\ &\cup \mathrm{Vert}(2e_2 + \Theta_{\mathrm{trop}}(e_1+e_3)S).\end{aligned}$$

If we set $S = \mathrm{Supp}(\varphi)$, we obtain

$$\mathrm{trop}(P)(S) = \mathrm{Vert}(\,\mathrm{Vert}(\{2e_1, e_1+e_2, 2e_2\}) \cup \{2e_1\} \,\cup\, \{2e_2\}).$$

Since

$$\mathrm{Vert}(\{2e_1, e_1+e_2, 2e_2\}) = \{2e_1, 2e_2\},$$

every $J \in \mathrm{trop}(P)(S)$, namely $2e_1$ and $2e_2$, occurs in three monomials in $\mathrm{trop}(P)(S)$ and S is indeed in $\mathrm{Sol}(\mathrm{trop}(P))$. Note that in the Newton polytope the point e_1+e_2, which is not a vertex, comes from only one monomial in $\mathrm{trop}(P)(S)$. Therefore, it is necessary to consider the vertices instead of the whole Newton polytope such that for instance Proposition 5.7 holds.

Remark 7.3. The Fundamental Theorem for systems of partial differential equations over a countable field such as $\overline{\mathbb{Q}}$ does in general not hold anymore by the following reasoning. According to [5, Corollary 4.7], there is a system of partial differential equations G over $\mathbb{Q}$ having a solution in $\mathbb{C}[[t_1,\ldots,t_m]]$ but no solution in $\overline{\mathbb{Q}}[[t_1,\ldots,t_m]]$. Taking $K=\overline{\mathbb{Q}}$ as base field, we have $\mathrm{Sol}(\mathrm{trop}(G)) \neq \emptyset$ because $\mathrm{Sol}(\mathrm{trop}(G)) = \mathrm{Supp}(\mathrm{Sol}(G))$ is non-empty in $\mathbb{C}$, but $\mathrm{Supp}(\mathrm{Sol}(G)) = \emptyset$.

In this paper we focus on formal power series solutions. A natural extension would be to consider formal Puiseux series instead. The following example shows that with the natural extension of our definitions to Puiseux series, the fundamental theorem does not hold, even for $m=n=1$.

Example 7.4. Let us consider $R_{1,1} = K[t]\{x\}$ and the differential ideal generated by the differential polynomial

$$P = 2tx_{(1)} - x_{(0)} = 2t\cdot\frac{\partial x}{\partial t} - x.$$

There is no non-zero formal power series solution φ of $P=0$, but $\varphi = ct^{1/2}$ is for any $c \in K$ a solution. In fact, $\{\varphi\}$ is the set of all formal Puiseux series solutions.

On the other hand, let $S \in \mathcal{P}(\mathbb{Z}_{\geq 0})$. Then every point J in

$$\mathrm{trop}(P)(S) = \mathrm{Vert}(\mathrm{Vert}(\{1\} + (\Theta_{\mathrm{trop}}(1)S) \cup \mathrm{Vert}(S))$$

occurs in both monomials except if $0 \in S$. Hence, for every $S \in \mathrm{Sol}(\mathrm{trop}(P))$ we know that $0 \notin S$. For every $I \geq 0$ we have that

$$\Theta(I)P = 2tx_{(I+1)} + (2I-1)x_{(I)} \in [P]$$

and

$$\mathrm{trop}(\Theta(I)P)(S) = \mathrm{Vert}(\mathrm{Vert}(\{(1)\} + (\Theta_{\mathrm{trop}}((I+1)S) \cup \mathrm{Vert}(\Theta_{\mathrm{trop}}(I)S)).$$

Similarly to above, every $J \in \mathrm{trop}(\Theta(I)P)(S)$ occurs in both monomials except if $I \in S$. Therefore, $I \notin S$ and so the only $S \in \mathcal{P}(\mathbb{Z}_{\geq 0})$ with $S \in \mathrm{Sol}(\mathrm{trop}([P]))$ is $S = \emptyset$. Hence, $\mathrm{Sol}(\mathrm{trop}([P])) = \{\emptyset\} = \mathrm{Supp}(\mathrm{Sol}([P]))$.

Now we want to consider formal Puiseux series solutions instead of formal power series solutions. Now let us set for $S \in \mathbb{Q}^m$ and $J = (j_1,\ldots,j_m) \in \mathbb{Z}^m_{\geq 0}$, the set $\Theta_{\mathrm{trop}}(J)S$ defined as

$$\left\{(s_1-j_1,\ldots,s_m-j_m) \;\middle|\; \begin{array}{c}(s_1,\ldots,s_m)\in S,\\ \forall 1\leq i\leq m,\ s_i<0 \text{ or } s_i-j_i\notin\mathbb{Z}_{<0}\end{array}\right\}$$

This is the natural definition, since only in the case when the exponent of a monomial is a non-negative integer, the derivative can be equal to zero. We have that $\Theta_{\mathrm{trop}}(J)(\mathrm{Supp}(\psi)) = \mathrm{Supp}(\Theta(J)\psi)$ for all Puiseux series ψ. For Val_J and the operations $\odot$ and $\oplus$ the definitions remain unchanged.

Let $Q \in [P]$. Then

$$Q = \sum_{k\in\mathcal{I}} Q_k\cdot\Theta(I_k)P$$

for some index-set $\mathcal{I}$ and $Q_k \in R_{m,n}$. For every I_k we know that $\mathrm{Supp}(\varphi) = \{(1/2)\} \in \mathrm{Sol}(\mathrm{trop}(\Theta(I_k)P))$. Let $\alpha \in \mathbb{Q}\cap(0,1)$. Then for every $J \in \mathrm{trop}(\Theta(I_k)P) \in \mathbb{Z}_{\geq 0}$ we have that $\Theta_{\mathrm{trop}}(J)\{(1/2)\} = \Theta_{\mathrm{trop}}(J)\{(\alpha)\} + \{(1/2-\alpha)\}$. Thus, $\{\alpha\} \in \mathrm{Sol}(\mathrm{trop}(\Theta(I_k)P))$. Since

$$\mathrm{trop}(Q_k\cdot\Theta(I_k)P) = \mathrm{trop}(Q_k)\odot\mathrm{trop}(\Theta(I_k)P),$$

the solvability remains by multiplication with Q_k. Therefore, $\{\alpha\} \in \mathrm{Sol}(\mathrm{trop}(Q_k\cdot\Theta(I_k)P))$ and consequently, $\{\alpha\} \in \mathrm{Sol}(\mathrm{trop}([P]))$. However, $\{\alpha\} \notin \mathrm{Supp}(\mathrm{Sol}([P])) = \{\emptyset, \{1/2\}\}$ for $\alpha \neq 1/2$.

We remark that P is an ordinary differential polynomial and by similar computations as here, the straight-forward generalization from formal power series to formal Puiseux series fails for the Fundamental Theorem in [1] as well.

We conclude this section by emphasizing that the Fundamental Theorem may help to find necessary conditions on the support of solutions of systems of partial differential equations, but in general it cannot be completely algorithmic. In fact, according to [5, Theorem 4.11], already determining the existence of a formal power series solution of a linear system with formal power series coefficients is in general undecidable.

Acknowledgements

This research project and the fifth author was supported by the European Commission, having received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement number 792432. The first author was supported by the Austrian Science Fund (FWF): P 31327-N32. The second author was supported by CONACYT Project 299261. The sixth author would like to thank the bilateral project ANR-17-CE40-0036 and DFG-391322026 SYMBIONT for its support. Partially supported by PAPIIT IN108320.

This work was started during the Tropical Differential Algebra workshop, which took place on December 2019 at Queen Mary University of London. We thank the organizers and participants for valuable discussions and initiating this collaboration. In particular, we want to thank Fuensanta Aroca, Alex Fink, Jeffrey Giansiracusa and Dima Grigoriev for their helpful comments during this week.

We thank the anonymous referees for their suggestions, which helped us to improve the exposition of this work.

REFERENCES

[1] Fuensanta Aroca, Cristhian Garay, and Zeinab Toghani. The Fundamental Theorem of Tropical Differential Algebraic Geometry. *Pacific J. Math.*, 283(2):257–270, 2016. arXiv:1510.01000v3.

[2] Joseph Becker, Jan Denef, Leonard Lipshitz, and Lou van den Dries. Ultraproducts and Approximation in Local Rings I. *Inventiones mathematicae*, 51:189–203, 1979.

[3] François Boulier and Mercedes Haiech. The Ritt-Raudenbush Theorem and Tropical Differential Geometry. Available at https://hal.archives-ouvertes.fr/hal-02403365, 2019.

[4] David Cox, John Little, and Donal O'Shea. *Ideals, Varieties and Algorithms. An introduction to computational algebraic geometry and commutative algebra.* Undergraduate Texts in Mathematics. Springer Verlag, New York, 3rd edition, 2007.

[5] J. Denef and L. Lipshitz. Power series solutions of algebraic differential equations. *Math. Ann.*, 267(2):213–238, 1984.

[6] Jeffrey Giansiracusa and Noah Giansiracusa. Equations of tropical varieties. *Duke Math. J.*, 165(18):3379–3433, 2016.

[7] Dima Grigoriev. Tropical differential equations. *Advances in Applied Mathematics*, 82:120–128, 2017.

[8] Ellis Robert Kolchin. *Differential Algebra and Algebraic Groups.* Academic Press, New York, 1973.

[9] Bernd Sturmfels. *Gröbner bases and convex polytopes*, volume 8 of *University Lecture Series.* American Mathematical Society, Providence, RI, 1996.

Special-case Algorithms for Blackbox Radical Membership, Nullstellensatz and Transcendence Degree

Abhibhav Garg
CSE, IIT Kanpur
abhibhav@cse.iitk.ac.in

Nitin Saxena
CSE, IIT Kanpur
nitin@cse.iitk.ac.in

ABSTRACT

Radical membership testing, resp. its special case of Hilbert's Nullstellensatz (HN), is a fundamental computational algebra problem. It is NP-hard; and has a famous PSPACE algorithm due to *effective* Nullstellensatz bounds. We identify a useful case of these problems where practical algorithms, & improved bounds, could be given—When transcendence degree (tr.deg) r of the input polynomials is smaller than the number of variables n. If d is the degree bound on the input polynomials, then we solve radical membership (even if input polynomials are *blackboxes*) in around d^r time. The prior best was $> d^n$ time (always, $d^n \geq d^r$). Also, we significantly improve effective Nullstellensatz degree-bound, when $r \ll n$.

Structurally, our proof shows that these problems reduce to the case of $r+1$ polynomials of tr.deg $\geq r$. This input instance (corresponding to none or a *unique annihilator*) is at the core of HN's hardness. Our proof methods invoke basic algebraic-geometry.

CCS CONCEPTS

• **Theory of computation → Algebraic complexity theory**; *Circuit complexity*; Problems, reductions and completeness.

ACM Reference Format:
Abhibhav Garg and Nitin Saxena. 2020. Special-case Algorithms for Blackbox Radical Membership, Nullstellensatz and Transcendence Degree. In *International Symposium on Symbolic and Algebraic Computation (ISSAC '20), July 20–23, 2020, Kalamata, Greece.* , 8 pages. https://doi.org/10.1145/3373207.3404030

1 INTRODUCTION

Given a set of polynomials $f_1, \ldots, f_n$, there is a natural certificate for the existence of a common root, namely the root itself. Hilbert's Nullstellensatz [Rab30, Zar47, Kru50] states that there is also a natural certificate for the nonexistence of a common root, when the underlying field is *algebraically closed*. Formally, the theorem states that the polynomials have no common root if and only if there exist polynomials $g_1, \ldots, g_n$ such that $1 = \sum f_i g_i$. We refer to the latter type of certificate as a *Nullstellensatz certificate*. These certificates are not polynomial sized: every common root can have exponential bit complexity, and every set of witness polynomials g_i can have exponential degrees. This problem is naturally of computational interest, since the generality of the statement affords reductions from many problems of interest. *Effective* versions of the Nullstellensatz have been extensively studied [Jel05, KPS$^+$01, KPS99, Som99, Som97, BS91, Kol88, Bro87], and they allow the decision problem of existence of common roots (called HN) to be solved in polynomial space. Koiran [Koi96] proved that under generalized Riemann Hypothesis, HN can be solved in AM [AB09, Ch.8], for fields of characteristic zero.

In this work, we relate the complexity of HN to the transcendence degree of the input polynomials. The *transcendence degree* (tr.deg) of polynomials $f_1, \ldots, f_m$ is defined as the size of any maximal subset of the polynomials that are algebraically independent. This notion is well defined since algebraic independence satisfies matroid properties [Oxl06]. We show that HN can be solved in time single-exponential in tr.deg. This can be seen as a generalization of the fact that HN can be solved in time exponential in the number of polynomials (or variables) in the system. We state our result in terms of the question of *radical membership*: $f_0 \in ?\sqrt{\langle f_1, \ldots, f_m \rangle}$. Note that the standard algorithms for both ideal membership [Her26] and radical computation [Lap06] are far slower than ours.

Given a set of polynomials $f_1, \ldots, f_m$ with tr.deg at most r, as blackboxes, we can perform radical membership tests for the ideal generated by $f_1, \ldots, f_m$ in time polynomial in d^r, m, n, where d is the degree-bound on the polynomials and n is the number of variables.

We also relate the tr.deg of the input polynomials to the degrees of the Nullstellensatz certificates, that is the degrees of g_i in $\sum f_i g_i = 1$; improving the best bounds by [Jel05].

Given a set of polynomials $f_1, \ldots, f_m$ with tr.deg r and without any common roots, there exist polynomials g_i of degree at most d^{r+1} such that $\sum f_i g_i = 1$.

We also give an output-sensitive algorithm to compute the tr.deg of polynomials. Slightly more formally, we show:

Given a set of polynomials $f_1, \ldots, f_m$, we can compute their tr.deg in time polynomial in d^r and m, n.

1.1 Previously known results

All three of the problems stated above have been extensively studied. We therefore only list some of the previously known results, and direct readers to the surveys [May97, BS91].

Nullstellensatz. The decidability of the ideal membership problem was established by Hermann [Her26] when she proved a doubly-exponential bound on witnesses to ideal membership. A lower bound of the same complexity by Mayr and Meyer [May89, MM82] showed that this problem is EXPSPACE complete. A number of different algorithms were developed for operations on ideals, most prominently the method of Gröbner basis [Buc65]. The proof of single-exponential bounds for the Nullstellensatz (discussed below) allowed special cases of the ideal membership problem, such

ISSAC '20, July 20–23, 2020, Kalamata, Greece

ACM ISBN 978-1-4503-7100-1/20/07…$15.00
https://doi.org/10.1145/3373207.3404030

as the case of unmixed and zero dimensional ideals to be solved in single-exponential time [DFGS91]. It also allowed the general Nullstellensatz problem to be solved in PSPACE. Giusti and Heintz [GH94] proved that the dimension of a variety can be computed by a randomized algorithm in single-exponential time, with the exponent being linear in n, which gives an algorithm of the same complexity for HN (by testing if the dimension is -1). All of the above results are independent of the underlying field characteristic. In 1996, Koiran [Koi96] gave an AM protocol (conditioned on GRH) for the Nullstellensatz problem, when the underlying field is $\mathbb{C}$ and the polynomials have integer coefficients. His method is completely different from the previous methods (of using the effective Nullstellensatz to reduce the system to a linear one). The positive characteristic case is an open problem, and the best known complexity remains PSPACE.

Effective Nullstellensatz. The projective version of the effective Nullstellensatz follows from the fundamental theorem of elimination theory [Laz77]. An affine version was first proved by Brownawell [Bro87] in characteristic 0 using analytic methods. It was later improved by Kollár [Kol88] who used local cohomology to improve the bounds and remove the condition on the characteristic. A more elementary proof that used bounds on the Hilbert function was given by Sombra [Som97], who also gave improved bounds based on some geometric properties of related varieties [Som99]. An even more elementary and significantly shorter proof was given by Jelonek [Jel05], who obtained improved bounds when the number of polynomials is lesser than the number of variables.

Transcendence degree. Algebraic independence was studied in computer science by [DGW07] in their study of explicit extractors. They proved that the rank of the Jacobian matrix is the same as the tr.deg for fields of characteristic zero (or large enough) which gives an efficient randomized method for computing the tr.deg. The problem was studied further in [Kay09], where the condition on the characteristic for the above algorithm was relaxed, and some hardness results were established. [GSS18] showed that the problem is in coAM $\cap$ AM, making it unlikely to be NP-hard, and conjecturing that the problem is in coRP for all characteristics. Algorithmically, the best known method for computing the tr.deg in fields of positive characteristic still has PSPACE complexity, by using the bounds of Perron [Per51, Pł05] to reduce the problem to solving an exponential sized linear system. This method takes time polynomial in d^{r^2} using the methods of [Csa75]. We refer the reader to the thesis [Sin19] for an exhaustive survey of related results; and applications in [ASSS12, PSS16].

Certain radical membership methods were developed by Gupta [Gup14] in his work on deterministic polynomial identity testing algorithms for heavily restricted depth-four circuits. The focus there however was on a *deterministic algorithm* for the above problem. Further, he restricts his attention to systems where the underlying field is $\mathbb{C}$.

1.2 Our results

Our algorithms will be Monte Carlo algorithms. We assume that our base field k is algebraically closed, but our algorithms only use operations in the field in which the coefficients of the inputs lie, which we denote by k_i. For example, k_i might be $\mathbb{F}_p$, and k would then be $\overline{\mathbb{F}_p}$. By time complexity we mean operations in k_i, where operations include arithmetic operations, finding roots, and computing GCD of polynomials. Our results are valid for any field where the above procedures are efficient, for example finite fields.

We relate the complexity of radical membership, and the degree bounds in effective Nullstellensatz, to the tr.deg of the input set of polynomials. We do this by showing that given a system of polynomials, we can reduce both the number of variables and the number of polynomials to one more than the tr.deg, while preserving the existence (resp. non-existence) of common roots. In particular, when the tr.deg of the input polynomials is constant, we get efficient algorithms for these problems.

Theorem 1.1 (Radical membership). *Suppose $f_1, \ldots, f_m$ and g are polynomials, in variables $x_1, \ldots, x_n$, of degrees $d_1, \ldots, d_m$ and d_g respectively, given as blackboxes. Suppose that $tr.deg(f_1, \ldots, f_m) \leq r$. Define $d := \max(\max_i d_i, d_g)$.*

Then, testing if g belongs to the radical of the ideal generated by $f_1, \ldots, f_m$ can be done in time polynomial in n, m and d^r, with randomness.

Remarks:

(1) The tr.deg r can be much smaller than n, and this improves the complexity significantly to d^r from the prior d^n [LL91]. On the other hand, the usual reduction from SAT to HN results in a set of polynomials with transcendence degree n, due to the presence of polynomials $x_i^2 - x_i$ (that enforce the binary 0/1 values).

(2) We also show that the tr.deg itself can be computed in time d^r, independent of the characteristic (Theorem 1.3). In the above statement therefore, we can always pick $r = \text{tr.deg}(\mathbf{f})$, and we can assume that r is not part of the input.

(3) The tr.deg is upper bounded by the number of polynomials, and therefore we generalize the case of few polynomials. It is surprising if one contrasts this case with that of *ideal* membership—where the instance with three polynomials (i.e. tr.deg=3) is as *hard* as the general instance making it EXPSPACE-complete. [1]

Next, we show that taking constant-free random linear combinations preserves the zeroset of the polynomials, if the number of linear combinations is at least one more than the tr.deg. This allows us to get bounds on the Nullstellensatz certificates that depend on the tr.deg.

Theorem 1.2 (Effective Nullstellensatz). *Suppose $f_1, \ldots, f_m$ are polynomials in $x_1, \ldots, x_n$, of degrees $d_1 \geq \cdots \geq d_m$ respectively, with an empty zeroset. Suppose further that $tr.deg(f_1, \ldots, f_m) = r$.*

Then, there exist polynomials h_i such that $\deg f_i h_i \leq \prod_{i=1}^{r+1} d_i$ that satisfy $\sum f_i h_i = 1$.

Remark: The prior best degree-bound for the case of 'small' tr.deg is $\prod_{i=1}^{m} d_i$ [Jel05]. Our bound is significantly better when the tr.deg r is 'smaller' than the number of polynomials m.

Finally, as stated before, we show that the tr.deg of a given system of polynomials can be computed in time polynomial in d^r (and m, n), where d is the maximum degree of the input polynomials, and r is

[1] Suppose $g \in \langle f_1, \ldots, f_m \rangle$ is an instance of ideal membership. This is equivalent to $z_1^m z_2^m g \in \langle z_1^{m+1}, z_2^{m+1}, \sum_i f_i z_1^i z_2^{m-i} \rangle$. Here, z_1, z_2 are fresh variables. This reduces the general instance of ideal membership to an instance where the ideal is generated by 3 elements. This transformation is from [Sap19].

their tr.deg. The algorithm is output-sensitive in the sense that the time-complexity depends on the output number r.

Theorem 1.3 (tr.deg). *Given as input polynomials $f_1, \ldots, f_m$, in variables $x_1, \ldots, x_n$, of degrees at most d, we can compute the tr.deg r of the polynomials in time polynomial in d^r, n, m.*

Remark: In the case when the characteristic of the field is greater than d^r, there is a much more efficient (namely, randomized polynomial time) algorithm using the Jacobian criterion [BMS13]. The algorithm presented here is useful when the characteristic is 'small'; whereas the previous best known time-complexity was $> d^{r^2}$ if one directly implements the PSPACE algorithm. Eg. for $d = O(1)$ and $r = O(\log n)$ our complexity is polynomial-time unlike the prior known algorithms.

A motivating example where our results are better than the known results is when the input blackboxes are implicitly of the form $f_i(h_1, \ldots, h_r), i \in [m]$, for $r \ll n$, where each h_i is an n-variate polynomial, and $m = n + 1$. Here, f_i's have transcendence degree r. Thus, our algorithms take time d^r; significantly less than d^n.

1.3 Proof ideas

Pf. idea Theorem 1.1: We first use the Rabinowitsch trick to reduce to HN: the case $g = 1$. Next, we perform a random linear variable-reduction. We show that replacing each x_i with a linear combination of r new variables z_j preserves the existence of roots. This is done by using the fact that a general linear hyperplane intersects a variety properly (Lemma 3.1). Once we are able to reduce the variables, we can interpolate to get dense representation of our polynomials, and invoke existing results about testing nonemptiness of varieties (Theorem 2.6).

Pf. idea Theorem 1.2: For the second theorem, we show that random linear combinations of the input polynomials, as long as we take at least $r+1$ many of them, preserve the zeroset. For this, we study the image of the polynomial map defined by the polynomials. We again use the theorem regarding the hyperplane intersection (Lemma 3.1). In order to get the degree bounds, we must allow these hyperplanes to depend on fewer variables, and allow their equations to be constant free. Once this is proved, we can use a bound (Theorem 2.5) on the Nullstellensatz certificates for the new polynomials (which is better since the polynomials are fewer in number) to obtain a bound for the original polynomials.

Pf. idea Theorem 1.3: The image of the polynomial map defined by the polynomials is such that the general fibre has codimension equal to the tr.deg. We first show that a random point, with coordinates from a subset which is not 'too large', satisfies this property. In order to efficiently compute the dimension of this fibre, we take intersections with hyperplanes; and apply Lemma 3.1 and Theorem 2.6.

2 NOTATION AND PRELIMINARIES

2.1 Notation

We reserve n for the number of variables $(x_1, \ldots, x_n)$, m for the number of polynomials $(f_1, \ldots, f_m)$ in our inputs. The polynomials have total degrees $d_1, \ldots, d_m$. We assume that the polynomials are labeled such that $d_1 \geq d_2 \geq \cdots \geq d_m$.

We use boldface to denote sequence of objects, when the indexing set is clear; for example, $\mathbf{x}$ denotes $x_1, \ldots, x_n$ and $\mathbf{f}$ denotes $f_1, \ldots, f_m$. The point $(0, \ldots, 0)$ will be represented by $\mathbf{0}$. We use k to denote the underlying field which we assume is algebraically closed, and k_i to denote the field in which the coefficients of the inputs lie. We use $\mathbb{A}^n$ to denote the n dimensional affine space over k. Given a variety X, we use $k[X]$ to denote its coordinate ring, and when X is irreducible we use $k(X)$ to denote its function field. We use $\mathbb{A}^n$ and $\mathbb{P}^n$ to denote the n dimensional affine and projective spaces respectively, and $\mathbb{P}^n_\infty$ to denote the hyperplane at infinity.

2.2 Algebraic-geometry facts

We use elementary facts from algebraic-geometry, for which [CLO07, SR13] are good references. We do not assume that our varieties (or zerosets) are irreducible. We will use the *Noether normalization lemma*. The following statement is useful, as it characterizes the linear maps which are Noether normalizing.

Theorem 2.1. [SR13, Thm.1.15] *If $X \subseteq \mathbb{P}^N$ is a closed subvariety disjoint from an ℓ-dimensional linear subspace $E \subseteq \mathbb{P}^N$ then the projection $\pi : X \to \mathbb{P}^{N-\ell-1}$ with centre E defines a* finite *map $X \to \pi(X)$.*

Here, by *projection with center E* we mean that the coordinate functions of the map are the same as a set of defining linear equations for E. By the above theorem, proving that a given map is Noether normalizing for a particular variety reduces to proving that the variety is disjoint from a linear subspace.

We will also use the following two statements from dimension theory, namely the theorem on the dimension of intersections with hypersurfaces, and the theorem on the dimension of fibres.

Theorem 2.2. [SR13, Thm.1.22] *If a form F is not zero on an irreducible projective variety X then* $\dim(X \cap V(F)) = \dim X - 1$.

Theorem 2.3 (Fibre dimension). [SR13, Thm.1.25] *Let $f : X \to Y$ be a* surjective *regular map between irreducible varieties. Then $\dim Y \leq \dim X$, and for every $y \in Y$, the fibre $f^{-1}(y)$ satisfies $\dim f^{-1}(y) \geq \dim X - \dim Y$ (equiv. $\operatorname{codim} f^{-1}(y) \leq \dim Y$).*

Further, there is a nonempty open subset $U \subset Y$: for every $y \in U$, $\dim f^{-1}(y) = \dim X - \dim Y$ (equiv. $\operatorname{codim} f^{-1}(y) = \dim Y$).

The above theorem also holds if we replace surjective by *dominant*. Every fibre either is empty (if the point is not in the image) or has the above bound on the dimension. We sketch a proof of a special case of the above in Appendix A since we require an intermediate statement in the proof of Theorem 1.3.

We will also require the Bézout inequality. The definition of degree we use is the version more common in computational complexity. The degree of a variety is the sum of the degrees of all its irreducible components, as opposed to just the components of highest dimension. For irreducible varieties, the degree is the number of points when intersected with a general linear subspace of complementary dimension. This definition affords the following version of the *Bézout inequality* [Hei83], which holds without any conditions on the type of intersection.

Theorem 2.4 (Bézout [Hei83]). *Let X, Y be subvarieties of $\mathbb{A}^n$. Then $\deg(X \cap Y) \leq \deg X \cdot \deg Y$.*

Following is a recent version of *effective Nullstellensatz* [Jel05].

Theorem 2.5. [Jel05, Thm.1.1] *Let $f_1, \ldots, f_m$ be nonconstant polynomials, from the ring $k[x_1, \ldots, x_n]$ with k algebraically closed, that have no common zeros. Assume* $\deg f_i = d_i$ *with* $d_1 \geq \cdots \geq d_m$, *and also* $m \leq n$. *Then, there exist polynomials h_i such that* $\deg f_i h_i \leq \prod_{i=1}^{m} d_i$ *satisfying* $\sum f_i h_i = 1$.

We will need the following algorithm for checking if a variety has dimension 0 (dim is an integer in the range $[-1, n]$). The statement assumes that the polynomials are given in the monomial (also called *dense*) representation. We only state the part of the theorem that we require. A discussion is provided in Appendix B. We note that the below theorem itself invokes results from [Laz81], section 8 of which proves that the operations occur in a field extension of degree at most d^n of the field k_i.

Theorem 2.6. [LL91, Part of Thm.1] *Let $f_1, \ldots, f_m$ be polynomials of degree at most d in n variables. There exists a randomized algorithm that checks if the dimension of the zeroset of $f_1, \ldots, f_m$ is 0 or not, in time polynomial in d^n, m. The error-probability is 2^{-d^n}.*

We will also require a bound on the degrees of *annihilators* of algebraically dependent polynomials. We refer to this bound as the Perron bound. It also plays a crucial role in the new proofs of effective Nullstellensatz (Theorem 2.5).

Theorem 2.7 (Perron bound). [BMS13, Cor.5] *Let $f_1, \ldots, f_m$ be algebraically dependent polynomials of degrees $d_1, \ldots, d_m$. Then there exists a nonzero polynomial $A(y_1, \ldots, y_m)$ of degree at most $\prod_{i=1}^{m} d_i$ such that $A(f_1, \ldots, f_m)$ is identically zero.*

We note that the theorem statement in [BMS13] has the bound as $(\max d_i)^m$, however their method of constructing a linear faithful homomorphism and then applying the bound from [Pł05] actually gives the above mentioned bound (even for the *weighted*-degree of A).

In the course of our proof, we will study the image of the polynomial map whose coordinate functions are $f_1, \ldots, f_m$. We list some properties of this image.

Lemma 2.8 (Polynomial map). *Let $f_1, \ldots, f_m$ be polynomials of degrees at most d, in variables $x_1, \ldots, x_n$. Set $r := tr.deg(f_1, \ldots, f_m)$. Let $F : \mathbb{A}^n \to \mathbb{A}^m$ be a polynomial map defined as*

$$F(a_1, \ldots, a_n) = (f_1(a_1, \ldots, a_n), \ldots, f_m(a_1, \ldots, a_n)).$$

Let Y be the (Zariski) closure of the image of $\mathbb{A}^n$ under F, that is $Y := \overline{F(\mathbb{A}^n)}$. Then,

(1) Y is irreducible.
(2) $\dim Y = r$.
(3) $\deg Y \leq d^r$.

Proof of Lemma 2.8. The first statement is a consequence of the fact that Y is the image of an irreducible set (namely $\mathbb{A}^n$) under a continuous map. Since $k[Y] = k[f_1, \ldots, f_m]$, we have $\text{tr.deg}(k(Y)) = r$, whence $\dim Y = r$ by definition. Here we used the fact that the dimension of an irreducible variety is the transcendence degree of its function field over the ground field. A proof of the third part can be found in [BCS97, 8.48]. □

3 MAIN RESULTS

We require a bound on the probability that a random linear hyperplane intersects a variety of a given dimension properly, that is such that the dimension of the variety decreases by exactly one. It is well known that the set of such hyperplanes form a Zariski open set in the space of all hyperplanes. We use an explicit bound on the probability of such an intersection based on the degree of the variety, both for the projective and the affine case. We will require that our intersecting hyperplanes have some structure: that their defining equations depend only on a few variables, depending on the dimension of the variety to be intersected. We establish all these facts in the next subsection. In the three subsections following that, we use this lemma to prove our three main results– Theorem 1.1, Theorem 1.2, and Theorem 1.3.

3.1 Intersection by a hyperplane

Lemma 3.1. *Let $V \subseteq \mathbb{P}^n$ be a projective variety of dimension r and degree D. Let S be a finite subset, of the underlying field k, not containing 0. Let ℓ be a linear form in $x_0, x_1, \ldots, x_{n-r}$ with each coefficient picked uniformly and independently from S. Let H be the hyperplane defined by ℓ. Then, with probability at least $1 - D/|S|$ we have* $\dim V \cap H = \dim V - 1$.

Analogously, if $V \subseteq \mathbb{A}^n$ is affine, ℓ is a linear polynomial in $x_1, \ldots, x_{n-r+1}$ and H its hyperplane; then $\dim V \cap H = \dim V - 1$ *with probability at least* $1 - 2D/|S|$.

Proof of Lemma 3.1. First we prove the projective case. Let $\ell := c_0 x_0 + \cdots + c_{n-r} x_{n-r}$, where the c_i are the coefficients picked uniformly at random from S. Let $\cup_{j=1}^{d} V_j$ be the decomposition of the dimension-r part of V into irreducible components. Then by definition, $\deg V \geq \sum \deg V_j$, and hence $d \leq D$. Pick a point p_j in V_j, for each j. We can always pick p_j so that not all of its first $n - r + 1$ coordinates are zero: if this was not possible then V_j would have to be contained in the variety defined by $x_0 = x_1 = \cdots = x_{n-r} = 0$, which has dimension only $r - 1$. By Theorem 2.2, $\dim H \cap \dim V_j = \dim V_j$ if and only if $V_j \subseteq H$ (since V_j and H are irreducible), and otherwise $\dim H \cap V_j = \dim V_j - 1$. The probability that this happens is upper bounded by the probability that $p_j \in H$. For a fixed j, this is equivalent to $\ell(p_j) = 0$. Since not all of the first $n - r + 1$ coordinates of p_j are zero, the above is bounded by $1/|S|$, by fixing all but one of the coordinates. By a union bound, with probability at most $d/|S|$, there exists some j where $\dim H \cap V_j = \dim V_j$. Therefore, with probability at least $1 - D/|S|$, we get $\dim V_j \cap H = \dim V_j - 1$ for every j, whence $\dim V \cap H = \dim V - 1$.

Now suppose V is affine. The difference from the projective case is that the intersection $V \cap H$ might be empty, and we need to bound the probability of this event. Let V^p be its projective closure. Then $\dim V^p = \dim V$ and $\deg V^p = \deg V$. By the previous part, we have $\dim V^p \cap H^p = \dim V^p - 1$ with probability $1 - D/|S|$. Then, the case $V \cap H = \emptyset$ only happens if $\dim V^p \cap H^p \cap \mathbb{P}^n_\infty = \dim V^p - 1$, where $\mathbb{P}^n_\infty$ is the hyperplane $x_0 = 0$ in $\mathbb{P}^n$. The irreducible components of V are in bijection with those of V^p, and hence V^p has no irreducible component contained in $\mathbb{P}^n_\infty$. Therefore, $\dim V^p \cap \mathbb{P}^n_\infty = \dim V^p - 1$. Further, by Bézout's theorem we have $\deg V^p \cap \mathbb{P}^n_\infty \leq \deg V^p$.

Now $H^p \cap \mathbb{P}^n_\infty$ is a hyperplane in $\mathbb{P}^n_\infty$ defined by the nonconstant part of ℓ. In particular, it is a hyperplane whose defining equation has coefficients picked uniformly and independently and we can apply the projective version of this lemma on $\mathbb{P}^n_\infty$. Therefore the probability that its intersection with $V^p \cap \mathbb{P}^n_\infty$ does not result in a reduction in the dimension is at most $D/|S|$. By a union bound, with

probability at least $1-2D/|S|$ it holds that $\dim V^p \cap H^p = \dim V - 1$ and $\dim V^p \cap H^p \cap \mathbb{P}^n_\infty = \dim V - 2$, whence $\dim V \cap H = \dim V - 1$ as required. □

An important fact to note is that our choice of variables for the linear form is arbitrary. The lemma works for any choice of $n-r+1$ variables, and this will be important when we use the lemma. Also, note that the above lemma works when our linear form involves more that $n-r+1$ variables.

Repeated applications of the above allow us: (1) to reduce a variety to dimension 0 by taking hyperplane sections, and (2) to find a linear subspace that avoids the variety.

3.2 Radical membership: Proof of Thm.1.1

Using the above lemma, we complete the proof of the main theorem:

Proof of Theorem 1.1. We first assume $g = 1$, which is the Nullstellensatz problem HN. Define $D := \prod_{i=1}^m d_i$, and $V := V(\langle \mathbf{f} \rangle)$. The set of common zeroes of these polynomials is the fibre of the point $\mathbf{0}$ under the map F defined in Lemma 2.8. The problem HN is thus equivalent to testing if a particular fibre of a polynomial map is nonempty. By the fibre dimension theorem (Theorem 2.3), the codimension of the zeroset—if it is nonempty—is bounded above by the dimension of the image of the map, which by Lemma 2.8 is r. The zeroset V is therefore either empty, or has dimension at least $n-r$. Assume that V is nonempty. By repeated applications of Bézout's theorem (Theorem 2.4), $\deg V \leq D$. Let S be a subset of the underlying field k_i (or an extension) of size at least $6(n-r)D$ that does not contain 0. We can sample from S in time polynomial in d, n, m, since S has size exponential in these parameters. Further, if we were required to go to an extension to form S, the degree of the extension would be polynomial in d, n, m. Pick $n-r$ random linear polynomials $\ell_1, \ldots, \ell_{n-r}$ with coefficients from S, and call their zero sets $H_1, \ldots, H_{n-r}$ respectively. By Lemma 3.1, the intersection $V \cap H_1$ has dimension $r-1$ with probability at least $1 - 1/(3(n-r))$. Further, by Bézout's theorem we get $\deg V \cap H_1 \leq \deg V \leq D$, since each H_i has degree one. Again by Lemma 3.1, the intersection $(V \cap H_1) \cap H_2$ has dimension $r-2$ with probability at least $1 - 1/(3(n-r))$, and $\deg V \cap H_1 \cap H_2 \leq D$. Repeating this for all H_i and using the union bound, we get $\dim V \cap H_1 \cap \cdots \cap H_{n-r} \geq 0$ with probability at least $2/3$.

Therefore, when the polynomials $\mathbf{f}$ have nonempty zeroset and are restricted to the r dimensional affine subspace $\cap H_i$, the new zeroset has dimension at least 0, and in particular is nonempty. If the zeroset of the polynomials was empty to begin with, then the restriction to the linear subspace also results in an empty zeroset.

This restriction can be performed by a variable reduction, as follows. Treating $\mathbb{A}^n$ as a vector space of dimension n over k, let H_0 be the linear subspace corresponding to the affine subspace $H := \cap H_i$. H_0 has dimension r, and hence has basis $a_1, \ldots, a_r$. Further, let vector b be such that $H = H_0 + b$. Define linear forms $c_1, \ldots, c_n$ in new variables $z_1, \ldots, z_r$ as $c_i := \sum_{j=1}^r a_{ji} z_j + b_i$, where a_{ji} is the i^{th} component of a_j. Define $f_i' := f_i(c_1, \ldots, c_n)$. Then by construction, the zeroset of $f_1', \ldots, f_m'$ is equal to $V \cap (\cap H_i)$. Further, $\deg f_i' = \deg f_i$, and these polynomials are in r variables. Also, the construction of these f_i' can be done in a blackbox manner, given blackboxes for f_i. This construction takes time polynomial in m, r, n.

We now repeatedly invoke Theorem 2.6 to check if f_i's have a common root. First we must convert them to a sparse representation. The polynomial f_i' has at most $\binom{r+d_i}{r}$ many monomials, and therefore we can find every coefficient in time polynomial in $\binom{r+d_i}{r}$ by simply solving a linear system. Applying Theorem 2.6, we can test whether the dimension of the zeroset of $f_1', \ldots, f_m'$ is 0 or not. However, we want to check if the dimension is at least 0. For this, we randomly sample r more hyperplanes $H_1', \ldots, H_r'$ as in the previous part of the proof, this time in the new variables $z_1, \ldots, z_r$. Let V' be the zeroset of $f_1', \ldots, f_m'$. We first use Theorem 2.6 to check if V' has dimension 0. If not, then we check if $V' \cap H_1'$ has dimension 0. If not, then we check $V' \cap H_1' \cap H_2'$, and so on. We return success if any one of the above iterations returns success (implying that the corresponding variety has dimension 0). Performing calculations similar to the ones earlier in the proof, we see that with high probability each intersection reduces the dimension by 1. If V' originally had dimension r', then after intersecting with r' hyperplanes, the algorithm of Theorem 2.6 returns success. If V' was empty, then the algorithm does not return success in any of the above iterations. This allows us to decide if V' has dimension at least 0. Finally, using the fact that the dimension of the zeroset of $f_1', \ldots, f_m'$ is at least 0 if and only if $\dim V \geq 0$, we get the required algorithm for HN.

We now estimate the time taken. Computing the dense representation takes time polynomial in d^r and m. Each of the at most r applications of Theorem 2.6 also take the same amount of time. The sampling steps take time polynomial in $\log nD$ (in turn polynomial in d, m) and only requires an extension of degree polynomial in n and $\log d$. The total time taken is therefore polynomial in m, d^r.

Now assume that g is an arbitrary polynomial. We reduce the problem to the case of $g = 1$ using Rabinowitsch trick [Rab30]. The polynomial g belongs to the radical of the ideal $\langle \mathbf{f} \rangle$ if and only if the polynomials $\mathbf{f}, 1 - yg$ have no common root (here y is a new variable). Further, if $\mathbf{f}$ have transcendence degree r, then the set $\mathbf{f}, 1 - yg$ has transcendence degree $r+1$. We therefore reduce the radical membership problem to HN problem, with a constant increase in the transcendence degree, number of polynomials and the number of variables. By the result in the previous paragraph, we can solve this in time polynomial in n, m and d^r. □

3.3 Effective Nullstellensatz: Proof of Thm.1.2

We now prove that by taking random linear combinations of the input polynomials, we can reduce the number of polynomials to be one more than the transcendence degree while preserving the existence of roots. This reduction gives degree bounds for the Nullstellensatz certificates. Note that this reduction does not help in Section 3.2's root-testing procedure, since we will only be saving a factor in m if we reduce the number of polynomials.

Theorem 3.2 (Generator reduction). *Let $f_1, \ldots, f_m$ be polynomials, in $x_1, \ldots, x_n$, of degrees atmost d and of tr.deg= r. Let $g_1, \ldots, g_{r+1}$ be polynomials defined as $g_i := \sum_{j=i}^m c_{ij} f_j$, where each c_{ij} is randomly picked from a finite subset S of k. Then with probability at least $1 - d^{(r+1)m}/|S|$, we have $V(\langle \mathbf{f} \rangle) = V(\langle \mathbf{g} \rangle)$.*

That we pick the linear combinations so that the first involves all polynomials, the second involves all except f_1, the third involves all except f_1, f_2 and so on is crucial for the improvement in the degree bounds.

Proof of Theorem 3.2. We prove this by studying the set Y defined in Lemma 2.8. Let $F : \mathbb{A}^n \to \mathbb{A}^m$ be the map with coordinate functions f_i. Let $Y := \overline{F(\mathbb{A}^n)}$, the closure of the image of F in $\mathbb{A}^m$. We use $y_1, \ldots, y_m$ to denote the coordinate functions of $\mathbb{A}^m$. By Lemma 2.8, Y has dimension r, and degree at most $D := d^r$. Let Y^p be the projective closure of Y. Then Y^p also has dimension r and degree at most D. Let $\ell_1, \ldots, \ell_{r+1}$ be the linear polynomials $\ell_i := \sum_{i \le j \le m} c_{ij} y_j$.

Consider the subspace defined by $y_0, \ell_1, \ldots, \ell_r$ in $\mathbb{P}^m$. The variety $Y^p \cap \mathbb{P}^m_\infty$, which is the intersection of Y^p with the hyperplane at infinity defined by $y_0 = 0$, has dimension $r-1$. Since $\ell_1, \ldots, \ell_r$ are random linear polynomials and $Y^p \cap \mathbb{P}^m_\infty$ is a variety of, degree at most D and, dimension $r-1$, we can repeatedly apply Lemma 3.1 to get a bound on the probability of proper intersections. Let H_i be the hyperplane defined by ℓ_i. We apply Lemma 3.1 starting from H_r. The equation ℓ_r has $m - r + 1$ coefficients, and therefore satisfies the conditions required for the lemma. By Bézout's theorem, the intersection has degree bounded by D, and dimension decreased by one. We then apply the theorem with H_{r-1} and so on, as in the proof of Theorem 1.1. In each iteration the variety considered has one less dimension than the previous iteration, but our linear polynomial has one more variable, and therefore we will always satisfy the conditions of Lemma 3.1.

We can now invoke Theorem 2.1 to say that the map $\mathbb{P}^m \to \mathbb{P}^r$ with coordinate functions $(y_0, \ell_1, \ldots, \ell_r)$ is Noether normalizing for Y^p. We call this map L'. We use $z_0, \ldots, z_r$ to denote the coordinate functions of $\mathbb{P}^r$. The map L' sends the affine chart $y_0 \neq 0$ to the affine chart $z_0 \neq 0$. Let L be the restriction of L' to this affine chart. Then L defines a map from $\mathbb{A}^m$ to $\mathbb{A}^r$, which is Noether normalizing for the variety Y; we also call this restricted map L. More explicitly, the map L has coordinate functions $(\ell_1, \ldots, \ell_r)$. Also, let the map $\mathbb{A}^m \to \mathbb{A}^{r+1}$ with coordinate functions $(\ell_1, \ldots, \ell_{r+1})$ be labelled M.

Since the map L is Noether normalizing, it has finite fibres. Let Q be the fibre of $\mathbf{0}$ in Y. We bound the size of this set. The map L is Noether normalizing, and hence it is surjective. The image $\mathbb{A}^r$ is normal, and hence the cardinality $|Q|$ of the fibre is bounded by the degree of the map [SR13, Theorem 2.28]. Here, by the degree of the map we mean the degree of $k(Y)$ over the pullback $L^*(k(\mathbb{A}^r))$. Note that $k(Y) = k(f_1, \ldots, f_m)$, and $L^*(k(\mathbb{A}^r)) = k(\ell_1(\mathbf{f}), \ldots, \ell_r(\mathbf{f}))$ after applying the same isomorphism. By Perron's bound, for each i there exists an annihilator of $f_i, l_1(\mathbf{f}), \ldots, l_r(\mathbf{f})$ of degree at most d^{r+1}. The degree of the extension, and hence $|Q|$, is bounded by $d^{m(r+1)}$.

Further, no point of Q, other than $\mathbf{0}$, has all of the last $m - r$ coordinates as zero. This follows from the fact that $L^{-1}(\mathbf{0})$ is a linear space of dimension $m - r$, and its intersection with $y_{r+1} = y_{r+2} = \cdots = y_m = 0$ has dimension 0. Consider now the linear form ℓ_{r+1}. For every $\mathbf{0} \neq q \in Q$, the probability that $\ell_{r+1}(q) = 0$ is at most $1/|S|$. Therefore, with probability at least $1 - d^{m(r+1)}/|S|$, the polynomial ℓ_{r+1} is nonzero on every nonzero point of Q.

Consider the polynomials $g_1, \ldots, g_{r+1}$, and let G be the polynomial map $\mathbb{A}^n \to \mathbb{A}^{r+1}$ with coordinate functions g_i. By the choice of ℓ_i in the previous paragraph, the map G is exactly the composition of the map $F : \mathbb{A}^n \to \mathbb{A}^m$ with $M : \mathbb{A}^m \to \mathbb{A}^{r+1}$. Let Q be as defined earlier, the fibre of $\mathbf{0}$ under L. By construction, the set $M^{-1}(\mathbf{0})$ is a subset of Q. But since the polynomial ℓ_{r+1} is nonzero on every nonzero point of Q, the set $M^{-1}(\mathbf{0})$ consists only of $\mathbf{0}$. Therefore, $F^{-1}(M^{-1}(\mathbf{0})) = F^{-1}(\mathbf{0})$. Since $G = M \circ F$ we get $G^{-1}(\mathbf{0}) = F^{-1}(\mathbf{0})$; which is the same as $V(\langle \mathbf{f} \rangle) = V(\langle \mathbf{g} \rangle)$. □

We use the above to prove our 2nd main result:

Proof of Theorem 1.2. Using Theorem 3.2, there exists polynomials $g_1, \ldots, g_{r+1}$ of degrees $d_1, \ldots, d_{r+1}$ that do not have a common root. By Theorem 2.5, there exist $h'_1, \ldots, h'_{r+1}$ such that $\deg g_i h'_i \le \prod_{i=1}^{r+1} d_i$ such that $\sum g_i h'_i = 1$. In this equation, substitute back the linear-combination of $f_1, \ldots, f_m$ for each g_i; whence we get the required h_i's. □

3.4 Computing tr.deg: Proof of Thm.1.3

We give a method of 'efficiently' computing the tr.deg of input polynomials $f_1, \ldots, f_m$. By Lemma 2.8 and the second part of Theorem 2.3, the tr.deg can be computed if we know the dimension of a general fibre. We need to get a bound on the points that violate the equality in Theorem 2.3. For this we follow the classical proof of the theorem and give effective bounds wherever required. For convenience we have provided a proof sketch in Appendix A, for the special case we need.

Lemma 3.3. *Let $h_1, \ldots, h_m$ be polynomials of degree at most d in n variables, and let W be the Zariski closure of the image of the map $\mathbf{h}$ with coordinates h_i. Let $S \subset k$ be of size $6nd^n$. If $a_1, \ldots, a_n$ are randomly picked from S, then with probability at least $5/6$, the fibre of $(h_1(\mathbf{a}), \cdots, h_m(\mathbf{a}))$ has codimension exactly $\dim W$.*

Proof. First assume that the h_i are algebraically *independent*. Then $W = \mathbb{A}^m$. Let the input variables be labelled such that $x_1, \ldots, x_{n-m}, h_1, \ldots, h_m$ are algebraically independent, and let $A_j(z_0, z_1, \ldots, z_{n-m}, w_1, \ldots, w_m)$ be the (minimal) annihilator of x_j over this set of variables, that is $A_j(x_j, x_1, \ldots, x_{n-m}, h_1, \ldots, h_m) = 0$. By the proof of Theorem 2.3 (Appendix A), a sufficient condition for point $a_1, \ldots, a_n$ to be such that $\mathbf{h}(\mathbf{a})$ has fibre of dimension exactly $n - m$ is that $A_j(x_j, x_1, \ldots, x_{n-m}, h_1(\mathbf{a}), \ldots, h_m(\mathbf{a}))$ is a nonzero polynomial. The polynomial A_j, when treated as polynomials in variables $z_0, \ldots, z_{n-m}$ with coefficients in $k[w_1, \ldots, w_m]$ are such that the leading monomial has coefficient a polynomial in $w_1, \ldots, w_m$ of weighted-degree at most $\prod_{i=1}^m d_i$ (by Perron bound). By the polynomial identity lemma [Ore22, DL78, Sch80, Zip79], if we pick each a_i randomly from a set of size $6 \prod_{i=1}^m d_i$ then, with probability at least $5/6$, none of the polynomials $A_j(x_j, x_1, \ldots, x_{n-m}, h_1(\mathbf{a}), \ldots, h_m(\mathbf{a}))$ is zero. In this case, the codimension of the fibre of $\mathbf{h}(\mathbf{a})$ is exactly m as claimed.

In the general case, the h_i may be algebraically dependent, and W is a subvariety of $\mathbb{A}^m$. Suppose $\dim W = \text{tr.deg}(\mathbf{h}) =: s$. Then we take s many random linear combinations g_i of the h_i, as in the proof of Theorem 1.2. The map defined by the g_i is dense in $\mathbb{A}^s$ and therefore the g_i ($i \in [s]$) are algebraically independent. By the previous paragraph, point $\mathbf{a}$ picked coordinatewise from S is such that the fibre of $\mathbf{g}(\mathbf{a})$ has codimension s. The fibre of $\mathbf{h}(\mathbf{a})$ is

a subset of the fibre of $\mathbf{g}(\mathbf{a})$, and therefore it has codimension at least s. Finally, by Theorem 2.3, the fibre has codimension at most s, whence the fibre of $\mathbf{h}(\mathbf{a})$ has codim $= s$ as required. □

Proof of Theorem 1.3. For each i, upwards from 1 to n, we do the following steps. We iterate till i reaches tr.deg r of the m polynomials. In the i-th iteration, we intersect $\mathbb{A}^n$ with $n-i$ random hyperplanes $\ell_1, \ldots, \ell_{n-i}$, as in the proof of Theorem 1.1 (Sec.3.2). Here, the coefficients are picked from a set S of size at least $n \cdot 18 \prod_{i=1}^{m} d_i$. We therefore reduce the problem to i variables.

Randomly pick point $\mathbf{a}$ where each coordinate (of the n many) is picked randomly from S. By Lemma 3.3 (& 3.1), with error-probability $\leq 1/6n$, the point $\mathbf{f}(\mathbf{a})$ has intersected fibre of dimension $(n-r)-(n-i) = (i-r)$. We need to check this algorithmically; which is done by interpolating the polynomials $\mathbf{f}$ after hyperplane intersections, and then using Theorem 2.6 (as detailed in Sec.3.2). If the intersected fibre dimension is zero, we have certified tr.deg $= i = r$; so we halt and return i as output. Else, we move to the next $i \mapsto i+1$. The interpolation step above is performed by solving a linear system which has size polynomial in d^i which is the count of the monomials of degree at most d in i variables.

Note that for $i < r$, with error-probability $\leq 1/6n$, the fibre of $\mathbf{f}(\mathbf{a})$ has an empty intersection with $\ell_1, \ldots, \ell_{n-i}$; which is dim$= -1$ and hence gets verified by Theorem 2.6.

By a union bound therefore, with error-probability $\leq 1/6$, the above algorithm gives the correct answer. For each i, the time complexity of the above steps is polynomial in d^i, m, which is the time taken for the interpolation step and to verify zero-dimension of the fibre. Therefore the algorithm as a whole takes time polynomial in d^r, n, m as claimed. □

4 CONCLUSION

We give algorithms for radical membership and tr.deg of systems of polynomials, in time that depends on the tr.deg. In both cases, our algorithms generalize the cases of 'few' input polynomials. We further give bounds on the degree of the Nullstellensatz certificates that depend on the tr.deg of the input polynomials. In all three cases, our bounds are significantly better than the previously known results in the regime when the tr.deg is much smaller than the number of variables and the number of polynomials.

Our work leaves the natural open problem of designing efficient algorithms when the tr.deg is 'larger'.

- For the blackbox radical membership problem, given the NP-hardness of HN, it is unlikely that a significantly better algorithm exists (unless other restrictions are put on the input polynomials).
- Could our methods, and the core hard instance thus identified, help in proving that HN is in AM? Currently, this is known only partially [Koi96].
- For the tr.deg problem however, we know that the problem is in coAM ∩ AM, making it unlikely to be NP hard. It is therefore likely that there is an efficient randomized algorithm whose time complexity is polynomial in n and m. This is already known in the case when the field has large/zero characteristic, and it is an open problem to extend this to other fields. A first step might be to give a subexponential time algorithm for the problem that works without any assumptions.

Acknowledgements. We thank Ramprasad Saptharishi for introducing us to the universality of 3-generators ideal membership problem. Nitin Saxena thanks the funding support from DST (DST/SJF/MSA-01/2013-14).

REFERENCES

[AB09] Sanjeev Arora and Boaz Barak. *Computational complexity: a modern approach*. Cambridge University Press, 2009.

[ASSS12] M. Agrawal, C. Saha, R. Saptharishi, and N. Saxena. Jacobian hits circuits: Hitting-sets, lower bounds for depth-D occur-k formulas & depth-3 transcendence degree-k circuits. In *Proceedings of the 44th ACM Symposium on Theory of Computing (STOC)*, pages 599–614, 2012. (SICOMP spl.issue, 45(4), 1533–1562, 2016).

[BCS97] Peter Bürgisser, Michael Clausen, and Mohammad Amin Shokrollahi. *Algebraic complexity theory*, volume 315 of *Grundlehren der mathematischen Wissenschaften*. Springer, 1997.

[BMS13] M. Beecken, J. Mittmann, and N. Saxena. Algebraic independence and blackbox identity testing. *Information and Computation*, 222:2 – 19, 2013. (Also, 38th International Colloquium on Automata, Languages and Programming, ICALP 2011).

[Bro87] W. Dale Brownawell. Bounds for the degrees in the Nullstellensatz. *Annals of Mathematics*, 126(3):577–591, 1987.

[BS91] Carlos A. Berenstein and Daniele C. Struppa. Recent improvements in the complexity of the effective Nullstellensatz. *Linear Algebra and its Applications*, 157:203 – 215, 1991.

[Buc65] B. Buchberger. *Ein Algorithmus zum Auffinden der Basiselemente des Restklassenringes nach einem nulldimensionalen Polynomideal*. PhD thesis, University of Innsbruck, 1965.

[CLO07] David A. Cox, John Little, and Donal O'Shea. *Ideals, Varieties, and Algorithms: An Introduction to Computational Algebraic Geometry and Commutative Algebra, 3/e (Undergraduate Texts in Mathematics)*. Springer-Verlag, Berlin, Heidelberg, 2007.

[Csa75] L. Csanky. Fast parallel matrix inversion algorithms. *16th Annual Symposium on Foundations of Computer Science (SFCS 1975)*, pages 11–12, 1975.

[DFGS91] Alicia Dickenstein, Noaï Fitchas, Marc Giusti, and Carmen Sessa. The membership problem for unmixed polynomial ideals is solvable in single exponential time. *Discrete Applied Mathematics*, 33(1-3):73–94, 1991.

[DGW07] Z. Dvir, A. Gabizon, and A. Wigderson. Extractors and rank extractors for polynomial sources. In *48th Annual IEEE Symposium on Foundations of Computer Science (FOCS'07)*, pages 52–62, Oct 2007.

[DL78] Richard A. Demillo and Richard J. Lipton. A probabilistic remark on algebraic program testing. *Information Processing Letters*, 7(4):193 – 195, 1978.

[GH94] Marc Giusti and Joos Heintz. La détermination des points isolés et de la dimension d'une variété algébrique peut se faire en temps polynomial. *Computational Algebraic Geometry and Commutative Algebra*, 34, 02 1994.

[GSS18] Zeyu Guo, Nitin Saxena, and Amit Sinhababu. Algebraic dependencies and pspace algorithms in approximative complexity. In *Proceedings of the 33rd Computational Complexity Conference*, CCC '18, 2018.

[Gup14] Ankit Gupta. Algebraic geometric techniques for depth-4 PIT & Sylvester-Gallai conjectures for varieties. *Electronic Colloquium on Computational Complexity (ECCC)*, 21:130, 2014.

[Hei83] Joos Heintz. Definability and fast quantifier elimination in algebraically closed fields. *Theoretical Computer Science*, 24(3):239 – 277, 1983.

[Her26] Grete Hermann. Die Frage der endlich vielen Schritte in der Theorie der Polynomideale. *Mathematische Annalen*, 95(1):736–788, Dec 1926.

[Jel05] Zbigniew Jelonek. On the effective Nullstellensatz. *Inventiones mathematicae*, 162(1):1–17, Oct 2005.

[Kay09] N. Kayal. The complexity of the annihilating polynomial. In *24th Annual IEEE Conference on Computational Complexity*, pages 184–193, July 2009.

[Koi96] Pascal Koiran. Hilbert's Nullstellensatz is in the polynomial hierarchy. *J. Complexity*, 12(4):273–286, 1996.

[Kol88] János Kollár. Sharp effective Nullstellensatz. *Journal of the American Mathematical Society*, 1(4):963–975, 1988.

[KPS99] Teresa Krick, Luis Miguel Pardo, and Martín Sombra. Arithmetic Nullstellensätze. *ACM SIGSAM Bulletin*, 33(3):17, 1999.

[KPS+01] Teresa Krick, Luis Miguel Pardo, Martín Sombra, et al. Sharp estimates for the arithmetic Nullstellensatz. *Duke Mathematical Journal*, 109(3):521–598, 2001.

[Kru50] Wolfgang Krull. Jacobsonsches Radikal und Hilbertscher Nullstellensatz. In *Proceedings of the International Congress of Mathematicians, Cambridge, Mass*, volume 2, pages 56–64, 1950.

[Lap06] Santiago Laplagne. An algorithm for the computation of the radical of an ideal. In *Proceedings of the International Symposium on Symbolic and Algebraic Computation*, pages 191–195, 2006.

[Laz77] Daniel Lazard. Algèbre linéaire sur $k[x_1, \ldots, x_n]$ et élimination. *Bulletin de la Société Mathématique de France*, 105:165–190, 1977.

[Laz81] Daniel Lazard. Résolution des Systèmes d'Équations algébriques. *Theoretical Computer Science*, 15(1):77 – 110, 1981.

[LL91] Y. N. Lakshman and Daniel Lazard. *On the Complexity of Zero-dimensional Algebraic Systems*, pages 217–225. Birkhäuser Boston, 1991.

[May89] Ernst Mayr. Membership in polynomial ideals over Q is exponential space complete. In B. Monien and R. Cori, editors, *STACS 89*, pages 400–406, Berlin, Heidelberg, 1989. Springer Berlin Heidelberg.

[May97] Ernst W. Mayr. Some complexity results for polynomial ideals. *J. Complexity*, 13(3):303–325, 1997.

[MM82] Ernst W Mayr and Albert R Meyer. The complexity of the word problems for commutative semigroups and polynomial ideals. *Advances in Mathematics*, 46(3):305 – 329, 1982.

[Ore22] Øystein Ore. Über höhere kongruenzen. *Norsk Mat. Forenings Skrifter*, 1(7):15, 1922.

[Oxl06] James G. Oxley. *Matroid Theory (Oxford Graduate Texts in Mathematics)*. Oxford University Press, Inc., USA, 2006.

[Per51] O. Perron. *Algebra: Die Grundlagen.* Number v. 1 in Göschens Lehrbücherei : 1. Gruppe, Reine u. angewandte Mathematik. Walter de Gruyter & Company, 1951.

[PSS16] Anurag Pandey, Nitin Saxena, and Amit Sinhababu. Algebraic independence over positive characteristic: New criterion and applications to locally low algebraic rank circuits. In *41st International Symposium on Mathematical Foundations of Computer Science, MFCS 2016, August 22-26, 2016 - Kraków, Poland*, pages 74:1–74:15, 2016. (Comput.Compl., 27(4), 617–670, 2018).

[Pł05] Arkadiusz Płoski. Algebraic dependence of polynomials after O.Perron and some applications. *Computational Commutative and Non-Commutative Algebraic Geometry*, pages 167–173, 2005.

[Rab30] JL Rabinowitsch. Zum Hilbertschen Nullstellensatz. *Mathematische Annalen*, 102(1):520–520, 1930.

[Sap19] Ramprasad Saptharishi. Private Communication, 2019.

[Sch80] J. T. Schwartz. Fast probabilistic algorithms for verification of polynomial identities. *J. ACM*, 27(4):701–717, October 1980.

[Sin19] Amit Kumar Sinhababu. *Power series in complexity: Algebraic Dependence, Factor Conjecture and Hitting Set for Closure of VP*. PhD thesis, Indian Institute of Technology Kanpur, 2019.

[Som97] Martín Sombra. Bounds for the Hilbert function of polynomial ideals and for the degrees in the Nullstellensatz. *Journal of Pure and Applied Algebra*, 117-118:565 – 599, 1997.

[Som99] Martín Sombra. A sparse effective Nullstellensatz. *Advances in Applied Mathematics*, 22(2):271 – 295, 1999.

[SR13] I.R. Shafarevich and M. Reid. *Basic Algebraic Geometry 1: Varieties in Projective Space*. SpringerLink : Bücher. Springer Berlin Heidelberg, 2013.

[Zar47] Oscar Zariski. A new proof of Hilbert's Nullstellensatz. *Bulletin of the American Mathematical Society*, 53(4):362–368, 1947.

[Zip79] Richard Zippel. Probabilistic algorithms for sparse polynomials. In *Proceedings of the International Symposiumon on Symbolic and Algebraic Computation*, EUROSAM '79, page 216–226, Berlin, Heidelberg, 1979. Springer-Verlag.

Sparse Multiplication for Skew Polynomials

Mark Giesbrecht
Cheriton School of Computer Science
University of Waterloo
mwg@uwaterloo.ca

Qiao-Long Huang
Research Center for Mathematics and Interdisciplinary Sciences Shandong University
huangqiaolong@sdu.edu.cn

Éric Schost
Cheriton School of Computer Science
University of Waterloo
eschost@uwaterloo.ca

Abstract

Consider the skew polynomial ring $L[x;\sigma]$, where L is a field and σ is an automorphism of L of order r. We present two randomized algorithms for the multiplication of *sparse* skew polynomials in $L[x;\sigma]$.

The first algorithm is Las Vegas; it relies on evaluation and interpolation on a normal basis, at successive powers of a normal element. For inputs $A, B \in L[x;\sigma]$ of degrees at most d, its expected runtime is $O^{\sim}(\max(d,r)rR^{\omega-2})$ operations in K, where $K = L^{\sigma}$ is the fixed field of σ in L and $R \leq r$ is the size of the Minkowski sum $\mathrm{supp}(A) + \mathrm{supp}(B)$ taken modulo r; here, the supports $\mathrm{supp}(A), \mathrm{supp}(B)$ are the exponents of non-zero terms in A and B.

The second algorithm is Monte Carlo; it is "super-sparse", in the sense that its expected runtime is $O^{\sim}(\log(d)Sr^{\omega})$, where S is the size of $\mathrm{supp}(A) + \mathrm{supp}(B)$. Using a suitable form of Kronecker substitution, we extend this second algorithm to handle multivariate polynomials, for certain families of extensions.

Keywords

Sparse polynomials; skew polynomials; multiplication

ACM Reference Format:
Mark Giesbrecht, Qiao-Long Huang, and Éric Schost. 2020. Sparse Multiplication for Skew Polynomials. In *International Symposium on Symbolic and Algebraic Computation (ISSAC '20), July 20–23, 2020, Kalamata, Greece.* ACM, New York, NY, USA, 8 pages. https://doi.org/10.1145/3373207.3404058

1 Introduction

Skew polynomial rings were introduced by Ore [24] as a non-commutative generalization of usual commutative polynomial rings. They have found numerous applications, as they allow one to work with linear differential equations, shift equations, or operators over finite fields, in an algebraic manner.

A very common construction is the following: let $K \subset L$ be finite fields and let $\sigma : L \to L$ be a K-automorphism of L, that is, a power of the qth power Frobenius automorphism, with $q = \#K$. For an indeterminate x over L, the ring $L[x;\sigma]$ of skew polynomials over L is the L-vector space of finite sums $A = \sum_{0 \leq i \leq d} a_i x^i$, with all a_i's in L, endowed with the usual addition, and where multiplication is determined by the commutation relation $xc = \sigma(c)x$ for any c in L. The *degree* $\deg(A)$ of A is the largest index i for which a_i is non-zero.

In particular, if σ is the qth power Frobenius automorphism itself, $L[x;\sigma]$ is isomorphic to the ring of linearized polynomials over K (endowed with addition and composition). Fundamental algorithms for such rings are presented in [14]. These polynomials can be used to construct algebraic codes [4, 5, 9, 27], have applications in cryptography [3, 32], underlie the construction of finite Drinfeld modules [17], etc.

In this paper, our framework is slightly more general: we assume that L is any field endowed with an automorphism σ, we let $K = L^{\sigma}$, and we assume that σ has finite order r; the rest of the definition is then as above. In particular, L is a separable extension of K, with $[L : K] = r$. For a list of examples that goes beyond finite fields, see Section 1 in [6].

We are interested in the cost of multiplying such skew polynomials. Given A and B in $L[x;\sigma]$ of degree at most d, the standard "schoolbook" multiplication algorithm uses $O(d^2)$ arithmetic operations $+, \times$ in L, and $O(d^2)$ applications of powers of σ. In [25, 26], Puchinger and Wachter-Zeh improved this to $O^{\sim}(d^{(\omega+1)/2})$ arithmetic operations in L and applications of powers of σ; here ω is such that over any ring, square matrix multiplication in size s can be done in $O(s^{\omega})$ ring operations. The best known value to date is $\omega \leq 2.373$ [7, 10], giving $(\omega + 1)/2 \leq 1.69$, hence resulting in a subquadratic bound in d.

However, this analysis overlooks the (non-trivial) question of how operations in L are actually implemented. In this paper, we will measure runtimes in terms of operations in K, using the structure of L as a K-vector space; this will be our main cost measure, but we will also count bit operations when warranted (when non-trivial operations on exponents take place, for instance).

As in [6], we will use two K-bases for L. The first one, written $\mathscr{W} = (\omega_0, \ldots, \omega_{r-1})$, is taken such that addition, multiplication and inversions in L use $O^{\sim}(r)$ operations $(+, \times, \div)$ in K; here, the "soft-Oh" notation indicates that we omit polylogarithmic factors in r. For instance, if L is given as $L = K[z]/f(z)$, for some $f \in K[z]$ of degree r, then we can take ω_i to be the residue class of z^i for all i. This will be called the working basis; our convention is that *the inputs and outputs of all algorithms will be given on this basis.*

The second basis will be a *normal basis* $\mathscr{N} = (\nu_0, \ldots, \nu_{r-1})$, such that $\sigma(\nu_i) = \nu_{i+1 \bmod r}$ for all i. In such a basis, addition and application of any power of σ take linear time $O(r)$. In our algorithms, we make the following assumption about the availability of representational data for a normal basis of L/K:

ISSAC '20, July 20–23, 2020, Kalamata, Greece

ACM ISBN 978-1-4503-7100-1/20/07…$15.00
https://doi.org/10.1145/3373207.3404058

(H): the bases $\mathscr{W}$ and $\mathscr{N}$, as well as the matrices $\boldsymbol{M}_{\mathscr{N}\to\mathscr{W}}$ and $\boldsymbol{M}_{\mathscr{W}\to\mathscr{N}}$ of change of basis between $\mathscr{W}$ and $\mathscr{N}$, are given.

In this context, Caruso and Le Borgne [6] give a Las Vegas algorithm for multiplication in $L[x,\sigma]$ of expected cost $O^{\sim}(dr^{\omega-1})$ operations in K when $d \geq r$; for $d \leq r$, they propose another algorithm, whose cost is $O(d^{\omega-2}r^2)$ operations in K. Note that this paper also assumes that in the basis $\mathscr{W}$, the application of σ takes quasi-linear time, that is, $O^{\sim}(r)$ operations in K. This is a reasonable assumption when K and L are finite fields, as [8] [1] show that any finite field extension of a finite field admits a basis in which the operations addition, multiplication, division and application of σ cost $O^{\sim}(r)$ operations in K.

However, in our context, we show that this assumption can be dropped. This gives the advantage of more flexibility in choosing the working basis, so we will not make such an assumption.

Note that we will not address the problem of *finding* a normal basis; this has been widely studied, and we refer the reader to [11, 15, 16, 19] and references therein.

The previous discussion assumes that the input A and B are "dense polynomials", that is, given by the array of all their coefficients; in this case, in degree d, input and output size are $\Theta(dr)$ elements in K, so Caruso and Le Borgne's result of $O^{\sim}(dr^{\omega-1})$ operations in K is close to optimal (and would be optimal if we could take $\omega = 2$). In this current paper, we revisit this question, taking into account the "sparsity" of A and B. Following [2], we define the following, for a polynomial $A = \sum_{i=1}^{t} a_i x^{e_i}$, in $L[x;\sigma]$ with $0 \leq e_1 < \cdots < e_t$ and all a_i non-zero:

- the *sparsity* $\#A$ is the number t in the expression above;
- the *support* $\mathrm{supp}(A)$ is the set of exponents $\{e_1,\ldots,e_t\} \subset \mathbb{N}$.

For two polynomials A and B, we have the inequalities

$$\#(AB) \leq \#\mathbb{S}(A,B) \leq \#A \cdot \#B,$$

where $\mathbb{S}(A,B)$ is the Minkowski sum

$$\mathbb{S}(A,B) := \{e_A + e_B \mid e_A \in \mathrm{supp}(A), e_B \in \mathrm{supp}(B)\}. \tag{1.1}$$

A strict inequality $\#(AB) < \#\mathbb{S}(A,B)$ occurs only in the presence of coefficient cancellations. We will often write $S := \#\mathbb{S}(A,B)$. Recalling that r is the order of σ, we will also define

$$\mathbb{S}_r(A,B) := \{(e_A+e_B) \bmod r \mid e_A \in \mathrm{supp}(A), e_B \in \mathrm{supp}(B)\}. \tag{1.2}$$

After we discuss reduction modulo central elements, we will see that $\mathbb{S}_r(A,B)$ contains the support of the polynomial $AB \bmod (x^r-1)$. If we write $R := \#\mathbb{S}_r(A,B)$, this means that we have $\#(AB \bmod (x^r-1)) \leq R$; note also the inequalities $R \leq r$ and $R \leq S$.

In this paper, we give two randomized algorithms for multiplying skew polynomials in $L[x;\sigma]$. The first one is Las Vegas; for inputs of degree at most d, it uses an expected $O^{\sim}(\max(d,r)rR^{\omega-2})$ operations in K, where R is as above. This algorithm is based on Caruso and Le Borgne's [6]; as in that reference, the whole multiplication procedure reduces to several instances of multiplication modulo x^r-1. Whereas the original algorithm uses $O^{\sim}(r^{\omega})$ operations in K for this task, ours takes $O^{\sim}(r^2R^{\omega-2})$ operations. Altogether, since $R \leq r$, our runtime is asymptotically never worse than that in [6], and can be better in many cases. Apart from this, Puchinger and Wachter-Zeh's algorithm performs the same computation with complexity $O^{\sim}(d^{(\omega+1)/2}r)$ operations. As stated in [6], the algorithm in [25, 26] is faster than the one in [6] for polynomials of small degree $d \leq r^{2/(5-\omega)}$. As $d \leq r^{2/(5-\omega)} \leq r$, in this case, our complexity is $O^{\sim}(r^2R^{\omega-2})$. So unless $d \leq \min(r^{2/(5-\omega)}, r^{2/(\omega+1)}R^{(2\omega-4)/(\omega+1)})$, our new algorithm is faster. The precise statement is as follows.

Theorem 1.1. *Let L be a field with automorphism σ of finite order r and $K = L^{\sigma}$, and assume we have representational data **(H)** for L/K as above. Given $A,B \in L[x;\sigma]$ of degrees at most d, there is a Las Vegas algorithm to compute AB with an expected cost of $O^{\sim}(\max(d,r)rR^{\omega-2})$ operations in K and $O^{\sim}(\max(d,r))$ bit operations, where $R \leq r$ is the cardinality of the set $\mathbb{S}_r(A,B)$ defined in* (1.2).

The second algorithm comes in two stages, one of which is Monte Carlo and the other Las Vegas. Overall, for a given probability of failure, its expected runtime is now polynomial in $\log(d)$, r and S, where S is the cardinality of the set $\mathbb{S}(A,B)$ defined in (1.1). Due to this logarithmic dependence in the degree d, we will call this algorithm *supersparse*.

The algorithm is inspired by the work of Arnold and Roche [2] on the multiplication of sparse commutative polynomials. We first compute $\mathbb{S}(A,B)$; once it is known, we compute at least half the coefficients of the product AB through multiplication modulo a well-chosen central polynomial of the form $x^{pr}-1$. After a logarithmic number of iterations, this gives us the whole product AB.

Theorem 1.2. *Let L be a field with automorphism σ of finite order r and $K = L^{\sigma}$, and assume we have representational data **(H)** for L/K as above. Given $A,B \in L[x;\sigma]$ of degrees at most d, and $\mu \in (0,1)$, there is an algorithm to compute AB with probability at least $1-\mu$, using an expected $O^{\sim}(\log(d)Sr^{\omega})$ operations in K and $O^{\sim}(\log(d)S(r+\log(1/\mu)))$ bit operations, where S is the cardinality of the set $\mathbb{S}(A,B)$ defined in* (1.1).

We also present a multivariate version of this algorithm whose cost is summarized as follows. Let $L[x_1,\ldots,x_n;\sigma_1,\ldots,\sigma_n]$ be a multivariate skew polynomial ring, with the relations $x_i a = \sigma_i(a)x_i$ and $x_i x_j = x_j x_i$, where each σ_i is an automorphism of L.

Theorem 1.3. *Let L be a field with automorphism σ of finite order r, $K = L^{\sigma}$ and $\sigma_i = \sigma^{e_i}$ for $0 \leq e_1,\ldots,e_n < r$, and assume we have the representation data **(H)** for L/K. Given $A,B \in L[x_1,\ldots,x_n;\sigma_1,\ldots,\sigma_n]$ of total degree at most D, and $\mu \in (0,1)$, there is an algorithm to compute AB with probability at least $1-\mu$, using an expected $O^{\sim}(nr^{\omega}S\log D)$ operations in K plus $O^{\sim}(n^2S\log D + nSr\log D + nS\log D\log(1/\mu) + S\log r\log(1/\mu))$ bit operations, where $S = \#\mathbb{S}(A,B)$.*

For historical perspective and comparison, algorithms to compute sparse multiplication of usual commutative polynomials has seen considerable research recently, both in theory and in practice. New algorithms for polynomials with at most t terms have been developed to keep the time proportional to the worst-case output size, $O(t^2)$, and low space complexity, both in theory and in practice [18, 22, 23]. This is particularly important for multivariate polynomials [30]. The aforementioned work of Arnold & Roche [2] adapts to the potential even smaller output size, and when the support is know [31] demonstrate greater improvements. See the excellent

[1] We thank the anonymous referee for pointing this out.

recent survey of [28] on the state of the art in sparse polynomial computation.

2 Sparse multiplication

In this section, we give a Las Vegas algorithm for the multiplication of sparse skew polynomials, proving Theorem 1.1. Our algorithm is based on Caruso and Le Borgne's [6]. As in that reference, the key operation is an evaluation-interpolation based multiplication algorithm modulo $x^r - 1$; the main difference is that the number of evaluations in our algorithm depends on the sparsity of the product. To build the main algorithm upon this special case, we will follow [6] with few modifications.

2.1 Preliminaries

2.1.1. Division modulo central elements. For a non-zero Z in the center of $L[x;\sigma]$, and for A in $L[x;\sigma]$, there are unique $Q, F \in L[x;\sigma]$ such that $A = QZ+F = ZQ+F$, with $F = 0$ or $\deg F < \deg Z$; we write $F = A \bmod Z$. This makes the canonical morphism

$$\begin{array}{rccc} \varepsilon: & L[x;\sigma] & \to & L[x;\sigma]/\langle Z\rangle, \\ & A & \mapsto & A \bmod Z, \end{array}$$

an endomorphism of K-algebras.

Since σ has order r, the equality $x^r c = \sigma^r(c)x^r = cx^r$ holds for all c in L. As a result, any polynomial of the form $Z = B(x^r)$, where $B \in K[x]$, is in the center of $L[x;\sigma]$ (actually, all central elements are of this form, but we won't need this). We will only use the very particular cases $Z = x^r - a$ and $Z = x^r - 1$, for which we have simple explicit formulas for the remainders. In particular, for the latter, if we consider a skew polynomial $C = c_1 x^{e_1} + \cdots + c_S x^{e_S} \in L[x;\sigma]$, with all c_i in L, then we have

$$C \bmod (x^r - 1) = c_1 x^{e_1 \bmod r} + \cdots + c_S x^{e_S \bmod r}, \qquad (2.1)$$

with $e_i \bmod r$ in $\{0, 1, \ldots, r-1\}$ for all i.

2.1.2. Scalar extension. Given $A, B \in L[x;\sigma]$, to compute the product AB, we first compute different reductions $AB \mod x^r - a_i$, where $a_i \in K$, then recover AB from these reductions by Chinese remainder algorithm. The number of reductions we need depends on the degree of the product AB. If it is large, as in [6], there may not be enough elements in ground field K, so we may have to replace K by an extension K'/K of sufficiently large cardinality. We write $s := [K' : K]$, and we assume that K' is given as $K[\xi]/g(\xi)$, for some degree-s irreducible $g \in K[\xi]$; in particular, all operations $+, \times, \div$ in K' take $O^\sim(s)$ operations in K.

We will then define $L' := L \otimes_K K'$; L' still has dimension r over K', but it does not have to be a field; it is in general a product of fields. The extension of σ to L' is the automorphism $\sigma' := \sigma \otimes_K \mathrm{id}$; it still has order r and admits K' as its fixed set.

The K-bases $\mathscr{W} = (\omega_0, \ldots, \omega_{r-1})$ and $\mathscr{N} = (\nu_0, \ldots, \nu_{r-1})$ of L extend to K'-bases $\mathscr{W}' = (\omega'_0, \ldots, \omega'_{r-1})$ and $\mathscr{N}' = (\nu'_0, \ldots, \nu'_{r-1})$ of L', with $\omega'_i = \omega_i \otimes_K 1$ and $\nu'_i = \nu_i \otimes_K 1$ for all i. In the new working basis $\mathscr{W}'$, addition, multiplication, and the inversion of invertible elements still take $O^\sim(r)$ operations $(+, \times, \div)$ in K', that is, $O^\sim(rs)$ operations in K; besides, $\mathscr{N}'$ is still a normal basis. Finally, the change-of-basis matrices between $\mathscr{W}$ and $\mathscr{N}$ still describe change-of-basis between $\mathscr{W}'$ and $\mathscr{N}'$ (but now seen as matrices over K').

To summarize, changing the ground field from K to K' affects almost nothing in our setup; the only point that will require our attention is that L' may not be a field, so non-zero elements may not be invertible.

2.2 Multiplication modulo $x^r - 1$

We start with a multiplication algorithm modulo $x^r - 1$. As explained above, we suppose that we are given a field extension K'/K of degree s, and we give an algorithm for multiplication in $L'[x;\sigma']/\langle x^r - 1\rangle$.

A skew polynomial $A \in L'[x;\sigma']$ defines a K'-linear mapping $A^* : L' \to L'$ obtained by evaluating A at σ'. For a in L', we will write $A(a)$ instead of $A^*(a)$; since σ' has order r, $A(a)$ is actually well-defined for A in $L'[x;\sigma']/\langle x^r - 1\rangle$.

This suggests an evaluation / interpolation strategy for multiplication $L'[x;\sigma']/\langle x^r - 1\rangle$. This idea is already in [6], but does not take sparsity into account there; the following algorithm achieves this, by using evaluation and interpolation at a geometric progression.

We first give the overview of the algorithm, then discuss subroutines and establish their cost bounds. Below, remember that elements of L' are always represented on the working basis $\mathscr{W}'$.

Algorithm 1: Sparse multiplication modulo $x^r - 1$.

Input: Two polynomials $A, B \in L'[x;\sigma']/\langle x^r - 1\rangle$.

Output: The product $AB \in L'[x;\sigma']/\langle x^r - 1\rangle$.

- **Step 1:** Compute $\mathbb{S}_r(A,B)$ as in (1.2) and let $R = \#\mathbb{S}_r(A,B)$.
- **Step 2:** Compute $b_i = B(\nu_0'^i)$, for $i = 0, 1, \ldots, R-1$ and let $\boldsymbol{B}$ be the $r \times R$ matrix over K' whose ith column is the coefficient vector of b_i for all i.
- **Step 3:** Compute $e_i = A(\nu'_i)$, for $i = 0, \ldots, r-1$ and let $\boldsymbol{E}$ be the $r \times r$ matrix over K' whose ith column is the coefficient vector of e_i for all i.
- **Step 4:** Compute $\boldsymbol{F} = \boldsymbol{E}\,\boldsymbol{M}_{\mathscr{W}\to\mathscr{N}}\,\boldsymbol{B}$ and let $f_0, \ldots, f_{R-1}$ be the elements of L' whose coefficient vectors are the columns of $\boldsymbol{F}$.
- **Step 5:** Return the unique polynomial $C = \sum_{\alpha \in \mathbb{S}_r(A,B)} c_\alpha x^\alpha$ such that $C(\nu_0'^i) = f_i$ for all i.

Proposition 2.1. *Under assumption* **H**, *Algorithm 1 computes the product AB using $O^\sim(R^{\omega-2}r^2 s)$ operations in K and $O^\sim(r)$ bit operations.*

Proof. Write $C = AB \in L'[x;\sigma']/\langle x^r - 1\rangle$. Since $C^* = A^* \circ B^*$, we get $C(\nu_0'^i) = A(B(\nu_0'^i))$, for $i = 0, \ldots, R-1$.

The product $\boldsymbol{E}\,\boldsymbol{M}_{\mathscr{W}\to\mathscr{N}}$ is by construction the matrix of $A^* : L' \to L'$ (in the working basis), and the columns of $\boldsymbol{B}$ are the coefficient vectors of $B(\nu_0'^i)$, for $i = 0, \ldots, R-1$, also written in the working basis. As a result, $C(\nu_0'^i) = f_i$ holds for $i = 0, \ldots, R-1$. In view of formula (2.1), we know that the support $\mathrm{supp}(C)$ is contained in $\mathbb{S}_r(A,B)$; then, we prove in §2.2.2 that Step 5 correctly recovers C.

In terms of runtime, Step 1 takes $O^\sim(r)$ bit operations (by §2.2.1 below) and Step 2 takes $O^\sim(r^2 s)$ operations in K (§2.2.2). Step 3 takes $O^\sim(r^2)$ operations in K' by [6, Prop. 1.6], which is also $O^\sim(r^2 s)$ operations in K. The cost of Step 4 is $O^\sim(R^{\omega-2}r^2 s)$ operations in K, using block matrix multiplication. Finally, Step 5 takes another $O^\sim(r^2 s)$ operations in K (§2.2.2). □

2.2.1. Computing the sumset. Given A and B as above, we show here how to compute the sumset $\mathbb{S}_r(A,B)$. Assume the supports of A, B are $\mathbb{S}_A, \mathbb{S}_B$, respectively, and let

$$\widetilde{A} = \sum_{d \in \mathbb{S}_A} y^d \in \mathbb{Z}[y], \quad \widetilde{B} = \sum_{d \in \mathbb{S}_B} y^d \in \mathbb{Z}[y]$$

be the commutative polynomials whose supports are $\mathbb{S}_A, \mathbb{S}_B$ and coefficients are all 1. To compute $\mathbb{S}_r(A,B) = \{(e_A+e_B) \bmod r \mid e_A \in \mathbb{S}_A, e_B \in \mathbb{S}_B\}$, it is enough to compute the support of $\widetilde{A}\widetilde{B} \bmod (y^r - 1)$. Using fast multiplication in $\mathbb{Z}[y]$, this takes $O^\sim(r)$ bit operations, as claimed.

2.2.2. Evaluation-interpolation at a geometric progression. Let $C = c_1 x^{e_1} + \cdots + c_t x^{e_t}$ be in $L'[x,\sigma']/\langle x^r - 1\rangle$, with $0 \le e_1 < \cdots < e_t < r$. Here we show how to evaluate C at the points $v_0'^i$, for $i = 0, 1, \ldots, R-1$, for some integer R, with $t \le R \le r$; we also show how to recover C from these values, assuming $e_1, \ldots, e_t$ are known.

For $i \ge 0$, the value $C(v_0'^i)$ is by definition $C^*(v_0'^i)$, that is,

$$\begin{aligned} C(v_0'^i) &= c_1 \sigma^{e_1}(v_0')^i + \cdots + c_t \sigma^{e_t}(v_0')^i \\ &= c_1 {v'_{e_1}}^i + \cdots + c_t {v'_{e_t}}^i. \end{aligned}$$

Taken all together for $i = 0, \ldots, R-1$, these equalities give

$$\begin{bmatrix} C(v_0'^0) \\ C(v_0'^1) \\ \vdots \\ C(v_0'^{R-1}) \end{bmatrix} = \begin{bmatrix} 1 & 1 & \cdots & 1 \\ v'_{e_1} & v'_{e_2} & \cdots & v'_{e_t} \\ \vdots & \vdots & & \vdots \\ {v'_{e_1}}^{R-1} & {v'_{e_2}}^{R-1} & \cdots & {v'_{e_t}}^{R-1} \end{bmatrix} \begin{bmatrix} c_1 \\ c_2 \\ \vdots \\ c_t \end{bmatrix}.$$

Proposition 2.2. *Given C and R as above, with $t \le R \le r$, we can compute $C(v_0'^i)$, for $i = 0, 1, \ldots, R-1$, using $O^\sim(r^2 s)$ operations in K. Given $e_1, \ldots, e_t$, we can recover $c_1, \ldots, c_t$ from these values using $O^\sim(r^2 s)$ operations in K as well.*

Proof. The matrix giving the values $C(v_0'^i)$ is transposed Vandermonde, built on the conjugates v'_{e_i}. A matrix-vector product by such a matrix takes $O^\sim(\max(R,t)) \in O^\sim(r)$ operations $+, \times$ in L'; this is $O^\sim(r^2)$ operations in K', and thus $O^\sim(r^2 s)$ operations in K.

Conversely, to recover C, we need to solve such a system (keeping only the first t rows). This takes $O^\sim(r)$ operations $+, \times$ in L' and $O(r)$ inversions - the former add up to $O^\sim(r^2 s)$ operations in K, as above. The terms we have to invert are products of the differences $v'_{e_i} - v'_{e_k}$, so they are all of the form $\alpha \otimes_K 1$, for various non-zero $\alpha \in L$, so they are all units in L'. As a result, these inversions cost a total $O^\sim(r^2)$ operations in K. □

2.3 Multiplication modulo $x^r - a$

This section follows closely [6, Sec. 2.1], with only a few minor differences; in particular, correctness of the procedure below is established in that reference.

Let K' and L' be as above, with $[K' : K] = s$, and let λ be a unit in L'. We define the *norm*

$$a := \lambda \sigma'(\lambda) \cdots \sigma'^{r-1}(\lambda),$$

Note that we know $a \in K'$ since $a = \sigma(a)$, which is why we need to extend K to K'. We now consider multiplication in $L'[x;\sigma']/\langle x^r - a\rangle$. The main idea is to reduce multiplication modulo $x^r - a$ to multiplication modulo $x^r - 1$. For this, define the L'-linear map $\delta : L'[x;\sigma'] \to L'[x;\sigma']$ by setting $\delta(x^i) = \lambda\sigma'(\lambda)\cdots\sigma'^{i-1}(\lambda)x^i$. As proved in [6], it induces an L'-algebra isomorphism $\delta : L'[x;\sigma']/\langle x^r - a\rangle \to L'[x;\sigma']/\langle x^r - 1\rangle$.

Algorithm 2: Multiplication modulo $x^r - a$.

Input:

- An element $\lambda \in L'^\times$.
- A, B in $L'[x;\sigma']/\langle x^r - a\rangle$, where $a = \lambda\sigma'(\lambda)\cdots\sigma'^{r-1}(\lambda)$.

Output: The product $AB \in L'[x;\sigma']/\langle x^r - a\rangle$.

Step 1: Compute $s_i = \sigma'^i(\lambda)$ for $i = 0, \ldots, r-1$.
Step 2: Compute $\lambda_i = s_0 \cdots s_{i-1}$ for $i = 1, \ldots, r$.
Step 3: Compute $A' = \delta(A)$ and $B' = \delta(B)$.
Step 4: Compute $C' = A'B' \in L'[x;\sigma']/\langle x^r - 1\rangle$ by Algorithm 1.
Step 5: Return $\delta^{-1}(C')$.

Before analyzing the whole procedure, we discuss the first step, computing all conjugates of λ. Reference [6] assumes that the application of σ in the working basis $\mathscr{W}$ of L takes quasi-linear time, that is, $O^\sim(r)$ operations in K; from this, we would deduce that applying σ' to an element of L' takes $O^\sim(rs)$ operations in K'. However, as noted in the introduction, we would rather not make such a strong assumption. If L is given as $L = K[z]/f(z)$, and thus $L' = K[z,\xi]/\langle f(z), g(\xi)\rangle$, given $\sigma(z \bmod f)$, von zur Gathen and Shoup's iterated Frobenius algorithm [12] allows us to compute all conjugates of λ in $O^\sim(r^2)$ operations in K', that is, $O^\sim(r^2 s)$ operations in K; this is optimal, up to logarithmic factors. We now show that this is still possible, working under the assumptions of this paper.

Proposition 2.3. *Under assumption* **H**, *given λ in L', one can compute the sequence $\lambda, \sigma'(\lambda), \ldots, \sigma'^{r-1}(\lambda)$ using $O^\sim(r^2 s)$ operations in K.*

Proof. Suppose that λ has coefficients $(\beta_0, \ldots, \beta_{r-1})$ on the working basis $\mathscr{W}'$ of L'. Under assumption **(H)**, we can compute its coefficients $(\gamma_0, \ldots, \gamma_{r-1})$ on the normal basis $\mathscr{N}'$ in $O(r^2)$ operations in K', that is, $O^\sim(r^2 s)$ operations in K, by a matrix-vector product with $\boldsymbol{M}_{\mathscr{W}\to\mathscr{N}}$.

Let $\boldsymbol{L} \in K'^{r\times r}$ be the matrix whose ith column contains the coefficients of $\sigma'^i(\lambda)$ on the working basis $\mathscr{W}'$, and let $\boldsymbol{M}_{\mathscr{N}\to\mathscr{W}}$ be the change-of-basis matrix from $\mathscr{N}'$ to $\mathscr{W}'$. Then, we have the equality

$$\boldsymbol{L} = \boldsymbol{M}_{\mathscr{N}\to\mathscr{W}} \begin{bmatrix} \gamma_0 & \gamma_{r-1} & \cdots & \gamma_1 \\ \gamma_1 & \gamma_0 & \cdots & \gamma_2 \\ \vdots & \vdots & \ddots & \vdots \\ \gamma_{r-1} & \gamma_{r-2} & \cdots & \gamma_0 \end{bmatrix}.$$

Since the right-hand is a Hankel matrix, we can left-multiply it by a vector in $O^\sim(r)$ operations in K'. Hence the total cost to compute $\boldsymbol{L}$ is $O^\sim(r^2)$ operations in K', that is, $O^\sim(r^2 s)$ operations in K. □

Corollary 2.4. *Under assumption* **H**, *Algorithm 2 computes the product AB using $O^\sim(R^{\omega-2} r^2 s)$ operations in K and $O^\sim(r)$ bit operations.*

Proof. The previous proposition gives the cost of computing $s_0, \ldots, s_{r-1}$; the products $\lambda_1, \ldots, \lambda_r$ can be deduced for another $O^\sim(r^2)$ operations in K', which is $O^\sim(r^2 s)$ operations in K; this

gives us A' and B'. To compute their product C', Proposition 2.1 takes $O^\sim(R^{\omega-2}r^2s)$ operations in K and $O^\sim(r)$ bit operations. Finally, to recover $\delta^{-1}(C')$, we have to invert all λ_is (they are units, by assumption); this takes $O^\sim(r^2s)$ operations in K again. □

2.4 Main algorithm

The description of the main algorithm is essentially taken from [6], but we replace the procedure for multiplication modulo $x^r - a$ given in that reference by ours. A more minor difference is that we simplify the algorithm by not fully exploiting some properties given in [6], that would allow us to save a factor $O^\sim(s)$; since s will be logarithmic in the input size, this is harmless. Finally, we show how fast multipoint evaluation is actually required to obtain the claimed runtime.

To compute the product AB in $L[x;\sigma]$, we compute its image modulo central moduli of the form $x^r - a_i$, for $a_0, a_1, \ldots$ as in the previous subsection. If K is a small finite field, we may have to extend it in order to guarantee the existence of sufficiently many such moduli. Suppose that A and B have degree at most d, so that $C = AB$ has degree at most $2d$, and let $e = \lceil 2d/r \rceil + 1$; this will be the number of moduli we need.

Lemma 2.5. *Let K' be an extension of K, let Γ be a subset of K' of cardinality at least $e(e+1)r$, and let $L' = L \otimes_K K'$. Fix a basis of L' over K'. Then for $\lambda_1, \ldots, \lambda_e$ in L', with coefficients taken uniformly at random in Γ, the probability that their norms $a_1, \ldots, a_e$ be non-zero and pairwise distinct is at least $1/2$.*

Proof. For λ in L', its norm $a = \lambda\sigma'(\lambda)\cdots\sigma'^{r-1}(\lambda)$ is the determinant of the multiplication endomorphism by λ (seen as a K'-linear map $L' \to L'$). Hence, it is a non-constant homogeneous polynomial of degree r in the coefficients of λ (on an arbitrary K'-basis of L'); we write it $\Delta(\lambda)$. Then, the conclusion we want is the non-vanishing of the product of all $\Delta(\lambda_i)$ and $\Delta(\lambda_i) - \Delta(\lambda_j)$, for $1 \le i < j \le e$. This is an expression of degree $e(e+1)r/2$ in the coefficients of the λ_i's, so the conclusion follows from the DeMillo-Lipton-Schwartz-Zippel lemma. □

Algorithm 3: Multiplication.

Input: Two polynomials $A, B \in L[x;\sigma]$ of degree at most d.
Output: AB with probability at least $1/2$, or `error`

- **Step 1:** Let $e = \lceil 2d/r \rceil + 1$.
- **Step 2:** Build an extension K' of K, such that $|K'| \ge e(e+1)r$ and let $s = [K' : K]$.
- **Step 3:** Pick a subset Γ of K' of cardinality at least $e(e+1)r$.
- **Step 4:** Pick $\lambda_1, \cdots, \lambda_e$ in $L' = L \otimes_K K'$. by choosing their coefficients uniformly at random in Γ.
- **Step 5:** Compute the norms $a_1, \ldots, a_e$ of $\lambda_1, \cdots, \lambda_e$. If any of them vanishes, raise an error.
- **Step 6:** Compute all $A_i = A \bmod (x^r - a_i)$ and $B_i = B \bmod (x^r - a_i)$.
- **Step 7:** Compute all $C_i = A_iB_i \bmod (x^r - a_i)$.
- **Step 8:** Recover $C = AB$ from $C_1, \ldots, C_e$.

Proposition 2.6. *Under assumption* **H**, *Algorithm 3 computes the product AB using an expected $O^\sim(\max(d,r)rR^{\omega-2})$ operations in K and $O^\sim(\max(d,r))$ bit operations, with probability of success at least $1/2$; otherwise, it raises an error.*

Proof. For K finite, s is $O(\log(dr))$, and K' can be built in an expected $O(\log(dr)^2)$ operations in K [29]; if K is infinite, we take $K' = K$ and $s = 1$. Given the λ_i's, the cost of computing all a_i's will be subsumed in that of the further steps. If the conclusions of Lemma 2.5 hold, then all λ_i's are invertible (so we can apply the algorithm of the previous section), and their norms a_i's are pairwise distinct.

Write $A = \sum_{j<r} \alpha_j(x^r)x^j$, $B = \sum_{j<r} \beta_j(x^r)x^j$ and $C = AB = \sum_{j<r} \gamma_j(x^r)x^j$, where all $\alpha_j, \beta_j, \gamma_j$ have degree at most $\lceil 2d/r \rceil = e-1$. Then, for $i \le e$, $A \bmod (x^r - a_i) = \sum_{j<r} \alpha_j(a_i)x^j$. Thus, Step 6 amounts to evaluating $\alpha_0, \ldots, \alpha_{r-1}$ at $a_1, \ldots, a_e$ (and similarly for B). This takes $O^\sim(\max(d,r))$ operations in K' by fast evaluation, which is also $O^\sim(\max(d,r))$ operations in K. Step 7 involves $O(e)$ calls to Algorithm 2; this costs $O^\sim(R^{\omega-2}er^2s)$ operations in K and $O^\sim(er)$ bit operations. The former number is $O^\sim(\max(d,r)rR^{\omega-2})$, and the latter $O^\sim(\max(d,r))$. Since $\mathbb{S}_r(A,B)$ is the sumset for all reductions C_i, the computation of $\mathbb{S}_r(A,B)$ needs to be done only once, reducing the overall computing time. Finally, given $C_i = C \bmod (x^r - a_i)$, as the $x^r - a_i$ are central elements in $L'[x;\sigma]$, the reductions C_i have the same form as in the commutative ring $L'[x]$, and we can regard C, C_i all of them as in the ring $L'[x]$. For e pairwise distinct a_i, we can recover C by r interpolations in degree $e-1$ in K' for another $O^\sim(\max(d,r))$ operations in K. If the a_i's are not pairwise distinct, the interpolation algorithm raises an error. □

Our main algorithm now repeats the procedure above until it succeeds; this will happen after an expected $O(1)$ attempts, thereby establishing Theorem 1.1.

3 A supersparse algorithm

Let again A and B be in $L[x;\sigma]$, both of degree at most d. We now give a multiplication algorithm whose complexity is polynomial in r, $\log(d)$ and S, where S is the size of the sumset $\mathbb{S}(A,B) = \mathrm{supp}(A) + \mathrm{supp}(B)$ (recall that the support of AB is contained in $\mathbb{S}(A,B)$). The first part of the algorithm is Monte Carlo, and costs $O^\sim(\log(d)S\log(1/\mu))$ bit operations, for a probability of failure at most μ; the rest of the algorithm is Las Vegas.

3.0.1. Outlook of the algorithm. The first step in our algorithm computes $\mathbb{S}(A,B)$ as defined above. For any given error tolerance μ, the algorithm in [2] achieves this with bit complexity $O^\sim(\log(d)S\log(1/\mu))$ and with probability at least $1-\mu$. Here, and in what follows, we write $S = \#\mathbb{S}(A,B)$.

Let us write $\mathbb{S}(A,B) = \{e_1, \ldots, e_S\}$ and $AB = c_1x^{e_1} + \cdots + c_Sx^{e_S} \in L[x;\sigma]$, with all c_i in L. For a non-zero multiple q of r, $x^q - 1$ is central, and we have

$$(AB) \bmod (x^q - 1) = c_1x^{e_1 \bmod q} + \cdots + c_Sx^{e_S \bmod q},$$

with $e_i \bmod q$ in $\{0, 1, \ldots, q-1\}$ for all i. If all $e_i \bmod q$ are pairwise distinct, and if we assume that $\mathbb{S}(A,B)$ is known, computing $(AB) \bmod (x^q - 1)$ allows us to recover AB.

However, even through randomization, we are not able to find a q satisfying such a condition and of growth rate less than quadratic in S. Instead, we use an approach coming from [1]: we allow for a certain number of $e_i \bmod q$ to coincide. We will then take q of the form $q = pr$, with p a prime whose size is well controlled. For p satisfying certain luckiness conditions, we will be able to recover at

least half the terms in AB; then, we compute the remaining terms recursively.

3.0.2. Finding a prime. Let n be a non-zero integer, and let $\mathbb{T}$ be a subset of $\{0,\dots,2d\}$. An element e in $\mathbb{T}$ is called a *collision* modulo n if there exists $e' \neq e$ in $\mathbb{T}$ such that $e \equiv e' \bmod n$.

Lemma 3.1. *One can find using an expected $O^\sim(\log(d)T)$ bit operations a prime p such that $p \in O(\log(d)T)$ and $\mathbb{T}$ has at most $T/2$ collisions modulo p, with $T = \#\mathbb{T}$.*

Proof. Let $\lambda = \max(21, \lceil 20(T-1)\ln(2d)/3\rceil)$. Then, Lemma 8 in [1] shows that if p is a random prime in $\{\lambda,\dots,2\lambda\}$, with probability at least $1/2$, $\mathbb{T}$ has less than $T/2$ collisions modulo p. In particular, trying an expected $O(1)$ such primes is sufficient to find a suitable one. By sieving, we can compute all primes up to 2λ in $O^\sim(\log(d)T)$ bit operations. Given a prime p in $\{\lambda,\dots,2\lambda\}$, we can compute $\mathbb{T} \bmod p$ in the same asymptotic cost. Counting collisions can be done by (for instance) sorting all $e_i \bmod p$, in $O^\sim(T\log\log(d))$ bit operations. □

3.0.3. Finding half the terms. Our main procedure is the following. In addition to A and B, we take as input an "approximation" P of the product AB; as output, we return a better approximation of AB, as specified below.

Algorithm 4: Half multiplication.
Input:

- A, B in $L[x;\sigma]$ of degrees at most d
- $P \in L[x;\sigma]$, such that all terms of P are terms of AB
- a set $\mathbb{T} \subset \{0,\dots,2d\}$ containing the support of $AB - P$

Output:

- P^* in $L[x;\sigma]$ such that all terms of P^* are terms of AB.
- a set $\mathbb{T}^* \subset \{0,\dots,2d\}$ containing the support of $AB - P^*$, such that $\#\mathbb{T}^* \leq \#\mathbb{T}/2$.

Step 1: find a prime $p \in O(\log(d)T)$ such that $\mathbb{T}$ has at most $T/2$ collisions modulo p, with $T = \#\mathbb{T}$.
Step 2: compute $u = (AB - P) \bmod (x^{pr} - 1)$.
Step 3: compute $f_1 = e_1 \bmod pr, \dots, f_T = e_T \bmod pr$.
Step 4: let $\mathbb{T}^* \subset \mathbb{T}$ be the set of collisions in $\mathbb{T}$ modulo pr.
Step 5: Let $P^* = P$. For $i = 1,\dots,T$, if e_i is not in $\mathbb{T}^*$, find the coefficient c_i of x^{f_i} in u and let $P^* = P^* + c_i x^{e_i}$.
Step 6: Return P^* and $\mathbb{T}^*$.

Proposition 3.2. *Algorithm 4 is correct. Under assumption* **H**, *it uses an expected $O^\sim(\log(d)Tr^\omega)$ operations in K and $O^\sim(\log(d)Tr)$ bit operations, with $T = \#\mathbb{T}$.*

Proof. By construction, all terms in P^* are either terms in P (in which case they are terms in AB), or terms in $AB - P$ (and thus in AB as well); they are thus always terms in AB, which shows that the first item holds.

Next, take a term in AB but not in P^*; then, it belongs to $\mathbb{T}$, but not to $\mathbb{T} - \mathbb{T}^*$; this proves that the support of $AB - P^*$ is in $\mathbb{T}^*$, as claimed. Finally, since $\mathbb{T}$ has at most $T/2$ collisions modulo p, it has at most $T/2$ collisions modulo pr; hence, we have $\#\mathbb{T}^* \leq \#\mathbb{T}/2$. Correctness is proved.

Next, we analyze the cost of this procedure. By Lemma 3.1, Step 1 takes an expected $O^\sim(\log(d)T)$ bit operations. At Step 2, we compute u by reducing A and B modulo $x^{pr}-1$, multiplying the remainders and reducing the product, and subtracting P mod $(x^{pr}-1)$.

Since p is $O^\sim(\log(d)T)$, using Theorem 1.1, the cost of computing the product modulo $x^{pr}-1$ is an expected $O^\sim(\log(d)Tr^\omega)$ operations in K and $O(\log(d)Tr)$ bit operations. This dominates the cost of the other steps. □

3.0.4. Main algorithm. The main procedure calls Algorithm 4 on rapidly decreasing supports $\mathbb{T}$; it finishes after $O(\log S)$ iterations, where S is the cardinality of $\mathbb{S}(A,B) = \mathrm{supp}(A) + \mathrm{supp}(B)$.

Algorithm 5: Multiplication.
Input:

- A, B in $L[x;\sigma]$ of degrees at most d
- error tolerance μ.

Output: with probability at least $1-\mu$, the product AB.

Step 1: compute $\mathbb{S}(A,B) = \mathrm{supp}(A) + \mathrm{supp}(B)$.
Step 2: let $P = 0$ and $\mathbb{T} = \mathbb{S}(A,B)$.
Step 3: while $\mathbb{T}$ is not empty do
  a: let $P, \mathbb{T} =$ **Half multiplication**$(A, B, P, \mathbb{T})$.
Step 4: return P.

The following proposition results directly from Proposition 3.2, using the algorithm of Arnold and Roche [2] for computing $\mathbb{S}(A,B)$ with a cost of $O^\sim(S\log(d)\log\frac{1}{\mu})$ bit operations. It establishes Theorem 1.2.

Proposition 3.3. *Algorithm 5 succeeds with probability at least $1-\mu$. Under assumption* **H**, *it uses an expected $O^\sim(\log(d)Sr^\omega)$ operations in K and an expected $O^\sim(\log(d)S(r+\log(1/\mu)))$ bit operations, with $S = \#\mathbb{S}(A,B)$.*

4 Multivariate skew polynomials

Finally, we extend our second univariate multiplication algorithm to certain multivariate cases, using Kronecker substitution. One may also use the algorithm of Section 2, but the result would be exponential in the number n of variables: the runtime of the algorithm of Section 2 is polynomial in the input degree, and Kronecker substitution produces univariate polynomials of degree exponential in n.

Multivariate skew polynomials have not been as intensively studied; refer to [13, 20, 21] for recent work. Let $L[x_1,\dots,x_n;\sigma_1,\dots,\sigma_n]$ be a multivariate skew polynomial ring, where L is a field, with the relations $x_i a = \sigma_i(a)x_i$ and $x_i x_j = x_j x_i$ for all i,j, and where each σ_i is an automorphism of L.

In [21], using a matrix of endomorphisms, the authors define more general multivariate skew polynomials. Our definition seems to correspond to a diagonal matrix containing automorphisms, which is only a special case of the definition in [21]. However, in [21], x_i, x_j do not commute for $i \neq j$, which is used to make sure the uniqueness of evaluation, defined as the remainder of a right division. In our definition, we assume $x_i x_j = x_j x_i$ for all i,j, and the evaluation at a point is defined as the value of function which replaces x_i in the skew polynomial with σ_i.

We assume that there exists an automorphism σ of L, having order r, and integers $e_1,\dots,e_n$ such that for all i, $\sigma_i = \sigma^{e_i}$, and

as before we let K be the fixed field of σ. This assumption is for instance valid when L is a finite field.

Consider integers $\boldsymbol{N} = (N_1, \ldots, N_n)$ and the L-linear mapping $\Psi_{\boldsymbol{N}}$ defined by

$$\begin{array}{rccc} \Psi_{\boldsymbol{N}} : & L[x_1, \ldots, x_n; \sigma_1, \ldots, \sigma_n] & \to & L[x; \sigma] \\ & x_1^{d_1} \cdots x_n^{d_n} & \mapsto & x^{d_1 N_1 + \cdots + d_n N_n} \end{array}$$

This is simply a Kronecker substitution, in a non-commutative setting.

Lemma 4.1. *If $N_i \equiv e_i \bmod r$ for all i, then $\Psi_{\boldsymbol{N}}$ is a K-algebra morphism.*

Proof. Since $\Psi_{\boldsymbol{N}}$ acts multiplicatively on monomials, the only property we have to verify is that for integers $(d_1, \ldots, d_n)$ and b in L, $\Psi_{\boldsymbol{N}}(x_1^{d_1} \cdots x_n^{d_n})\Psi_{\boldsymbol{N}}(b) = \Psi_{\boldsymbol{N}}(x_1^{d_1} \cdots x_n^{d_n} b)$. The former equals $\sigma^{\sum_{i=1}^n d_i N_i}(b) x^{\sum_{i=1}^n d_i N_i}$, while the latter is $\sigma^{\sum_{i=1}^n d_i e_i}(b) x^{\sum_{i=1}^n d_i N_i}$. Our assumption implies that the exponents $\sum_{i=1}^n d_i N_i$ and $\sum_{i=1}^n d_i e_i$ are the same modulo r, and the conclusion follows. □

For $D \geq 0$, let $L[x_1, \ldots, x_n; \sigma_1, \ldots, \sigma_n]_D$ be the L-vector space of skew polynomials of total degree less than D. We now discuss conditions on $\boldsymbol{N}$ that ensures that the restriction of $\Psi_{\boldsymbol{N}}$ to $L[x_1, \ldots, x_n; \sigma_1, \ldots, \sigma_n]_D$ is injective.

Lemma 4.2. *Let D be a positive integer. Assume $N_i \in \mathbb{N}_{>0}$ satisfy $D \leq N_1$ and $N_i D \leq N_{i+1}$ for $1 \leq i < n$. Then the restriction of $\Psi_{\boldsymbol{N}}$ to $L[x_1, \ldots, x_n; \sigma_1, \ldots, \sigma_n]_D$ is injective.*

Proof. Suppose $m \in \mathbb{N}_{>0}$ can be represented as $m = \sum_{i=1}^n d_i N_i$ with $\sum_{i=1}^n d_i < D$, and in particular $0 \leq d_i < D$. It suffices to show that this relation defines d_n uniquely; once this is known, we set $m' = m - d_n N_n$ and the claim follows by induction. Precisely, we prove that $d_n = \lfloor \frac{m}{N_n} \rfloor$. Since

$$\begin{aligned} d_n N_n \leq m = \sum_{i=1}^{n} d_i N_i &\leq (D-1)(1 + \sum_{i=1}^{n-1} N_i) + d_n N_n \\ &= D(1 + \sum_{i=1}^{n-1} N_i) - (1 + \sum_{i=1}^{n-1} N_i) + d_n N_n \\ &\leq (D + \sum_{i=2}^{n} N_i) - (1 + \sum_{i=1}^{n-1} N_i) + d_n N_n \\ &= D - 1 - N_1 + N_n + d_n N_n \\ &< N_n + d_n N_n = (d_n + 1) N_n. \end{aligned}$$

Dividing by N_n on both sides, we get $d_n \leq \frac{m}{N_n} < (d_n + 1)$. Since d_n is an integer, we get $d_n = \lfloor \frac{m}{N_n} \rfloor$, and we are done. □

The following algorithm describes how to compute the d_i's.

Algorithm 6: Index.

Input:

- Positive integers $N_1, \ldots, N_n$, where $D \leq N_1$ and $N_i D \leq N_{i+1}$ for $i = 1, \ldots, n-1$.
- A positive integer $m = d_1 N_1 + \cdots + d_n N_n$, where $0 \leq d_i < D$ for $i = 1, \ldots, n$.

Output: The indices $d_1, \ldots, d_n$.

Step 1: For $i = n, \ldots, 1$ do
  a: Let $d_i = \lfloor \frac{m}{N_i} \rfloor$.
  b: Let $m = m - d_i N_i$.
Step 2: Return $d_1, \ldots, d_n$.

Lemma 4.3. *Algorithm 6 is correct and requires $O^{\sim}(n \log(D) + n \log(N_n))$ bit operations.*

Proof. Correctness comes from the expression $d_n = \lfloor \frac{m}{N_n} \rfloor$, which was established in the proof of Lemma 4.2. As to complexity, each iteration of Step 1 costs a constant number of arithmetic operations. Since $m \leq D N_n$ and $N_1 < N_2 < \cdots < N_n$, the height of m is $O(\log(D) + \log(N_n))$, and the total cost of Step 1 is $O^{\sim}(n \log(D) + n \log(N_n))$ bit operations. □

Taking into account the constraints in the two previous lemmas, we obtain the following construction of integers $N_1, \ldots, N_n$.

Lemma 4.4. *Given a positive integer D, set $N_0 = 1$ and define $\boldsymbol{N} = (N_1, \ldots, N_n)$ recursively by*

$$N_{i+1} = e_{i+1} + k_{i+1} r, \text{ where } k_{i+1} = \max\{\lceil \frac{D N_i - e_{i+1}}{r} \rceil, 0\}.$$

Then $\boldsymbol{N}$ satisfies the conditions of Lemmas 4.1 and 4.2, and $N_i \leq r D^{n+1}$ holds for all i.

Proof. The congruence conditions clearly hold. For $i \geq 0$, we claim that $N_i D \leq N_{i+1} \leq N_i D + r$; the left-hand side then proves the inequalities needed in Lemma 4.2.

If $k_{i+1} = 0$, then $N_i D \leq e_{i+1}$, so $N_{i+1} = e_{i+1}$, which means $N_i D \leq N_{i+1}$. On the other hand, since $0 \leq e_{i+1} < r$, we have $N_{i+1} \leq N_i D + r$. If $k_{i+1} > 0$, then $k_{i+1} = \lceil \frac{D N_i - e_{i+1}}{r} \rceil$, so we have $\frac{D N_i - e_{i+1}}{r} \leq k_{i+1} < \frac{D N_i - e_{i+1}}{r} + 1$. This gives $D N_i - e_{i+1} \leq k_{i+1} r < D N_i - e_{i+1} + r$, and thus $D N_i \leq N_{i+1} < D N_i + r$. In either case, we proved the claim. This inequalities also imply (by induction) that all N_i's satisfy $N_i \leq D^i + r(D^i - 1)/(D-1)$, and thus $N_i \leq r D^{n+1}$. □

Corollary 4.5. *Let D and $\boldsymbol{N}$ as in the previous lemma, and let C be in $L[x_1, \ldots, x_n; \sigma_1, \ldots, \sigma_n]_D$, with $\#C \leq S$. Given $\Psi_{\boldsymbol{N}}(C)$ we can recover C in $O^{\sim}(S n^2 \log(D) + S n \log(r))$ bit operations.*

Proof. Apply Algorithm 6 to all terms of $\Psi_{\boldsymbol{N}}(C)$. Each instance takes $O^{\sim}(n \log(D) + n \log(N_n))$ bit operations, and the previous lemma proved that $\log(N_n)$ is $O(n \log(D) + \log(r))$. □

We can now present our sparse multivariate multiplication algorithm.

Algorithm 7: Multivariate Multiplication.

Input:

- A, B in $L[x_1, \ldots, x_n; \sigma_1, \ldots, \sigma_n]$
- error tolerance μ

Output: with probability at least $1 - \mu$, the product AB

Step 1: let $D = \deg A + \deg B + 1$.
Step 2: let $\boldsymbol{N}$ be as in Lemma 4.4.
Step 3: compute $\widetilde{A} = \Psi_{\boldsymbol{N}}(A)$ and $\widetilde{B} = \Psi_{\boldsymbol{N}}(B)$
Step 4: compute $\widetilde{C} = \widetilde{A}\widetilde{B}$ by calling Algorithm 5 with inputs $\widetilde{A}$, $\widetilde{B}$ and μ
Step 5: return $\Psi_{\boldsymbol{N}}^{-1}(\widetilde{C})$

Proposition 4.6. *Algorithm 7 computes AB with probability at least $1-\mu$ and costs $O^{\sim}(nr^{\omega}S\log D)$ field operations in K plus $O^{\sim}(n^2S\log D + nSr\log D + nS\log D\log(1/\mu) + S\log r\log(1/\mu))$ bit operations, where $S = \#\mathbb{S}(A,B)$.*

Proof. Correctness comes from Lemma 4.4: if the product $\widetilde{A}\widetilde{B}$ computed in Step 4 is correct, then the output is the product AB. By Proposition 3.3, Algorithm 5 returns the correct product with probability at least $1-\mu$, so we are done.

Step 2 needs n operations. Since the bit-lengths are $O(n\log D + \log r)$, the bit cost is $O^{\sim}(n^2\log D + n\log r)$. At Step 3, since $\#A, \#B \leq S$, we use at most $O^{\sim}(n^2S\log D + nS\log r)$ bit operations. At Step 4, the degree of $\widetilde{f}\cdot\widetilde{g}$ is $\widetilde{d}$, so by Proposition 3.3 we use $O^{\sim}(r^{\omega}S\log(\widetilde{d}))$ operations in K and $O^{\sim}(\log(\widetilde{d})S(r+\log(1/\mu)))$ bit operations. Since $\widetilde{d} \leq DN_n$ and N_n is in $O(rD^n)$, this is $O^{\sim}(nr^{\omega}S\log D)$ operations in K and $O^{\sim}(nSr\log D + S\log r\log(1/\mu) + nS\log D\log(1/\mu))$ bit operations. In Step 5, by Lemma 4.3, we use $O^{\sim}(nS\log D + nS\log N_n) = O^{\sim}(n^2S\log D + nS\log r)$ bit operations. □

5 Conclusions

In this paper, we present new multiplication algorithms for skew polynomials. Our first new algorithm is a Las Vegas algorithm for multiplication in $L[x;\sigma]$; the second algorithm is for multiplication of "supersparse" polynomials in $L[x;\sigma]$. Its cost is sensitive to the number of non-zero terms, and is significantly faster than previous algorithms when the product has large degree but few terms.

Finally, we consider multiplying sparse multivariate skew polynomials in $L[x_1,\ldots,x_n;\sigma_1,\ldots,\sigma_n]$. We introduced a non-commutative Kronecker substitution scheme, and present an algorithm with polynomial runtime in the input and output size. This is a particular improvement over standard dense algorithms, which could be of exponential complexity in the number of non-zero input terms.

Acknowledgement

The authors would like to acknowledge the careful anonymous review of this paper.

References

[1] A. Arnold, M. Giesbrecht, and D. Roche. 2013. Faster sparse interpolation of straight-line programs. In *International Workshop on Computer Algebra in Scientific Computing*. Springer, 61–74.
[2] A. Arnold and D. Roche. 2015. Output-sensitive algorithms for sumset and sparse polynomial multiplication. In *ISSAC'15*. ACM Press, 29–36.
[3] D. Boucher, P. Gaborit, W. Geiselmann, O. Ruatta, and F. Ulmer. 2010. Key exchange and encryption schemes based on non-commutative skew polynomials. In *International Workshop on Post-Quantum Cryptography*. Springer, 126–141.
[4] D. Boucher, W. Geiselmann, and F. Ulmer. 2007. Skew-cyclic codes. *Applicable Algebra in Engineering, Communication and Computing* 18, 4 (2007), 379–389.
[5] D. Boucher and F. Ulmer. 2009. Coding with skew polynomial rings. *Journal of Symbolic Computation* 44, 12 (2009), 1644–1656.
[6] X. Caruso and J. Le Borgne. 2017. Fast multiplication for skew polynomials. In *ISSAC'17*. ACM, 77–84.
[7] D. Coppersmith and S. Winograd. 1990. Matrix multiplication via arithmetic progressions. *J. Symb. Comput.* 9, 3 (1990), 251–280.
[8] J.-M. Couveignes and R. Lercier. 2009. Elliptic periods for finite fields. *Finite Fields Their Appl.* 15, 1 (2009), 1–22.
[9] E. Gabidulin. 1985. Theory of codes with maximum rank distance. *Problemy Peredachi Informatsii* 21, 1 (1985), 3–16.
[10] F. Le Gall. 2014. Powers of tensors and fast matrix multiplication. In *ISSAC'14*. ACM Press, 296–303.
[11] J. von zur Gathen and M. Giesbrecht. 1990. Constructing normal bases in finite fields. *J. Symb. Comput* 10 (1990), 547–570.
[12] J. von zur Gathen and V. Shoup. 1992. Computing Frobenius maps and factoring polynomials. *Computational Complexity* 2, 3 (1992), 187–224.
[13] W. Geiselmann and F. Ulmer. 2019. Skew Reed-Muller codes. *Contemporary mathematics* (2019), 107–116.
[14] M. Giesbrecht. 1998. Factoring in skew-polynomial rings over finite fields. *Journal of Symbolic Computation* 26, 4 (1998), 463–486.
[15] M. Giesbrecht, A. Jamshidpey, and É Schost. 2019. Quadratic-Time Algorithms for Normal Elements. In *ISSAC'19*. ACM Press, 179–186.
[16] K. Girstmair. 1999. An algorithm for the construction of a normal basis. *Journal of Number Theory* 78, 1 (1999), 36–45.
[17] D. Goss. 1996. *Basic Structures of Function Field Arithmetic*. Springer Berlin Heidelberg.
[18] S. Johnson. 1974. Sparse polynomial arithmetic. *ACM SIGSAM Bulletin* 8, 3 (1974), 63–71.
[19] E. Kaltofen and V. Shoup. 1998. Subquadratic-time factoring of polynomials over finite fields. *Math. Comp.* 67, 223 (1998), 1179–1197.
[20] U. Martínez-Penas. 2019. Classification of multivariate skew polynomial rings over finite fields via affine transformations of variables. *arXiv: 1908.06833* (2019).
[21] U. Martínez-Penas and F. R. Kschischang. 2019. Evaluation and interpolation over multivariate skew polynomial rings. *Journal of Algebra* 525 (2019), 111–139.
[22] M. Monagan and R. Pearce. 2009. Parallel Sparse Polynomial Multiplication Using Heaps. In *ISSAC'09*. 263–269.
[23] M. Monagan and R. Pearce. 2011. Sparse Polynomial Pseudo Division Using a Heap. *J. Symb. Comp.* 46, 7 (2011), 807–822.
[24] O. Ore. 1933. Theory of non-commutative polynomials. *Annals of Mathematics* (1933), 480–508.
[25] S. Puchinger and A. Wachter-Zeh. 2016. Sub-quadratic decoding of Gabidulin codes. In *2016 IEEE International Symposium on Information Theory (ISIT)*. IEEE, 2554–2558.
[26] S. Puchinger and A. Wachter-Zeh. 2018. Fast operations on linearized polynomials and their applications in coding theory. *Journal of Symbolic Computation* 89 (2018), 194–215.
[27] F. Kschischang R. Koetter. 2008. Coding for errors and erasures in random network coding. *IEEE Transactions on Information Theory* 54, 8 (2008), 3579–3591.
[28] D. Roche. 2018. What can (and can't) we do with sparse polynomials?. In *ISSAC'18*. 25–30.
[29] V. Shoup. 1994. Fast construction of irreducible polynomials over finite fields. *Journal of Symbolic Computation* 17, 5 (1994), 371–391.
[30] J. van der Hoeven and G. Lecerf. 2012. On the Complexity of Multivariate Blockwise Polynomial Multiplication. In *ISSAC'12*. 211–218.
[31] J. van der Hoeven and G. Lecerf. 2013. On the bit-complexity of sparse polynomial and series multiplication. *J. Symbolic Computation* 50 (2013), 227–254.
[32] Y. Zhang. 2010. A secret sharing scheme via skew polynomials. In *2010 International Conference on Computational Science and Its Applications*. IEEE, 33–38.

Essentially Optimal Sparse Polynomial Multiplication

Pascal Giorgi
LIRMM, Univ. Montpellier, CNRS
Montpellier, France
pascal.giorgi@lirmm.fr

Bruno Grenet
LIRMM, Univ. Montpellier, CNRS
Montpellier, France
bruno.grenet@lirmm.fr

Armelle Perret du Cray
LIRMM, Univ. Montpellier, CNRS
Montpellier, France
armelle.perret-du-cray@lirmm.fr

ABSTRACT

We present a probabilistic algorithm to compute the product of two univariate sparse polynomials over a field with a number of bit operations that is quasi-linear in the size of the input and the output. Our algorithm works for any field of characteristic zero or larger than the degree. We mainly rely on sparse interpolation and on a new algorithm for verifying a sparse product that has also a quasi-linear time complexity. Using Kronecker substitution techniques we extend our result to the multivariate case.

CCS CONCEPTS

• **Computing methodologies** → **Algebraic algorithms**; • **Theory of computation** → **Design and analysis of algorithms**; • **Mathematics of computing** → **Probabilistic algorithms**.

KEYWORDS

arithmetic, sparse polynomial multiplication, sparse interpolation, probabilistic verification

ACM Reference Format:
Pascal Giorgi, Bruno Grenet, and Armelle Perret du Cray. 2020. Essentially Optimal Sparse Polynomial Multiplication. In *International Symposium on Symbolic and Algebraic Computation (ISSAC '20), July 20–23, 2020, Kalamata, Greece.* ACM, New York, NY, USA, 8 pages. https://doi.org/10.1145/3373207.3404026

1 INTRODUCTION

Polynomials are one of the most basic objects in computer algebra and the study of fast polynomial operations remains a very challenging task. Polynomials can be represented using either the dense representation, that stores all the coefficients in a vector, or the more compact sparse representation, that only stores nonzero monomials. In the dense representation, we know quasi-optimal algorithms for decades. Yet, this is not the case for sparse polynomials.

In the sparse representation, a polynomial $F = \sum_{i=0}^{D} f_i X^i \in R[X]$ is expressed as a list of pairs (e_i, f_{e_i}) such that all the f_{e_i} are nonzero. We denote by $\#F$ its *sparsity*, i.e. the number of nonzero coefficients. Let F be a polynomial of degree D, and B a bound on the size of its coefficients. Then, the size of the sparse representation of F is $O(\#F(B + \log D))$ bits. It is common to use $B = 1 + \max_i(\lfloor \log_2(|f_{e_i}|)\rfloor)$ if $R = \mathbb{Z}$ and $B = 1 + \lfloor \log_2 q \rfloor$ if $R = \mathbb{F}_q$. The sparse representation naturally extends to polynomials in n variables: Each exponent is replaced by a vector of exponents which gives a total size of $O(\#F(B + n \log D)$.

Several problems on sparse polynomials have been investigated to design fast algorithms, including arithmetic operations, interpolation and factorization. We refer the interested readers to the excellent survey by Roche and the references therein [21]. Contrary to the dense case, note that *fast* algorithms for sparse polynomials have a (poly-)logarithmic dependency on the degree. Unfortunately, as shown by several NP-hardness results, such fast algorithms might not even exist unless P = NP. This is for instance the case for GCD computations [18].

In this paper, we are interested in the problem of sparse polynomial multiplication. In particular, we provide the first quasi-optimal algorithm whose complexity is quasi-linear in both the input and the output sizes.

1.1 Previous work

The main difficulty and the most interesting aspect of sparse polynomial multiplication is the fact that the size of the output does not exclusively depend on the size of the inputs, contrary to the dense case. Indeed, the product of two polynomials F and G has at most $\#F\#G$ nonzero coefficients. But it may have as few as 2 nonzero coefficients.

Example 1. Let $F = X^{14} + 2X^7 + 2$, $G = 3X^{13} + 5X^8 + 3$ and $H = X^{14} - 2X^7 + 2$. Then $FG = 3X^{27} + 5X^{22} + 6X^{20} + 10X^{15} + 3X^{14} + 6X^{13} + 10X^8 + 6X^7 + 6$ has nine terms, while $FH = X^{28} + 4$ has only two.

The product of two polynomials of sparsity T can be computed by generating the T^2 possible monomials, sorting them by increasing degree and merging those with the same degree. Using radix sort, this algorithm takes $O(T^2(\mathsf{M}_R + \log D))$ bit operations, where M_R denotes the cost of one operation in R. A major drawback of this approach is its space complexity that exhibits a T^2 factor, even if the result has less than T^2 terms. Many improvements have been proposed to reduce this space complexity, to extend the approach to multivariate polynomials, and to provide fast implementations in practice [14–16]. Yet, none of these results reduces the T^2 factor in the time complexity.

In general, no complexity improvement is expected as the output polynomial may have as many as T^2 nonzero coefficients. However, this number of nonzero coefficients can be overestimated, giving the opportunity for output-sensitive algorithms. Such algorithms have first been proposed for special cases. Notably, when the output size is known to be small due to sufficiently structured inputs [20], especially in the multivariate case [9, 10], or when the support of the output is known in advance [11]. It is possible to go one step

ISSAC '20, July 20–23, 2020, Kalamata, Greece

ACM ISBN 978-1-4503-7100-1/20/07…$15.00
https://doi.org/10.1145/3373207.3404026

further by studying the conditions for small outputs. A first reason is exponent *collisions*. Let $F = \sum_{i=1}^{T} f_i X^{\alpha_i}$ and $G = \sum_{j=1}^{T} g_j X^{\beta_j}$. A collision occurs when there exist distinct pairs of indices (i_1, j_1) and (i_2, j_2) such that $\alpha_{i_1} + \beta_{j_1} = \alpha_{i_2} + \beta_{j_2}$. Such collisions decrease the number of terms of the result. The second reason is coefficient cancellations. In the previous example, the resulting coefficient is $(f_{i_1} g_{j_1} + f_{i_2} g_{j_2})$, which could vanish depending on the coefficient values. Taking into account the exponent collisions amounts to computing the *sumset* of the exponents of F and G, that is $\{\alpha_i + \beta_j : 1 \le i, j \le T\}$. Arnold and Roche call this set the *structural support* of the product FG and its size the *structural sparsity* [2]. If $H = FG$, then the structural sparsity S of the product FG satisfies $2 \le \#H \le S \le T^2$. Observe that although $\#H$ and S can be close, their difference can reach $O(T^2)$ as shown by the next example.

Example 2. Let $F = \sum_{i=0}^{T-1} X^i$, $G = \sum_{i=0}^{T-1} (X^{Ti+1} - X^{Ti})$ and $H = FG$. We have $\#F = T$, $\#G = 2T$ and the structural sparsity of FG is $T^2 + 1$ while $H = X^{T^2} - 1$ has sparsity 2.

For polynomials with nonnegative integer coefficients, the support of H is exactly the sumset of the exponents of F and G, the structural support of $H = FG$. In this case, Cole and Hariharan describe a multiplication algorithm requiring $\tilde{O}(S \log^2 D)$[1] operations in the RAM model with $O(\log(CD))$ word size [5], where $\log(C)$ bounds the bitsize of the coefficients. Arnold and Roche improve this complexity to $\tilde{O}(S \log D + \#H \log C)$ bit operations for polynomials with both positive and negative integer coefficients [2]. Note that they also extend their result to finite fields and to the multivariate case. A recent algorithm of Nakos avoids the dependency on the structural sparsity for the case of integer polynomials [17], using the same word RAM model as Cole and Hariharan. Unfortunately, the bit complexity of this algorithm ($\tilde{O}((T \log D + \#H \log^2 D) \log(CD) + \log^3 D)$) is not quasi-linear.

In the dense case, quasi-optimal multiplication algorithms rely on the well-known evaluation-interpolation scheme. In the sparse settings, this approach is not efficient. The fastest multiplication algorithms mentioned above [2, 17] mainly rely on a different method called *sparse interpolation*[2], that has received considerable attention. See e.g. the early results of Prony [19] and Ben-Or and Tiwari [4] or the recent results by Huang [12]. Despite extensive analysis of this problem, no quasi-optimal algorithm exists yet. We remark that it is not the only difficulty. Simply using a quasi-optimal sparse interpolation algorithm would not be enough to get a quasi-optimal sparse multiplication algorithm [1].

1.2 Our contributions

Our main result is summarized in Theorem 1.1. We extend the complexity notations to O_ϵ and $\tilde{O}_\epsilon$ for hiding some polynomial factors in $\log(\frac{1}{\epsilon})$. Let $F = \sum_{i=1}^{T} f_i X^{e_i}$. We use $\|F\|_\infty = \max_i |f_i|$ to denote its height, $\#F$ for its number of nonzero terms and $\mathrm{supp}(F) = \{e_1, \ldots, e_T\}$ its support.

Theorem 1.1. *Given two sparse polynomials F and G over $\mathbb{Z}$, Algorithm SparseProduct computes $H = FG$ in $\tilde{O}_\epsilon(T(\log D + \log C))$ bit operations with probability at least $1 - \epsilon$, where $D = \deg(H)$, $C = \max(\|F\|_\infty, \|G\|_\infty, \|H\|_\infty)$ and $T = \max(\#F, \#G, \#H)$. The algorithm extends naturally to finite fields with characteristic larger than D with the same complexity where C denotes the cardinality.*

This result is based on two main ingredients. We adapt Huang's algorithm [12] to interpolate FG in quasi-linear time. Note that the original algorithm does not reach quasi-linear complexity.

Sparse interpolation algorithms, including Huang's, require a bound on the sparsity of the result. We replaced this bound by a guess on the sparsity and an *a posteriori* verification of the product, as in [17]. However, using the classical polynomial evaluation approach for the verification does not yield a quasi-linear bit complexity (see Section 3). Therefore, we introduce a novel verification method that is essentially optimal.

Theorem 1.2. *Given three sparse polynomials F, G and H over $\mathbb{F}_q$ or $\mathbb{Z}$, Algorithm VerifySP tests whether $FG = H$ in $\tilde{O}_\epsilon(T(\log D + B))$ bit operations, where $D = \deg(H)$, B is a bound on the bitsize of the coefficients of F, G and H, and $T = \max(\#F, \#G, \#H)$. The answer is always correct if $FG = H$, and the probability of error is at most ϵ otherwise.*

Finally, using Kronecker substitution, we show that our sparse polynomial multiplication algorithm extends to the multivariate case with a quasi-linear bit complexity $\tilde{O}_\epsilon(T(n \log d + B))$ where n is the number of variables and d the maximal partial degree on each variable. Nevertheless, over finite fields this approach requires an exponentially large characteristic. Using the randomized Kronecker substitution [3] we derive a fast algorithm for finite fields of characteristic polynomial in the input size. Its bit complexity is $\tilde{O}_\epsilon(nT(\log d + B))$. Even though it is not quasi-optimal, it achieves the best known complexity for this case.

2 PRELIMINARIES

We denote by $\mathsf{I}(n) = O(n \log n)$ the bit complexity of the multiplication of two integers of at most n bits [8]. Similarly, we denote by $\mathsf{M}_q(D) = O(D \log(q) \log(D \log q) 4^{\log^* D})$ the bit complexity of the multiplication of two *dense* polynomials of degree at most D over $\mathbb{F}_q$ where q is prime [7]. The cost of multiplying two elements of $\mathbb{F}_{q^s}$ is $O(\mathsf{M}_q(s))$. The cost of multiplying two dense polynomials over $\mathbb{Z}$ of heights at most C and degrees at most D is $\mathsf{M}_{\mathbb{Z}}(D, C) = \mathsf{I}(D(\log C + \log D))$ [24, Chapter 8].

Since our algorithms use reductions *modulo* $X^p - 1$ for some prime number p, we first review useful related results.

Theorem 2.1 (Rosser and Schoenfeld [22]). *If $\lambda \ge 21$, there are at least $\frac{3}{5}\lambda / \ln \lambda$ prime numbers in $[\lambda, 2\lambda]$.*

Proposition 2.2 ([23, Chapter 10]). *There exists an algorithm RandomPrime(λ, ϵ) that returns an integer p in $[\lambda, 2\lambda]$, such that p is prime with probability at least $1 - \epsilon$. Its bit complexity is $\tilde{O}_\epsilon(\log^3 \lambda)$.*

We need two distinct properties on the reductions *modulo* $X^p - 1$. The first one is classical in sparse interpolation to bound the probability of exponent collision in the residue (see [2, Lemma 3.3]).

Proposition 2.3. *Let H be a polynomial of degree at most D and sparsity at most T, $0 < \epsilon < 1$ and $\lambda = \max(21, \frac{10}{3\epsilon} T^2 \ln D)$. Then*

[1] Here, and throughout the article, $\tilde{O}(f(n))$ denotes $O(f(n) \log^k(f(n)))$ for some constant $k > 0$.

[2] Despite their similar names, dense and sparse polynomial interpolation are actually two quite different problems.

with probability at least $1-\epsilon$, RandomPrime$(\lambda, \frac{\epsilon}{2})$ *returns a prime number* p *such that* $H \bmod X^p - 1$ *has the same number of terms as* H*, that is no collision of exponents occurs.*

The second property allows to bound the probability that a polynomial vanishes *modulo* $X^p - 1$.

Proposition 2.4. *Let* H *be a nonzero polynomial of degree at most* D *and sparsity at most* T, $0 < \epsilon < 1$ *and* $\lambda = \max(21, \frac{10}{3\epsilon} T \ln D)$. *Then with probability at least* $1-\epsilon$, RandomPrime$(\lambda, \frac{\epsilon}{2})$ *returns a prime number* p *such that* $H \bmod X^p - 1 \neq 0$.

Proof. For $H \bmod X^p - 1$ to be nonzero, it is sufficient that there exists one exponent e of H that is not congruent to any other exponent e_j modulo p. In other words, it is sufficient that p does not divide any of the $T-1$ differences $\delta_j = e_j - e$. Noting that $\delta_j \leq D$, the number of primes in $[\lambda, 2\lambda]$ that divide at least one δ_j is at most $\frac{(T-1)\ln D}{\ln \lambda}$. Since there exist $\frac{3}{5}\lambda/\ln\lambda$ primes in this interval, the probability that a prime randomly chosen from it divides at least one δ_j is at most $\epsilon/2$. RandomPrime$(\lambda, \epsilon/2)$ returns a prime in $[\lambda, 2\lambda]$ with probability at least $1-\epsilon/2$, whence the result. □

The next two propositions are used to reduce integer coefficients modulo some prime number and to construct an extension field.

Proposition 2.5. *Let* $H \in \mathbb{Z}[X]$ *be a nonzero polynomial,* $0 < \epsilon < 1$ *and* $\lambda \geq \max(21, \frac{10}{3\epsilon} \ln \|H\|_\infty)$. *Then with probability at least* $1-\epsilon$, RandomPrime$(\lambda, \frac{\epsilon}{2})$ *returns a prime* q *such that* $H \bmod q \neq 0$.

Proof. Let h_i be a nonzero coefficient of H. A random prime from $[\lambda, 2\lambda]$ divides h_i with probability at most $\frac{5}{3}\ln\|H\|_\infty/\lambda \leq \epsilon/2$. Since RandomPrime$(\lambda, \epsilon/2)$ returns a prime in $[\lambda, 2\lambda]$ with probability at least $1-\epsilon/2$ the result follows. □

Proposition 2.6 ([23, Chapter 20]). *There exists an algorithm that, given a finite field* $\mathbb{F}_q$*, an integer* s *and* $0 < \epsilon < 1$*, computes a degree-*s *polynomial in* $\mathbb{F}_q[X]$ *that is irreducible with probability at least* $1-\epsilon$*. Its bit complexity is* $\tilde{O}_\epsilon(s^3 \log q)$.

3 SPARSE POLYNOMIAL PRODUCT VERIFICATION

Verifying a product $FG = H$ of dense polynomials over an integral domain R simply falls down to testing $F(\alpha)G(\alpha) = H(\alpha)$ for some random point $\alpha \in R$. This approach exhibits an optimal linear number of operations in R but it is not deterministic. (No optimal deterministic algorithm exists yet.) When $R = \mathbb{Z}$ or $\mathbb{F}_q$, a divide and conquer approach provides a quasi-linear complexity, namely $\tilde{O}(DB)$ bit operations where B bounds the bitsize of the coefficients.

For sparse polynomials with T nonzero coefficients, evaluation is not quasi-linear since the input size is only $O(T(\log D + B))$. Indeed, computing α^D requires $O(\log D)$ operations in R which implies a bit complexity of $\tilde{O}(\log(D)\log(q))$ when $R = \mathbb{F}_q$. Applying this computation to the T nonzero monomials gives a bit complexity of $\tilde{O}(T\log(D)\log(q))$. We mention that the latter approach can be improved to $\tilde{O}((1 + T/\log\log(D))\log(D)\log(q))$ using Yao's result [25] on simultaneous exponentiation. When $R = \mathbb{Z}$, the best known approach to avoid expression swell is to pick a random prime p and to perform the evaluations modulo p. One needs to choose $p > D$ in order to have a nonzero probability of success. Therefore, the bit complexity contains a $T\log^2 D$ factor.

Our approach to obtain a quasi-linear complexity is to perform the evaluation *modulo* $X^p - 1$ for some random prime p. This requires to evaluate the polynomial $[(FG) \bmod X^p - 1]$ on α without computing it.

3.1 Modular product evaluation

Lemma 3.1. *Let* F *and* G *be two sparse polynomials in* $R[X]$ *with* $\deg F, \deg G \leq p-1$ *and* $\alpha \in R$. *Then* $(FG) \bmod X^p - 1$ *can be evaluated on* α *using* $O((\#F + \#G)\log p)$ *operations in* R.

Proof. Let $H = (FG) \bmod X^p - 1$. The computation of H corresponds to the linear map

$$\underbrace{\begin{pmatrix} h_0 \\ h_1 \\ \vdots \\ h_{p-1} \end{pmatrix}}_{\vec{h}} = \underbrace{\begin{pmatrix} f_0 & f_{p-1} & \cdots & f_1 \\ f_1 & f_0 & \cdots & f_2 \\ \vdots & \vdots & & \vdots \\ f_{p-1} & f_{p-2} & \cdots & f_0 \end{pmatrix}}_{T_F} \underbrace{\begin{pmatrix} g_0 \\ g_1 \\ \vdots \\ g_{p-1} \end{pmatrix}}_{\vec{g}}$$

where f_i (resp. g_i, h_i) is the coefficient of degree i of F (resp. G, H). Computing $H(\alpha)$ corresponds to the inner product $\vec{\alpha}_p\vec{h} = \vec{\alpha}_p T_F \vec{g}$ where $\vec{\alpha}_p = (1, \alpha, \ldots, \alpha^{p-1})$. This evaluation can be computed in $O(p)$ operations in R [6]. Here we reuse similar techniques in the context of sparse polynomials.

To compute $H(\alpha)$, we first compute $\vec{c} = \vec{\alpha}_p T_F$, and then the inner product $\vec{c}\vec{g}$. If $\mathrm{supp}(G) = \{j_1, \ldots, j_{\#G}\}$ with $j_1 < \cdots < j_{\#G} < p$, we only need the corresponding entries of $\vec{c}$, that is all c_{j_k}'s for $1 \leq k \leq \#G$. Since $c_j = \sum_{\ell=0}^{p-1} \alpha^\ell f_{(\ell-j) \bmod p}$, we can write $c_j = f_{p-j} + \alpha \sum_{\ell=0}^{p-2} \alpha^\ell f_{(\ell-j+1) \bmod p}$, that is $c_j = \alpha c_{j-1} + (1-\alpha^p) f_{p-j}$.

Applying this relation as many times as necessary, we obtain a relation to compute $c_{j_{k+1}}$ from c_{j_k}:

$$c_{j_{k+1}} = \alpha^{j_{k+1}-j_k} c_{j_k} + (1-\alpha^p) \sum_{\ell=j_k+1}^{j_{k+1}} \alpha^\ell f_{p-\ell}.$$

Each nonzero coefficient f_t of F appears in the definition of $c_{j_{k+1}}$ if and only if $p - j_{k+1} \leq t < p - j_k$. Thus, each f_t is used exactly once to compute all the c_{j_k}'s. Since for each summand, one needs to compute α^ℓ for some $\ell < p$, the total cost for computing all the sums is $O(\#F \log p)$ operations in R. Similarly, the computation of $\alpha^{j_{k+1}-j_k} c_{j_k}$ for all k costs $O(\#G \log p)$. The last remaining step is the final inner product which costs $O(\#G)$ operations in R, whence the result. □

The complexity is improved to $O(\log p + (\#F + \#G)\log p/\log\log p)$ using again Yao's algorithm [25] for simultaneous exponentiation.

3.2 A quasi-linear time algorithm

Given three sparse polynomials F, G and H in $R[X]$, we want to assert that $H = FG$. Our approach is to take a random prime p and to verify this assertion modulo $X^p - 1$ through modular product evaluation. This method is explicitly described in the algorithm VerifySP that works over any large enough integral domain R. We further extend the description and the analysis of this algorithm for the specific cases $R = \mathbb{Z}$ and $R = \mathbb{F}_q$ in the next sections.

Algorithm 1 VerifySP

Input: $H, F, G \in R[X]$; $0 < \epsilon < 1$.
Output: True if $FG = H$, False with probability $\geq 1 - \epsilon$ otherwise.

1: Define $c_1 > \frac{10}{3}$ and $c_2 > 1$ such that $\frac{10}{3c_1} + (1 - \frac{10}{3c_1})\frac{1}{c_2} \leq \epsilon$
2: $D \leftarrow \deg(H)$
3: **if** $\#H > \#F\#G$ or $D \neq \deg(F) + \deg(G)$ **then return** False
4: $\lambda \leftarrow \max(21, c_1(\#F\#G + \#H)\ln D)$
5: $p \leftarrow \text{RandomPrime}(\lambda, \frac{5}{3c_1})$
6: $(F_p, G_p, H_p) \leftarrow (F \bmod X^p - 1,\ G \bmod X^p - 1,\ H \bmod X^p - 1)$
7: Define $\mathcal{E} \subset R$ of size $> c_2 p$ and choose $\alpha \in \mathcal{E}$ randomly.
8: $\beta \leftarrow [(F_p G_p) \bmod X^p - 1](\alpha)$ ▷ using Lemma 3.1
9: **return** $\beta = H_p(\alpha)$

Theorem 3.2. *If R is an integral domain of size $\geq 2c_1c_2\#F\#G\ln D$ VerifySP works as specified and it requires $O_\epsilon(T\log(T\log D))$ operations in R plus $O_\epsilon(T\mathsf{I}(\log D))$ bit operations where $D = \deg(H)$ and $T = \max(\#F, \#G, \#H)$.*

Proof. Step 3 dismisses two trivial mistakes and ensures that D is a bound on the degree of each polynomial. If $FG = H$, the algorithm returns True for any choice of p and α. Otherwise, there are two sources of failure. Either $X^p - 1$ divides $FG - H$, whence $(FG)_p(\alpha) = H_p(\alpha)$ for any α. Or α is a root of the nonzero polynomial $(FG-H) \bmod X^p - 1$. Since $FG-H$ has at most $\#F\#G+\#H$ terms, the first failure occurs with probability at most $\frac{10}{3c_1}$ by Prop. 2.4. And since $(FG - H) \bmod X^p - 1$ has degree at most $p - 1$ and $\mathcal{E}$ has $c_2 p$ points, the second failure occurs with probability at most $\frac{1}{c_2}$. Altogether, the failure probability is at most $\frac{10}{3c_1} + (1 - \frac{10}{3c_1})\frac{1}{c_2}$.

Let us remark that $c_1, c_2 = O(\frac{1}{\epsilon})$ and $p = O(\frac{1}{\epsilon}T^2\log D)$. Step 5 requires only $\tilde{O}(\log^3(\frac{1}{\epsilon}T\log D))$ bit operations by Proposition 2.2. The operations in Step 6 are T divisions by p on integers bounded by D which cost $O_\epsilon(T\mathsf{I}(\log D))$ bit operations, plus T additions in R. The evaluation of $F_pG_p \bmod X^p - 1$ on α at Step 8 requires $O(T\log(\frac{1}{\epsilon}T\log D))$ operations in R by Lemma 3.1. The evaluation of H_p on α costs $O(T\log(T\log D))$ operations in R. Other steps have negligible costs. □

3.3 Analysis over finite fields

The first easy case is the case of large finite fields: If there are enough points for the evaluation, the generic algorithm has the same guarantee of success and a quasi-linear time complexity.

Corollary 3.3. *Let F, G and H be three polynomials of degree at most D and sparsity at most T in $\mathbb{F}_q[X]$ where $q > 2c_1c_2\#F\#G\ln(D)$. Then Algorithm VerifySP has bit complexity $O_\epsilon(n\log^2(n)4^{\log^* n})$ where $n = T(\log D + \log q)$ is the input size.*

Proof. By definition of n, the cost of Step 6 is $O_\epsilon(n\log n)$ bit operations. Each ring operation in $\mathbb{F}_q$ costs $O(\log(q)\log\log(q)4^{\log^* q})$ bit operations which implies that the bit complexity of Step 8 is $O_\epsilon(T\log(T\log D)\log(q)\log\log(q)4^{\log^* q})$. Since $T\log q$ and $T\log D$ are bounded by n and $\log\log q \leq \log n$, the result follows. □

We shall note that even if $q < 2c_1c_2\#F\#G\ln(D)$ we can make our algorithm to work by using an extension field and this approach achieves the same complexity.

Theorem 3.4. *One can adapt algorithm VerifySP to work over finite fields $\mathbb{F}_q$ such that $q < 2c_1c_2\#F\#G\ln(D)$. The bit complexity is $O_\epsilon(n\log(n)\log\log(n)4^{\log^* n})$, where $n = T(\log D + \log q)$ is the input size.*

Proof. To have enough elements in the set $\mathcal{E}$, we need to work over $\mathbb{F}_{q^s}$ where $q^s > c_2p \geq q^{s-1}$. An irreducible degree-s polynomial can be computed in $\tilde{O}(s^3\log q) = \tilde{O}(\log(T\log D)/\log q)$ by Proposition 2.6. Since α is taken in $\mathbb{F}_{q^s}$, the complexity becomes $O_\epsilon(T\mathsf{I}(\log D) + T\log(T\log D)\mathsf{M}_q(s))$ bit operations. Remarking that $T \leq D$ we have $T\log(T\log D) \leq T\log(D\log D) = O(n)$. Since $s\log q = O(\log(T\log D)) = O(\log n)$ we can obtain $\mathsf{M}_q(s) = O(\log(n)\log\log(n)4^{\log^* n})$ which implies that the second term of the complexity is $O(n\log(n)\log\log(n)4^{\log^* n})$. The first term is negligible since it is $O(n\log n)$.

In order to achieve the same probability of success, we fix an error probability $1/c_3 < 1$ for Proposition 2.6 and we take constants c_1 and c_2 in VerifySP such that $1-(1-\frac{10}{3c_1})(1-\frac{1}{c_2})(1-\frac{1}{c_3}) \leq \epsilon$. □

We note that for very sparse polynomials over some fields, the complexity is only dominated by the operations on the exponents.

Corollary 3.5. *VerifySP has bit complexity $O_\epsilon(n\log n)$ in the following cases:*

(i) $s = 1$ and $\log q = O(\log^{1-\alpha} D)$ for some constant $0 < \alpha < 1$,
(ii) $s > 1$ and $T = \Theta(\log^k D)$ for some constant k.

Proof. In both cases the cost of reducing the exponents *modulo* p is $O_\epsilon(n\log n)$ bit operations. In the first case, each multiplication in $\mathbb{F}_q$ costs $O(\log(q)\log\log(q)4^{\log^* q}) = O(\log D)$ bit operations as $\log\log(q)4^{\log^* q} = O(\log^\alpha D)$. In the second case, $n = O(\log^{k+1} D)$ and $s\log q = O_\epsilon(\log(T^2\log D)) = O_\epsilon(\log\log D)$ which implies $\mathsf{M}_q(s) = O_\epsilon(s\log(q)\log(s\log q)4^{\log^* s}) = O_\epsilon(\log D)$. In both cases, the algorithm performs $O_\epsilon(T\log(T\log D)) = O_\epsilon(T\log n)$ operations in $\mathbb{F}_q$ (or in $\mathbb{F}_{q^s}$). Therefore the bit complexity is $O_\epsilon(n\log n)$. □

The following generalization is used in our quasi-linear multiplication algorithm given in Section 4.

Corollary 3.6. *Let $(F_i, G_i)_{0\leq i<m}$ and H be sparse polynomials over $\mathbb{F}_q$ of degree at most D and sparsity at most T. We can verify if $\sum_{i=0}^{m-1} F_iG_i = H$, with error probability at most ϵ when they are different, in $O_\epsilon(m(T\mathsf{I}(\log D)+T\log(mT\log D)\mathsf{M}_q(s)))$ bit operations.*

3.4 Analysis over the integers

In order to keep a quasi-linear time complexity over the integers, we must work over a prime finite field $\mathbb{F}_q$ to avoid the computation of too large integers. Indeed, $H_p(\alpha)$ could have size $p\log(\alpha) = O_\epsilon(T^2\log(D)\log(\alpha))$ which is not quasi-linear in the input size.

Theorem 3.7. *One can adapt algorithm VerifySP to work over the integers. The bit complexity is $O_\epsilon(n\log n\log\log n)$, where $n = T(\log D+\log C)$ is the input size with $C = \max(\|F\|_\infty, \|G\|_\infty, \|H\|_\infty)$.*

Proof. Before Step 6, we choose a random prime number $q = \text{RandomPrime}(\mu, \frac{5}{3c_2})$ with $\mu = c_2\max(p, \ln(C^2T + C))$ and we perform all the remaining steps modulo q. Let us assume that the polynomial $\Delta = FG - H \in \mathbb{Z}[X]$ is nonzero. Our algorithm only fails in

the following three cases: p is such that $\Delta_p = \Delta \bmod X^p - 1 = 0$; q is such that $\Delta_p \equiv 0 \bmod q$; α is a root of Δ_p in $\mathbb{F}_q$.

Using Proposition 2.4, Δ_p is nonzero with probability at least $1 - \frac{10}{3c_1}$. Actually, with the same probability, the proof of the proposition shows that at least one coefficient of Δ is preserved in Δ_p. Since $\|\Delta\|_\infty \leq C^2T + C$, Proposition 2.5 ensures that $\Delta_p \not\equiv 0 \bmod q$ with probability at least $1 - \frac{10}{3c_2}$. Finally, q has been chosen so that $\mathbb{F}_q$ has at least $c_2 p$ elements whence α is not a root of $\Delta_p \bmod q$ with probability at least $1 - \frac{1}{c_2}$. Altogether, taking $c_1, c_2 \geq \frac{10}{3}$ such that $1 - (1 - \frac{10}{3c_1})(1 - \frac{10}{3c_2})(1 - \frac{1}{c_2}) \leq \epsilon$, our adaptation of VerifySP has an error probability at most ϵ.

The reductions of F, G and H *modulo* q add a term $O(T\mathsf{I}(\log C))$ to the complexity. Since operations in $\mathbb{F}_q$ have cost $\mathsf{I}(\log q)$, the complexity becomes $O(T\mathsf{I}(\log D) + T\mathsf{I}(\log C) + T\log(T\log D)\mathsf{I}(\log q))$ bit operations. The first two terms are in $O(n \log n)$. Moreover, $q = O_\epsilon(\log(C^2T) + p)$ and $p = O_\epsilon(T^2 \log D)$, thus $\log q = O_\epsilon(\log(\log C + T \log D)) = O_\epsilon(\log n)$. Since $T \leq D$, $T \log(T \log D) = O(n)$ and the third term in the complexity is $O_\epsilon(n \log n \log\log n)$. □

As over small finite fields, the complexity is actually better for very sparse polynomials.

Corollary 3.8. *If $T = \Theta(\log^k D)$ for some k, VerifySP has bit complexity $O_\epsilon(n \log n)$.*

Proof. If $T = \Theta(\log^k D)$, $T \log(T \log D) = \tilde{O}(\log^k D) = o(n)$, thus the last term of the complexity in the proof of Theorem 3.7 becomes negligible with respect to the first two terms. □

For the same reason as for finite fields, we extend the verification algorithm to a sum of products.

Corollary 3.9. *Let $(F_i, G_i)_{0 \leq i < m}$ and H be sparse polynomials of degree at most D, sparsity at most T, and height at most C. We can verify if $\sum_{i=0}^{m-1} F_i G_i = H$, with probability of error at most ϵ when they are different, in $O_\epsilon(mT\mathsf{I}(\log D) + mT\mathsf{I}(\log C) + mT \log(mT \log D)\mathsf{I}(\log(m \log C + mT \log D)))$ bit operations.*

We shall only use this algorithm with $m = 2$ and thus refer to it as VerifySumSP$(H, F_0, G_0, F_1, G_1, \epsilon)$.

4 SPARSE POLYNOMIAL MULTIPLICATION

Given two sparse polynomials F and G, our algorithm aims at computing the product $H = FG$ through sparse polynomial interpolation. We avoid the difficulty of computing an *a priori* bound on the sparsity of H needed for sparse interpolation by using our verification algorithm of Section 3. Indeed, one can start with an arbitrary small sparsity and double it until the interpolated polynomial matches the product according to VerifySP.

The remaining difficulty is to interpolate H in quasi-optimal time given a sparsity bound, which is not yet achieved in the general case. In our case, we first analyze the complexity of Huang's sparse interpolation algorithm [12] when the input is a sum of sparse products. In order to obtain the desired complexity we develop a novel approach that interleaves two levels of Huang's algorithm.

4.1 Analysis of Huang's sparse interpolation

In [12] Huang proposes an algorithm that interpolates a sparse polynomial H from its SLP representation, achieving the best known complexity for this problem, though it is not optimal. Its main idea is to use the dense polynomials $H_p = H \bmod X^p - 1$ and $H'_p = H' \bmod X^p - 1$ where H' is the derivative of H and p a *small* random prime. Indeed, if cX^e is a term of H that does not collide during the reduction *modulo* $X^p - 1$, H_p contains the monomial $cX^{e \bmod p}$ and H'_p contains $ceX^{e-1 \bmod p}$, hence c and e can be recovered by a mere division. Of course, the choice of p is crucial for the method to work. It must be small enough to get a low complexity, but large enough for collisions to be sufficiently rare.

Lemma 4.1. *There exists an algorithm FindTerms that takes as inputs a prime p, two polynomials $H_p = H \bmod X^p - 1$, $H'_p = H' \bmod X^p - 1$, and bounds $D \geq \deg(H)$ and $C \geq \|H\|_\infty$ and it outputs an approximation H^* of H that contains at least all the monomials of H that do not collide* modulo $X^p - 1$. *Its bit complexity is $O(T\mathsf{I}(\log CD))$, where $T = \#H$.*

Proof. It is a straightforward adaptation of [12, Algorithm 3.4 (UTerms)]. Here, taking C as input allows us to only recover coefficients that are at most C in absolute value and therefore to perform divisions with integers of bitsize at most $\log(CD)$. □

Corollary 4.2. *Let H be a sparse polynomial such that $\#H \leq T$, $\deg H \leq D$ and $\|H\|_\infty \leq C$, and $0 < \epsilon < 1$. If $\lambda = \max(21, \frac{10}{3\epsilon} T^2 \ln D)$ and $p = \mathrm{RandomPrime}(\lambda, \frac{\epsilon}{2})$, then with probability at least $1 - \epsilon$, FindTerms$(p, H \bmod X^p - 1, H' \bmod X^p - 1, D, C)$ returns H.*

Proof. With probability at least $1 - \epsilon$, no collision occurs in $H \bmod X^p - 1$, and consequently neither in $H' \bmod X^p - 1$, by Proposition 2.3. In this case FindTerms correctly computes H, according to Lemma 4.1. □

Theorem 4.3. *There exists an algorithm InterpSumSP that takes as inputs $2m$ sparse polynomials $(F_i, G_i)_{0 \leq i < m}$, three bounds $T \geq \#H$, $D > \deg(H)$ and $C \geq \|H\|_\infty$ where $H = \sum_{i=0}^{m-1} F_i G_i$, a constant $0 < \mu < 1$ and the list $\mathcal{P}$ of the first $2N$ primes for $N = \max(1, \lfloor \frac{32}{5}(T - 1)\log D \rfloor)$, and outputs H with probability at least $1 - \mu$.*

Its bit complexity is $\tilde{O}_\mu(mT_1 \log(D_1) \log(C_1 D_1))$ where T_1, D_1 and C_1 are bounds on the sparsity, the degree and the height of H and each F_i and G_i.

Proof. It is identical to the proof of [12, Algorithm 3.9 (UIPoly)] taking into account that H is not given as an SLP anymore but as $\sum_{i=0}^{m-1} F_i G_i$ where the polynomials F_i and G_i are given as sparse polynomials. □

Remark 4.4. *A finer analysis of algorithm InterpSumSP leads to a bit complexity $O_\mu(m \log T_1 \mathsf{M}_\mathbb{Z}(T_1 \log(D_1) \log(T_1 \log D_1), T_1 C_1 D_1))$.*

Remark 4.5. *Even when InterpSumSP returns an incorrect polynomial, it has sparsity at most $2T$, degree less than D and coefficients bounded by C.*

4.2 Multiplication

Our idea is to compute different candidates to FG with a growing sparsity bound and to verify the result with VerifySP. Unfortunately, a direct call to InterpSumSP with the correct sparsity $T = \max(\#F, \#G, \#(FG))$ yields a bit complexity $\tilde{O}(T \log(D) \log(CD))$ if the coefficients are bounded by C and the degree by D. We shall

remark that it is not nearly optimal since the input and output size are bounded by $T \log D + T \log C$.

To circumvent this difficulty, we first compute the reductions $F_p = F \bmod X^p - 1$ and $G_p = G \bmod X^p - 1$ of the input polynomials, as well as the reductions $F'_p = F' \bmod X^p - 1$ and $G'_p = G' \bmod X^p - 1$ of their derivatives, for a random prime p as in Corollary 4.2. The polynomials $H_p = FG \bmod X^p - 1$ and $H'_p = (FG)' \bmod X^p - 1$ can be computed using InterpSumSP and VerifySP. Indeed, we first compute F_pG_p by interpolation and then reduce it *modulo* $X^p - 1$ to get H_p. Similarly for H'_p we first interpolate $F'_pG_p + F_pG'_p$ before its reduction. Finally we can compute the polynomial FG from H_p and H'_p using FindTerms according to Corollary 4.2. Our choice of p, which is polynomial in the input size, ensures that each call to InterpSumSP remains quasi-linear.

Algorithm 2 SparseProduct

Input: $F, G \in \mathbb{Z}[X]$. $0 < \mu_1, \mu_2 < 1$ with $\frac{\mu_1}{2} \le \mu_2$.
Output: $H \in \mathbb{Z}[X]$ s.t. $H = FG$ with probability at least $1 - \mu_1$.

1: $t \leftarrow \max(\#F, \#G)$, $D \leftarrow \deg(F) + \deg(G)$, $C \leftarrow t\|F\|_\infty\|G\|_\infty$
2: $\lambda \leftarrow \max(21, \frac{20}{3\mu_1}(\#F\#G)^2 \ln D)$, $\mu^* \leftarrow \mu_2 - \frac{\mu_1}{2}$
3: $p \leftarrow \text{RandomPrime}(\lambda, \frac{\mu_1}{4})$
4: $F_p \leftarrow F \bmod X^p - 1$, $G_p \leftarrow G \bmod X^p - 1$
5: $F'_p \leftarrow F' \bmod X^p - 1$, $G'_p \leftarrow G' \bmod X^p - 1$
6: **repeat**
7: $\quad N \leftarrow \max(1, \lfloor \frac{32}{5}(t-1)\log p \rfloor)$
8: $\quad \mathcal{P} \leftarrow \{\text{the first } 2N \text{ primes in increasing order}\}$
9: $\quad H_1 \leftarrow \text{InterpSumSP}([(F_p, G_p)], t, 2p, C, \frac{\mu^*}{2}, \mathcal{P})$
10: $\quad H_2 \leftarrow \text{InterpSumSP}([(F_p, G'_p), (F'_p, G_p)], t, 2p, CD, \frac{\mu^*}{2}, \mathcal{P})$
11: $\quad t \leftarrow 2t$
12: **until**
$\quad \text{VerifySP}(H_1, F_p, G_p, \frac{\mu_1}{2})$ **and** ▷ $H_1 = F_pG_p$
$\quad \text{VerifySumSP}(H_2, F_p, G'_p, F'_p, G_p, \frac{\mu_1}{2})$ ▷ $H_2 = F'_pG_p + F_pG'_p$
13: $H_p \leftarrow H_1 \bmod X^p - 1$, $H'_p \leftarrow H_2 \bmod X^p - 1$.
14: **return** FindTerms (p, H_p, H'_p, D, C).

Lemmas 4.6 and 4.7 respectively provide the correctness and complexity bound of algorithm SparseProduct. Together, they consequently form a proof of Theorem 1.1 by taking $\epsilon = \mu_1 + \mu_2$. Note that this approach translates *mutatis mutandis* to the multiplication of sparse polynomials over $\mathbb{F}_q$ where the characteristic of $\mathbb{F}_q$ is larger than D.

Lemma 4.6. *Let F and G be two sparse polynomials over $\mathbb{Z}$. Then algorithm SparseProduct returns FG with probability at least $1 - \mu_1$.*

Proof. Since FG has sparsity at most $\#F\#G$, Corollary 4.2 implies that if $H_p = FG \bmod X^p - 1$ and $H'_p = (FG)' \bmod X^p - 1$, the probability that FindTerms does not return FG is at most $\frac{\mu_1}{2}$. The other reason for the result to be incorrect is that one of these equalities does not hold, which means that one of the two verifications fails. Since this happens with probability at most $\frac{\mu_1}{2}$, SparseProduct returns FG with probability at least $1 - \mu_1$. □

Lemma 4.7. *Let F and G be two sparse polynomials over $\mathbb{Z}$, $T = \max(\#F, \#G, \#(FG))$, $D = \deg(FG)$, $C = \max(\|F\|_\infty, \|G\|_\infty, \|FG\|_\infty)$ and $\epsilon = \mu_1 + \mu_2$. Then algorithm SparseProduct has bit complexity $\tilde{O}_\epsilon(T(\log D + \log C))$ with probability at least $1 - \mu_2$. Writing $n = T(\log D + \log C)$, the bit complexity is $O_\epsilon(n \log^2 n \log^2 T(\log T + \log\log n))$.*

Proof. In order to obtain the given complexity, we first need to prove that with high probability InterpSumSP never computes polynomials with a sparsity larger than $4\#(FG)$.

Let $T_p = \max(\#(F_pG_p), \#(F_pG'_p + F'_pG_p))$. If $t \le 2T_p$ then the polynomials H_1 and H_2 satisfy $\#H_1, \#H_2 \le 4T_p$ by Remark 4.5. Unfortunately, T_p could be as large as T^2 and t might reach values larger than T_p. We now prove that: *(i)* with probability at least $1 - \mu^*$ the maximal value of t during the algorithm is less than $2T_p$; *(ii)* with probability at least $1 - \frac{\mu_1}{2}$, $T_p \le \#(FG)$. Together, this will prove that $\#H_1, \#H_2 \le 4\#(FG)$ with probability at least $1 - \mu^* - \frac{\mu_1}{2} = 1 - \mu_2$.

(i) As soon as $t \ge T_p$, Steps 9 and 10 compute both F_pG_p and $F_pG'_p + F'_pG_p$ with probability at least $1 - \mu^*$ by Theorem 4.3. Since VerifySP never fails when the product is correct, the algorithm ends when $T_p \le t < 2T_p$ with probability at least $1 - \mu^*$.

(ii) Let us define the polynomials $\hat{F}_p$ and $\hat{G}_p$ obtained from F_p and G_p by replacing each nonzero coefficient by 1. The choice of p in Step 3 ensures that with probability at least $1 - \frac{\mu_1}{2}$ there is no collision in $(\hat{F}_p\hat{G}_p) \bmod X^p - 1$ by applying Proposition 2.3 to the product $\hat{F}_p\hat{G}_p$. In that case, there is also no collision in $F_pG_p \bmod X^p - 1$ and in $F_pG'_p + F'_pG_p \bmod X^p - 1$ since $\text{supp}(F_pG_p) \subset \text{supp}(\hat{F}_p\hat{G}_p)$. Therefore, there are as many nonzero coefficients in F_pG_p as in $F_pG_p \bmod X^p - 1$, which is equal to $FG \bmod X^p - 1$. Thus with probability at least $1 - \frac{\mu_1}{2}$ we have $\#(F_pG_p) = \#(FG) \le T$ and similarly $\#(F'_pG_p + F_pG'_p) = \#((FG)') \le T$.

In the rest of the proof, we assume that the loop stops with $t \le 2T_p$ and that $T_p \le T$. In particular, the number of iterations of the loop is $O(\log T)$. Since $2p = O(\frac{1}{\epsilon}T^4 \log D)$, Steps 9 and 10 have a bit complexity $\tilde{O}_\epsilon(T \log(p) \log(pCD)) = \tilde{O}_\epsilon(T \log CD)$ by Theorem 4.3. Using Remark 4.5, VerifySP and VerifySumSP have polynomials of height at most tCD as inputs. By Corollary 3.9, Step 12 has bit complexity $O_\epsilon(T \log(T \log p)\mathsf{I}(\log CD)) = \tilde{O}_\epsilon(T \log CD)$. The list $\mathcal{P}$ can be computed incrementally, adding new primes when necessary. At the end of the loop, $\mathcal{P}$ contains $O(T \log 2p)$ primes, which means that it is computed in $O_\epsilon(T \log(p) \log^2(T \log p) \log\log(T \log p))$ bit operations [24, Chapter 18], that is $\tilde{O}_\epsilon(T \log\log D)$ since $\log p = O(\log(T \log D))$.

The total cost for the $O(\log T)$ iterations of the loop is still $\tilde{O}_\epsilon(T \log(CD))$. Step 14 runs in time $O_\epsilon(T\mathsf{I}(\log CD))$ by Lemma 4.1 as the coefficients of H'_p are bounded by $2TC^2D$ with $T \le D$ and $\#H_p, \#H'_p \le \#H$. Since other steps have negligible costs this yields a complexity of $\tilde{O}_\epsilon(T(\log C + \log D))$ with probability at least $1 - \mu_2$.

Using Remark 4.4, we can provide a more precise complexity for Steps 9 and 10 which is $O_\epsilon(\log T \mathsf{M}_\mathbb{Z}(T \log(p) \log(T \log p), pDTC))$ bit operations. It is easy to observe that the $\log T$ repetitions of these steps provide the dominant term in the complexity. A careful simplification yields a bit complexity $O_\epsilon(n \log^2 n \log^2 T(\log T + \log\log n)$ for SparseProduct where $n = T(\log D + \log C)$ bounds both input and output sizes. □

4.3 Multivariate case

Using classical Kronecker substitution [24, Chapter 8] one can extend straightforwardly SparseProduct to multivariate polynomials. Let $F, G \in \mathbb{Z}[X_1, \ldots, X_n]$ with $\|F\|_\infty, \|G\|_\infty \leq C$ and $\deg_{X_i}(F) + \deg_{X_i}(G) < d$. Writing $F_u(X) = F(X, X^d, \ldots, X^{d^{n-1}})$ and $G_u(X) = G(X, X^d, \ldots, X^{d^{n-1}})$, one can easily retrieve FG from the univariate product $F_u G_u$. It is easy to remark that the Kronecker substitution preserve the sparsity and the height, and it increases the degree to $\deg F_u, \deg G_u < d^n$. If F and G are sparse polynomials with at most T nonzero terms, their sizes are at most $T(n \log d + \log C)$ which is exactly the sizes of F_u and G_u. Since the Kronecker and inverse Kronecker substitutions cost $\tilde{O}(Tn \log d)$ bit operations, one can compute $F_u G_u$ using SparseProduct within the following bit complexity.

Corollary 4.8. *There exists an algorithm that takes as inputs $F, G \in \mathbb{Z}[X_1, \ldots, X_n]$ and $0 < \epsilon < 1$, and computes FG with probability at least $1 - \epsilon$, using $\tilde{O}_\epsilon(T(n \log d + \log C))$ bit operations where $T = \max(\#F, \#G, \#(FG))$, $d = \max_i(\deg_{X_i} FG)$ and $C = \max(\|F\|_\infty, \|G\|_\infty)$.*

Over a finite field $\mathbb{F}_{q^s}$ for some prime q, the previous technique requires that $q > d^n$ since SparseProduct requires q to be larger than the degree. The *randomized Kronecker substitution* method introduced by Arnold and Roche [3] allows to apply SparseProduct to fields of smaller characteristic. The idea is to define univariate polynomials $F_s(X) = F(X^{s_1}, \ldots, X^{s_n})$ and $G_s(X) = G(X^{s_1}, \ldots, X^{s_n})$ for some random vector $\vec{s} = (s_1, \ldots, s_n)$ such that these polynomials have much smaller degrees than those obtained with classical Kronecker substitution. As a result, we obtain an algorithm that works for much smaller q of order $\tilde{O}(nd\#F\#G)$.

Our approach is to first use some randomized Kronecker substitutions to estimate the sparsity of FG by computing the sparsity of $H_s = F_s G_s$ for several distinct random vectors $\vec{s}$. With high probability, the maximal sparsity is close to the one of FG. Then, we use this information to provide a bound to some (multivariate) sparse interpolation algorithm. Note that our approach is inspired from [13] that slightly improves randomized Kronecker substitution.

Lemma 4.9. *Let $H \in \mathbb{F}_{q^s}[X_1, \ldots, X_n]$ of sparsity T, and $\vec{s}$ be a vector chosen uniformly at random in S^n where $S \subset \mathbb{N}$ is finite. The expected sparsity of $H_s(X) = H(X^{s_1}, \ldots, X^{s_n})$ is at least $T(1 - \frac{T-1}{\#S})$.*

Proof. If we fix two distinct exponent vectors $\vec{e}_u$ and $\vec{e}_v$ of H, they *collide* in H_s if and only if $\vec{e}_u \cdot \vec{s} = \vec{e}_v \cdot \vec{s}$. Since $\vec{e}_u \neq \vec{e}_v$, they differ at least on one component, say $e_{u,j_0} \neq e_{v,j_0}$. The equality $\vec{e}_u \cdot \vec{s} = \vec{e}_v \cdot \vec{s}$ is then equivalent to

$$s_{j_0} = \sum_{j \neq j_0} \frac{e_{v,j} - e_{u,j}}{e_{u,j_0} - e_{v,j_0}} s_j.$$

Writing Y for the right-hand side of this equation we have

$$\Pr[\vec{e}_u \cdot \vec{s} = \vec{e}_v \cdot \vec{s}] = \Pr[s_{j_0} = Y] = \sum_y \Pr[s_{j_0} = Y | Y = y] \Pr[Y = y]$$

where the (finite) sum ranges over all possible values y of Y. Since s_{j_0} is chosen uniformly at random in S, $\Pr[s_{j_0} = Y | Y = y] = \Pr[s_{j_0} = y] \leq 1/\#S$ and the probability that $\vec{e}_u$ and $\vec{e}_v$ collide is at most $1/\#S$. This implies that the expected number of vectors that collide is at most $T(T-1)/\#S$. □

Corollary 4.10. *Let H be as in Lemma 4.9 and $\vec{v}_1, \ldots, \vec{v}_\ell \in S^n$ be some vectors chosen uniformly and independently at random. Then $\Pr[\max_i \#H_{v_i} \leq T(1 - 2\frac{T-1}{\#S})] \leq 1/2^\ell$.*

Proof. For each $\vec{v}_i$, the expected number of terms that collide in $H_{v_i}(X)$ is at most $T(T-1)/\#S$ by Lemma 4.9. Using Markov's inequality, we have $\Pr[\#H_{v_i} \leq T - 2T(T-1)/\#S] \leq 1/2$. Since the vectors $\vec{v}_i$ are independent, the result follows. □

Algorithm 3 SparsityEstimate

Input: $F, G \in \mathbb{F}_{q^s}[X_1, \ldots, X_n]$, $0 < \epsilon < 1$, $\lambda > 1$.
Output: An integer t such that $t \leq \lambda \#(FG)$.

1: $N \leftarrow \lceil 2\frac{\#F\#G-1}{1-1/\lambda} \rceil$, $\ell \leftarrow \lceil \log \frac{2}{\epsilon} \rceil$.
2: $t' \leftarrow 0$, $\mu \leftarrow \frac{\epsilon}{4\ell}$.
3: **repeat** ℓ times
4: $\quad \vec{s} \leftarrow$ random element of $\{0, \ldots, N-1\}^n$.
5: $\quad F_s \leftarrow F(X^{s_1}, \ldots, X^{s_n})$, $G_s \leftarrow G(X^{s_1}, \ldots, X^{s_n})$
6: $\quad H_s \leftarrow$ SparseProduct(F_s, G_s, μ, μ)
7: $\quad t' \leftarrow \max(t', \#H_s)$
8: **return** $\lambda t'$.

Lemma 4.11. *Algorithm SparsityEstimate is correct when $q \geq \frac{4D\#F\#G}{1-1/\lambda}$ where $D = \max(\deg F, \deg G)$. With probability at least $1 - \epsilon$, it returns an integer $t \geq \#(FG)$ using $\tilde{O}_\epsilon(T(n \log d + s \log q))$ bit operations where $T = \max(\#(FG), \#F, \#G)$ and $d = \max_i(\deg_{X_i} FG)$.*

Proof. Since each polynomial H_s has sparsity at most $\#(FG)$, SparsityEstimate returns an integer bounded by $\lambda\#(FG)$. SparseProduct can be used in step 5 since $\deg H_s = \deg F_s + \deg G_s \leq 2ND \leq q$ by the definition of N. Assuming that SparseProduct returns no incorrect answer during the loop, Corollary 4.10 applied to the product FG implies that $t' \geq \#(FG)(1 - 2(\#(FG) - 1)/N)$ with probability $\geq 1 - \epsilon/2$ at the end of the loop. By definition of N and since $\#F\#G \geq \#(FG)$, $t' \geq \#(FG)/\lambda$. Taking into account the probability of failure of SparseProduct, the probability that $\lambda t' \geq \#(FG)$ is at least $1 - \frac{3\epsilon}{4}$.

The computation of F_s and G_s requires $O(Tn\mathsf{l}(\log \max(d, N)) + Ts \log q)$ bit operations in Step 5. Since $\max(\#F_s, \#G_s, \#H_s) \leq T$ and $\deg H_s = O(ndT^2)$ in Step 6, the bit complexity of each call to SparseProduct is $\tilde{O}_\mu(T(\log(nd) + s \log q))$ with probability at least $1 - \mu$ using Lemma 4.7. Therefore, SparsityEstimate requires $\tilde{O}_\epsilon(T(n \log d + s \log q))$ bit operations with probability at least $1 - \epsilon/4$. Together with the probability of failure this concludes the proof. □

Theorem 4.12. *There exists an algorithm that takes as inputs two sparse polynomials F and G in $\mathbb{F}_{q^s}[X_1, \ldots, X_n]$ and $0 < \epsilon < 1$ that returns the product FG in $\tilde{O}_\epsilon(nT(\log d + s \log q))$ bit operations with probability at least $1 - \epsilon$, where $T = \max(\#F, \#G, \#(FG))$, $d = \max_i(\deg_{X_i} FG)$, $D = \deg FG$ and assuming that $q = \Omega(D\#F\#G + DT \log(D) \log(T \log D))$.*

Proof. The algorithm computes an estimate t on the sparsity of FG using SparsityEstimate$(F, G, \frac{\epsilon}{2}, \lambda)$ for some constant λ. The second step interpolates FG using Huang and Gao's algorithm [13, Algorithm 5 (MulPolySI)] which is parameterized by a univariate

sparse interpolation algorithm. Originally, its inputs are a polynomial given as a blackbox and bounds on its degree and sparsity. In our case, the blackbox is replaced by F and G, the sparsity bound is t and the univariate interpolation algorithm is SparseProduct.

The algorithm MulPolySI requires $O_\epsilon(n \log t + \log^2 t)$ interpolation of univariate polynomials with degree $\tilde{O}(tD)$ and sparsity at most t. Each interpolation with SparseProduct is done with μ_1, μ_2 such that $\mu_1 + \mu_2 = \epsilon/4(n+1)\log t$, so that MulPolySI returns the correct answer in $\tilde{O}_\epsilon(nT(\log d + s \log q))$ bit operations with probability at least $1 - \frac{\epsilon}{2}$ [13, Theorem 6]. Altogether, our two-step algorithm returns the correct answer using $\tilde{O}_\epsilon(nT(\log d + s \log q))$ bit operations with probability at least $1 - \epsilon$. The value of q is such that it bounds the degrees of the univariate polynomials returned by SparseProduct during the algorithm. □

4.4 Small characteristic

We now consider the case of sparse polynomial multiplication over a field $\mathbb{F}_{q^s}$ with characteristic smaller than the degree of the product FG (or, in the multivariate case, smaller than the degree of the product after randomized Kronecker substitution). We can no more use Huang's interpolation algorithm since it uses the derivative to encode the exponents into the coefficients and thus it only keeps the value of the exponents *modulo* q. Our idea to circumvent this problem is similar to the one in [2] that is to rather consider the polynomials over $\mathbb{Z}$ before calling our algorithm SparseProduct.

The following proposition is only given for the multivariate case as it encompasses univariate's one. It matches exactly with the complexity result given by Arnold and Roche [2].

Proposition 4.13. *There exists an algorithm that takes as inputs two sparse polynomials F and G in $\mathbb{F}_{q^s}[X_1, \ldots, X_n]$ and $0 < \epsilon < 1$ that returns the product FG in $\tilde{O}_\epsilon(S(n \log d + s \log q))$ bit operations with probability at least $1 - \epsilon$, where S is the structural sparsity of FG and $d = \max_i(\deg_{X_i} FG)$.*

Proof. If $s = 1$, the coefficients of F and G map easily to the integers in $\{0, \ldots, q-1\}$. Therefore, the product FG can be obtained by using an integer sparse polynomial multiplication, as the one in Corollary 4.8, followed by some reductions *modulo* q. Unfortunately, mapping the multiplication over the integers implies that the cancellations that could have occurred in $\mathbb{F}_q$ do not hold anymore. Consequently, the support of the product in $\mathbb{Z}$ before modular reduction is exactly the structural support of FG.

If $s > 1$, the coefficients of F and G are polynomials over $\mathbb{F}_q$ of degree $s-1$. As previously, mapping $\mathbb{F}_q$ to integers, F and G can be seen as $F_Y, G_Y \in \mathbb{Z}[Y][X_1, \ldots, X_n]$ where the coefficients are polynomials in $\mathbb{Z}[Y]$ of degree at most $s-1$ and height at most $q-1$.

If $T = \max(\#F, \#G)$, the coefficients of $F_Y G_Y$ are polynomials of degree at most $2s-2$ and height at most Tsq^2. Therefore, the product $FG \in \mathbb{F}_{q^s}$ can be computed by: *(i)* computing $F_B, G_B \in \mathbb{Z}[X_1, \ldots, X_n]$ by evaluating the coefficients of F_Y and G_Y at $B = Tsq^2$ (Kronecker substitution); *(ii)* computing the product $H_B = F_B G_B$; *(iii)* writing the coefficients of H_B in base B to obtain $H_Y = F_Y G_Y$ (Kronecker segmentation); *(iv)* and finally mapping back the coefficients of H_Y from $\mathbb{Z}[Y]$ to $\mathbb{F}_{q^s}$.

Similarly as the case $s = 1$, H_B and then H_Y have at most S nonzero coefficients. The Kronecker substitutions in *(i)* require $\tilde{O}(Ts \log q)$ bit operations, while the Kronecker segmentations in *(iii)* need $\tilde{O}(Ss \log q)$ bit operations. In *(iv)* we first compute Ss reductions *modulo* q on integers smaller than B, and then S polynomial divisions in $\mathbb{F}_q[Y]$ with polynomial of degree $O(s)$. Thus, it can be done in $\tilde{O}(Ss \log q)$ bit operations. Finally the computation in *(ii)* is dominant and it requires $\tilde{O}_\epsilon(S(n \log d + s \log q))$ bit operations with probability at least $1 - \epsilon$ using Corollary 4.8. □

REFERENCES

[1] A. Arnold. 2016. *Sparse polynomial interpolation and testing.* Ph.D. Dissertation. University of Waterloo.

[2] A. Arnold and D. S. Roche. 2015. Output-sensitive algorithms for sumset and sparse polynomial multiplication. In *ISSAC'15*. ACM, 29–36. https://doi.org/10.1145/2755996.2756653 arXiv:1501.05296

[3] A. Arnold and D. S. Roche. 2014. Multivariate sparse interpolation using randomized Kronecker substitutions. In *ISSAC'14*. ACM, 35–42. https://doi.org/10.1145/2608628.2608674 arXiv:1401.6694

[4] M. Ben-Or and P. Tiwari. 1988. A Deterministic Algorithm for Sparse Multivariate Polynomial Interpolation. In *STOC'88*. ACM, 301–309. https://doi.org/10.1145/62212.62241

[5] R. Cole and R. Hariharan. 2002. Verifying candidate matches in sparse and wildcard matching. In *STOC'02*. ACM, 592–601. https://doi.org/10.1145/509907.509992

[6] P. Giorgi. 2018. A probabilistic algorithm for verifying polynomial middle product in linear time. *Inform. Process. Lett.* 139 (2018), 30–34. https://doi.org/10.1016/j.ipl.2018.06.014

[7] D. Harvey and J. van der Hoeven. 2019. Faster polynomial multiplication over finite fields using cyclotomic coefficient rings. *J. Complexity* 54 (2019). https://doi.org/10.1016/j.jco.2019.03.004

[8] D. Harvey and J. van der Hoeven. 2019. Integer multiplication in time O(n log n). https://hal.archives-ouvertes.fr/hal-02070778

[9] J. van der Hoeven, R. Lebreton, and É. Schost. 2013. Structured FFT and TFT: Symmetric and Lattice Polynomials. In *ISSAC'13*. ACM, 355–362. https://doi.org/10.1145/2465506.2465526

[10] J. van der Hoeven and G. Lecerf. 2012. On the Complexity of Multivariate Blockwise Polynomial Multiplication. In *ISSAC'12*. ACM, 211–218. https://doi.org/10.1145/2442829.2442861

[11] J. van der Hoeven and G. Lecerf. 2013. On the bit-complexity of sparse polynomial and series multiplication. *J. Symb. Comput.* 50 (2013), 227–254. https://doi.org/10.1016/j.jsc.2012.06.004

[12] Q. Huang. 2019. Sparse Polynomial Interpolation over Fields with Large or Zero Characteristic. In *ISSAC'19*. ACM, 219–226. https://doi.org/10.1145/3326229.3326250

[13] Q. Huang and X. Gao. 2019. Revisit Sparse Polynomial Interpolation Based on Randomized Kronecker Substitution. In *CASC'19*. Springer, 215–235. https://doi.org/10.1007/978-3-030-26831-2_15

[14] S. C. Johnson. 1974. Sparse polynomial arithmetic. *ACM SIGSAM Bulletin* 8, 3 (1974), 63–71. https://doi.org/10.1145/1086837.1086847

[15] M. Monagan and R. Pearce. 2009. Parallel sparse polynomial multiplication using heaps. In *ISSAC'09*. ACM, 263. https://doi.org/10.1145/1576702.1576739

[16] M. Monagan and R. Pearce. 2011. Sparse polynomial division using a heap. *J. Symb. Comput.* 46, 7 (2011). https://doi.org/10.1016/j.jsc.2010.08.014

[17] V. Nakos. 2019. Nearly Optimal Sparse Polynomial Multiplication. arXiv:1901.09355

[18] D. A. Plaisted. 1984. New NP-hard and NP-complete polynomial and integer divisibility problems. *Theor. Comput. Sci.* 31, 1 (1984), 125–138. https://doi.org/10.1016/0304-3975(84)90130-0

[19] R. Prony. 1795. Essai expérimental et analytique sur les lois de la Dilatabilité de fluides élastique et sur celles de la Force expansive de la vapeur de l'eau et de la vapeur de l'alkool, à différentes températures. *J. École Polytechnique* 1, Floréal et Prairial III (1795), 24–76. https://gallica.bnf.fr/ark:/12148/bpt6k433661n/f32.item

[20] D. S. Roche. 2011. Chunky and equal-spaced polynomial multiplication. *J. Symb. Comput.* 46, 7 (2011), 791–806. https://doi.org/10.1016/j.jsc.2010.08.013

[21] D. S. Roche. 2018. What Can (and Can't) we Do with Sparse Polynomials?. In *ISSAC'18*. ACM, 25–30. https://doi.org/10.1145/3208976.3209027 arXiv:1807.08289

[22] J. B. Rosser and L. Schoenfeld. 1962. Approximate formulas for some functions of prime numbers. *Illinois J. Math.* 6, 1 (1962), 64–94. https://doi.org/10.1215/ijm/1255631807

[23] V. Shoup. 2008. *A Computational Introduction to Number Theory and Algebra* (second ed.). Cambridge University Press.

[24] J. von zur Gathen and J. Gerhard. 2013. *Modern Computer Algebra* (3rd ed.). Cambridge University Press.

[25] A. C. Yao. 1976. On the Evaluation of Powers. *SIAM J. Comput.* 5, 1 (1976), 100–103. https://doi.org/10.1137/0205008

Fast In-place Algorithms for Polynomial Operations: Division, Evaluation, Interpolation

Pascal Giorgi
LIRMM, Univ. Montpellier, CNRS
Montpellier, France
pascal.giorgi@lirmm.fr

Bruno Grenet
LIRMM, Univ. Montpellier, CNRS
Montpellier, France
bruno.grenet@lirmm.fr

Daniel S. Roche
United States Naval Academy
Annapolis, Maryland, U.S.A.
roche@usna.edu

ABSTRACT

We consider space-saving versions of several important operations on univariate polynomials, namely power series inversion and division, division with remainder, multi-point evaluation, and interpolation. Now-classical results show that such problems can be solved in (nearly) the same asymptotic time as fast polynomial multiplication. However, these reductions, even when applied to an in-place variant of fast polynomial multiplication, yield algorithms which require at least a linear amount of extra space for intermediate results. We demonstrate new in-place algorithms for the aforementioned polynomial computations which require only constant extra space and achieve the same asymptotic running time as their out-of-place counterparts. We also provide a precise complexity analysis so that all constants are made explicit, parameterized by the space usage of the underlying multiplication algorithms.

CCS CONCEPTS

• **Computing methodologies** → **Algebraic algorithms**; • **Theory of computation** → **Design and analysis of algorithms**.

KEYWORDS

polynomials, power series, algorithms, space complexity, division, inversion, multipoint evaluation, interpolation

ACM Reference Format:
Pascal Giorgi, Bruno Grenet, and Daniel S. Roche. 2020. Fast In-place Algorithms for Polynomial Operations: Division, Evaluation, Interpolation. In *International Symposium on Symbolic and Algebraic Computation (ISSAC '20), July 20–23, 2020, Kalamata, Greece.* ACM, New York, NY, USA, 8 pages. https://doi.org/10.1145/3373207.3404061

1 INTRODUCTION

Computations with dense univariate polynomials or truncated power series over a finite ring are of central importance in computer algebra and symbolic computation. Since the discovery of sub-quadratic ("fast") multiplication algorithms [3, 4, 10, 12, 19], a major research task was to reduce many other polynomial computations to the cost of polynomial multiplication.

This project has been largely successful, starting with symbolic Newton iteration for fast inversion and division with remainder [14], product tree algorithms for multi-point evaluation and interpolation [15], the "half-GCD" fast Euclidean algorithm [18], and many more related important problems [2, 6]. Not only are these problems important in their own right, but they also form the basis for many more, such as polynomial factorization, multivariate and/or sparse polynomial arithmetic, structured matrix computations, and further applications in areas such as coding theory and public-key cryptography.

But the use of fast arithmetic frequently comes at the expense of requiring extra *temporary space* to perform the computation. This can make a difference in practice, from the small scale where embedded systems engineers seek to minimize hardware circuitry, to the medium scale where a space-inefficient algorithm can exceed the boundaries of (some level of) cache and cause expensive cache misses, to the large scale where main memory may simply not be sufficient to hold the intermediate values. In a streaming model, where the output must be written only once, in order, explicit time-space tradeoffs prove that fast multiplication algorithms will always require up to linear extra space. And indeed, all sub-quadratic polynomial multiplication algorithms we are aware of — in their original formulation — require linear extra space [3, 4, 10, 12, 19].

However, if we treat the output space as pre-allocated random-access memory, allowing values in output registers to be both read and written multiple times, then improvements are possible. In-place quadratic-time algorithms for polynomial arithmetic are described in [16]. A series of recent results provide explicit algorithms and reductions from arbitrary fast multiplication routines which have the same *asymptotic* running time, but use only constant extra space [8, 11, 17]. That is, these algorithms trade a *constant* increase in the running time for a *linear* reduction in the amount of extra space. So far, these results are limited to multiplication routines and related computations such as middle and short product. Applying in-place multiplication algorithms directly to other problems, such as those considered in this paper, does not immediately yield an in-place algorithm for the desired application problem.

1.1 Our work

In this paper, we present new in-place algorithms for power series inversion and division, polynomial division with remainder, multi-point evaluation, and interpolation. These algorithms are *fast* because their running time is only a constant time larger than the fastest known out-of-place algorithms, parameterized by the cost of dense polynomial multiplication.

Our space complexity model is the one of [8, 11, 17] where input space is read only while output space is pre-allocated and can be used to store intermediate results. In that model, the space complexity is measured by only counting the auxiliary space required

ISSAC '20, July 20–23, 2020, Kalamata, Greece

ACM ISBN 978-1-4503-7100-1/20/07…$15.00
https://doi.org/10.1145/3373207.3404061

	Time	Space	Reference
Power series inversion	$(\lambda_m + \lambda_s)M(n)$	$\frac{1}{2}\max(c_m, c_s+1)n$	[9, Alg. MP-inv]
at precision n	$\lambda_m \mathsf{M}(n)\log_{\frac{c_m+2}{c_m+1}}(n)$	$O(1)$	Theorem 2.3
Power series division	$(\lambda_m + \frac{3}{2}\lambda_s)\mathsf{M}(n)$	$\frac{c_m+1}{2}n$	[9, Alg. MP-div-KM]
	$\lambda_m \mathsf{M}(n)\log_{\frac{c_m+3}{c_m+2}}(n)$	$O(1)$	Theorem 2.5
at precision n	$O(\mathsf{M}(n))$	αn, for any $\alpha > 0$	Remark 2.7
	$\left(\lambda_m(\frac{c+1}{2} + \frac{1}{c}) + \lambda_s(1+\frac{1}{c})\right)\mathsf{M}(n)$	$O(1)^{\ddagger}$	Corollary 2.6
Euclidean division	$(\lambda_m + \frac{3}{2}\lambda_s)\mathsf{M}(m) + \lambda_s\mathsf{M}(n)$	$\max(\frac{c_m+1}{2}m - n, c_s n)$	standard algorithm
of polynomials	$2\lambda_s\mathsf{M}(m) + (\lambda_m + \lambda_s)\mathsf{M}(n)$	$(1+\max(\frac{c_m}{2}, \frac{c_s+1}{2}, c_s))n$	$\lceil \frac{m}{n} \rceil$ balanced div. (precomp)
in sizes $(m+n-1, n)$	$\left(\lambda_m(\frac{c+1}{2} + \frac{1}{c}) + \lambda_s(2+\frac{1}{c})\right)\mathsf{M}(m)$	$O(1)$	Theorem 2.8
multipoint evaluation	$3/2\mathsf{M}(n)\log(n)$	$n\log(n)$	[2]
size-n polynomial on n points	$7/2\mathsf{M}(n)\log(n)$	n	[7], Lemma 3.1
	$(4 + 2\lambda_s/\log(\frac{c_s+3}{c_s+2}))\mathsf{M}(n)\log(n)$	$O(1)$	Theorem 3.4
interpolation	$5/2\mathsf{M}(n)\log(n)$	$n\log(n)$	[2]
size-n polynomial on n points	$5\mathsf{M}(n)\log(n)$	$2n$	[6, 7], Lemma 3.3
	$\simeq 105\mathsf{M}(n)\log(n)$	$O(1)$	Theorem 3.6

Table 1: Summary of complexity analyses, omitting non-dominant terms and assuming $c_f \le c_s \le c_m$. We use $c = c_m + 3$. For $O(1)^{\ddagger}$ space, the memory model is changed such that the input dividend can be overwritten. Here, and throughout the paper, the base of the logarithms is 2 if not otherwise stated.

during the computation, excluding input and output spaces. We shall mention that a single memory location or register may contain either an element of the coefficient ring, or a pointer to the input or output space. It follows that in-place algorithms are those that require only a constant number of extra memory locations.

For all five problems, we present in-place variants which have nearly the same asymptotic running time as their fastest out-of-place counterparts. The power series inversion and division algorithms incur an extra $\log(n)$ overhead when quasi-linear multiplication is used, while the polynomial division, evaluation, and interpolation algorithms keep the same asymptotic runtime as the fastest known algorithm. Our reductions essentially trade a small amount of extra runtime for a significant decrease in space usage.

Our motivation in this work is mainly theoretical. We address the existence of such fast in-place algorithms as we already did for polynomial multiplications [8]. To further extend our result, we compare precisely the number of arithmetic operations in our algorithms with the best known theoretical bounds. These results are summarized in Table 1.

Of course further work is needed to determine the practicability of our approach. In particular cache misses play a predominant role when dealing with memory management. Studying the cache complexity of all these algorithms, for instance in the idealized cache model [5], would give more precise insights. However, the practicability will heavily depend on the underlying multiplication algorithms. Due to their diversity and the need for fine-tuned implementations, we leave this task to future work.

1.2 Notation

By a size-n polynomial, we mean a polynomial of degree $\le n-1$. As usual, we denote by $\mathsf{M}(n)$ a bound on the number of operations in $\mathbb{K}$ to multiply two size-n polynomials, and we assume that $\alpha\mathsf{M}(n) \le \mathsf{M}(\alpha n)$ for any constant $\alpha \ge 1$. All known multiplication algorithms have at most a linear space complexity. Nevertheless, several results reduce this space complexity at the expense of a slight increase in the time complexity [8, 11, 17, 20]. To provide tight analyses, we consider multiplication algorithms with time complexity $\lambda_f\mathsf{M}(n)$ and space complexity $c_f n$ for some constants $\lambda_f \ge 1$ and $c_f \ge 0$.

Let us recall that the middle product of a size-$(m+n-1)$ polynomial $F \in \mathbb{K}[X]$ and a size-n polynomial $G \in \mathbb{K}[X]$ is the size-m polynomial defined as $\mathsf{MP}(F,G) = (FG \operatorname{div} X^{n-1}) \bmod X^m$. We denote by $\lambda_m\mathsf{M}(n)$ and $c_m n$ the time and space complexities of the middle product of size $(2n-1, n)$. Then, a middle product in size $(m+n-1, n)$ where $m < n$ can be computed with $\lceil \frac{n}{m} \rceil \lambda_m\mathsf{M}(m)$ operations in $\mathbb{K}$ and $(c_m+1)m$ extra space. Similarly, the short product of two size-n polynomials $F, G \in \mathbb{K}[X]$ is defined as $\mathsf{SP}(F,G) = FG \bmod X^n$ and we denote by $\lambda_s\mathsf{M}(n)$ and $c_s n$ its time and space complexities.

On the one hand, the most time-efficient algorithms achieve $\lambda_f = \lambda_m = \lambda_s = 1$ while $2 \le c_f, c_m, c_s \le 4$, using the *Transposition principle* [2, 9] for $\lambda_m = \lambda_f$. On the other hand, the authors recently proposed new space-efficient algorithms reaching $c_f = 0$, $c_m = 1$ and $c_s = 0$ while λ_f, λ_m and λ_s remain constants [8].

Writing $F = \sum_{i=0}^{d} f_i X^i \in \mathbb{K}[X]$, we will use $\mathrm{rev}(F) \in \mathbb{K}[X]$ to denote the reverse polynomial of F, that is, $\mathrm{rev}(F) = X^d F(1/X)$, whose computation does not involve any operations in $\mathbb{K}$. Note that we will use abusively the notation $F_{[a..b[}$ to refer to the chunk of F that is the polynomial $\sum_{i=a}^{b-1} f_i X^i$, and the notation $F_{[a]}$ for the coefficient f_a. Considering our storage, the notation $F_{[a..b[}$ will also serve to refer to some specific registers associated to F. When necessary, our algorithms indicate with WS the output registers used as work space.

2 INVERSION AND DIVISIONS

In this section, we present in-place algorithms for the inversion and the division of power series as well as the Euclidean division of polynomials. As a first step, we investigate the space complexity from the literature for these computations.

2.1 Space complexity of classical algorithms

Power series inversion. Power series inversion is usually computed through Newton iteration: If G is the inverse of F at precision k then $H = G+(1-GF)G \mod X^{2k}$ is the inverse of F at precision $2k$. This allows one to compute F^{-1} at precision n using $O(\mathsf{M}(n))$ operations in $\mathbb{K}$, see [6, Chapter 9]. As noticed in [9, Alg. `MP-inv`] only the coefficients of degree k to $2k-1$ of H are needed. Thus, assuming that $G_{[0..k[} = F^{-1} \mod X^k$, one step of Newton iteration computes k new coefficients of F^{-1} into $G_{[k..2k]}$ as

$$G_{[k..2k[} = -\mathsf{SP}(\mathsf{MP}(F_{[1..2k[}, G_{[0..k[}), G_{[0..k[}). \tag{1}$$

The time complexity is then $(\lambda_m + \lambda_s)\mathsf{M}(n)$ for an inversion at precision n. For space complexity, the most consuming part is the last iteration of size $\frac{n}{2}$. It needs $\max(c_m, c_s+1)\frac{n}{2}$ extra registers: One can compute the middle product in $G_{[\frac{n}{2}..n[}$ using $c_m\frac{n}{2}$ extra registers, then move it to $\frac{n}{2}$ extra registers and compute the short product using $c_s\frac{n}{2}$ registers.

Power series division. Let $F, G \in \mathbb{K}[[X]]$, the fast approach to compute $F/G \mod X^n$ is to first invert G at precision n and then to multiply the result by F. The complexity is given by one inversion and one short product at precision n. Actually, Karp and Markstein remarked in [13] that F/G can be directly computed during the last iteration. Applying this trick, the complexity becomes $(\lambda_m + \frac{3}{2}\lambda_s)\mathsf{M}(n)$ [9], see also [1]. The main difference with inversion is the storage of the short product of size $\frac{n}{2}$, yielding a space complexity of $\max(c_m+1, c_s+1)\frac{n}{2}$.

Euclidean division of polynomials. Given two polynomials A, B of respective size $m+n-1$ and n, the fast Euclidean division computes the quotient $A \operatorname{div} B$ as $\mathrm{rev}(\mathrm{rev}(A)/\mathrm{rev}(B))$ viewed as power series at precision m [6, Chapter 9]. The remainder R is retrieved with a size-n short product, yielding a total time complexity of $(\lambda_m + \frac{3}{2}\lambda_s)\mathsf{M}(m) + \lambda_s\mathsf{M}(n)$. Since the remainder size is not determined by the input size we assume that we are given a maximal output space of size $n-1$. As this space remains free when computing the quotient, this step requires $\frac{1}{2}\max(c_m+1, c_s+1)m - n + 1$ extra space, while computing the remainder needs $c_s n$.

As a first result, when $m \le n$, using space-efficient multiplication is enough to obtain an in-place $O(\mathsf{M}(n))$ Euclidean division. Indeed, the output space is enough to compute the *small* quotient, while the remainder can be computed in-place [8].

When $m > n$, the space complexity becomes $O(m-n)$. In that case, the Euclidean division of A by B can also be computed by $\lceil \frac{m}{n} \rceil$ *balanced* Euclidean divisions of polynomials of size $2n-1$ by B. It actually corresponds to a variation of the *long division algorithm*, in which each step computes n new coefficients of the quotient. To save some time, one can precompute the inverse of $\mathrm{rev}(B)$ at precision n, which gives a time complexity $(\lambda_m+\lambda_s)\mathsf{M}(n) + \frac{m}{n}2\lambda_s\mathsf{M}(n) \le 2\lambda_s\mathsf{M}(m) + (\lambda_m+\lambda_s)\mathsf{M}(n)$ and space complexity $(1+\max(\frac{c_m}{2}, \frac{c_s+1}{2}, c_s))n$.

Finally, one may consider to only compute the quotient or the remainder. Computing quotient only is equivalent to power series division. For the computation of the remainder, it is not yet known how to compute it without the quotient. In that case, we shall consider space usage for the computation and the storage of the quotient. When m is large compared to n, one may notice that relying on balanced divisions does not require one to retain the whole quotient, but only its n latest computed coefficients. In that case the space complexity only increases by n. Since we can always perform a middle product via two short products, we obtain the following result.

Lemma 2.1. *Given $A \in \mathbb{K}[X]$ of size m and $B \in \mathbb{K}[X]$, monic of size n, and provided n registers for the output, the remainder $A \bmod B$ can be computed using $2\lambda_s\mathsf{M}(m) + 3\lambda_s\mathsf{M}(n) + O(m+n)$ operations in $\mathbb{K}$ and $(c_s+2)n$ extra registers.*

2.2 In-place power series inversion

We notice that during the first Newton iterations, only a few coefficients of the inverse have been already written. The output space thus contains lots of free registers, and the standard algorithm can use them as working space. In the last iterations, the number of free registers becomes too small to perform a standard iteration. Our idea is then to *slow down* the computation. Instead of still doubling the number of coefficients computed at each iteration, the algorithm computes less and less coefficients, in order to be able to use the free output space as working space. We denote these two phases as acceleration and deceleration phases.

The following easy lemma generalizes Newton iteration to compute only $\ell \le k$ new coefficients from an inverse at precision k.

Lemma 2.2. *Let F be a power series and $G_{[0..k[}$ contain its inverse at precision k. Then for $0 < \ell \le k$, if we compute*

$$G_{[k..k+\ell[} = -\mathsf{SP}\left(\mathsf{MP}\left(F_{[1..k+\ell[}, G_{[0..k[}\right), G_{[0..\ell[}\right) \tag{2}$$

then $G_{[0..k+\ell[}$ contains the inverse of F at precision $k+\ell$.

Algorithm 1 is an in-place fast inversion algorithm. Accelerating and decelerating phases correspond to $\ell = k$ and $\ell < k$.

Algorithm 1 In-Place Fast Power Series Inversion (InPlaceInv)

Input: $F \in \mathbb{K}[X]$ of size n, such that $F_{[0]}$ is invertible;
Output: $G \in \mathbb{K}[X]$ of size n, such that $FG = 1 \mod X^n$.
Required: MP and SP alg. using extra space $\le c_m n$ and $\le c_s n$.

1: $G_{[0]} \leftarrow F_{[0]}^{-1}$
2: $k \leftarrow 1, \quad \ell \leftarrow 1$
3: **while** $\ell > 0$ **do**
4: $\quad G_{[n-\ell..n[} \leftarrow \mathsf{MP}(F_{[1..k+\ell[}, G_{[0..k[})$ ▷ WS: $G_{[k..n-\ell[}$
5: $\quad G_{[k..k+\ell[} \leftarrow \mathsf{SP}(G_{[0..\ell[}, -G_{[n-\ell..n[})$ ▷ WS: $G_{[k+\ell..n-\ell[}$
6: $\quad k \leftarrow k+\ell$
7: $\quad \ell \leftarrow \min\left(k, \left\lfloor \frac{n-k}{c} \right\rfloor\right)$ where $c = 2 + \max(c_m, c_s)$
8: $G_{[k..n[} \leftarrow \mathsf{SP}(G_{[0..n-k[}, -\mathsf{MP}(F_{[1..n[}, G_{[0..k[}))$ ▷ $O(1)$ space

Theorem 2.3. *Algorithm 1 is correct. It uses $O(1)$ space, and either $\lambda_m\mathsf{M}(n)\log_{\frac{c_m+2}{c_m+1}}(n) + O(\mathsf{M}(n))$ operations in $\mathbb{K}$ when $\mathsf{M}(n)$ is quasi-linear, or $O(\mathsf{M}(n))$ operations in $\mathbb{K}$ when $\mathsf{M}(n) = n^{1+\gamma}$, $0 < \gamma \le 1$.*

Proof. Steps 4 and 5, and Step 8, correspond to Equation (2). They compute ℓ new coefficients of G when k of them are already written in the output, whence Lemma 2.2 implies the correctness.

Step 4 needs $(c_m + 2)\ell$ free registers for its computation and its storage. Then $(c_s + 2)\ell$ free registers are needed to compute $\mathsf{SP}(G_{[0..\ell[}, G_{[n-\ell..n[})$ using ℓ registers for $G_{[n-\ell..n[}$ and $(c_s + 1)\ell$ registers for the short product computation and its result. For this computation to be done in-place, we need $c\ell \le n - k$. Since at most k new coefficients can be computed, the maximal number of new coefficients in each step is $\ell = \min\left(k, \left\lfloor \frac{n-k}{c} \right\rfloor\right)$.

Each iteration uses $O(\mathsf{M}(k))$ operations in $\mathbb{K}$: $O(\lceil k/\ell \rceil\, \mathsf{M}(\ell))$ for the middle product at Step 4 and $O(\mathsf{M}(\ell))$ for the short product at Step 5. The accelerating phase stops when $k > \frac{n-k}{c+1}$, that is, $k > \frac{n}{c+2}$. It costs $\sum_{i=0}^{\lfloor \log \frac{n}{c+2} \rfloor} \mathsf{M}(2^i) = O(\mathsf{M}(n))$. During the decelerating phase, each iteration computes a constant fraction of the remaining coefficients. Hence, this phase lasts for $\delta = \log_{\frac{c}{c-1}} n$ steps.

Let ℓ_i and k_i denote the values of ℓ and k at the i-th iteration of the deceleration phase and $t_i = n - k_i$. Then one iteration of the deceleration phase costs one middle product in sizes $(n - t_i + \lfloor \frac{t_i}{c} \rfloor - 1, n - t_i)$ and one short product in size $\lfloor \frac{t_i}{c} \rfloor$. The total cost of all the short products amounts to $\sum_i \mathsf{M}(t_i) = O(\mathsf{M}(n))$ since $\sum_i t_i \le cn$. The cost of the middle product at the i-th step is

$$\lambda_m \left\lceil (n - t_i)/\left\lfloor \tfrac{t_i}{c} \right\rfloor \right\rceil \mathsf{M}\left(\left\lfloor \tfrac{t_i}{c} \right\rfloor\right) = \lambda_m \mathsf{M}(n) + O(n).$$

Therefore, the total cost of all the middle products is at most $\lambda_m \mathsf{M}(n) \log_{\frac{c}{c-1}}(n) + O(\mathsf{M}(n))$ and is dominant in the complexity. We can choose the in-place short products of [8] and get $c = c_m + 2$. The complexity is then $\lambda_m \mathsf{M}(n) \log_{\frac{c_m+2}{c_m+1}}(n) + O(\mathsf{M}(n))$.

If $\mathsf{M}(n) = n^{1+\gamma}$ with $0 < \gamma \le 1$, the cost of each iteration is $O\left(\left\lceil \frac{n-t_i}{\ell_i} \right\rceil \ell_i^{1+\gamma}\right)$. Since $\ell_0 \le n$, we have $\ell_i < n(\frac{c-1}{c})^i + c$, whence

$$\sum_{i=1}^{\delta} \left\lceil \frac{n - t_i}{\ell_i} \right\rceil \ell_i^{1+\gamma} \le n \sum_{i=1}^{\delta} \ell_i^{\gamma} \le n \sum_{i=1}^{\delta} \left(n \left(\frac{c-1}{c} \right)^i + c \right)^{\gamma}.$$

Since $0 < \gamma \le 1$, we have $(\alpha + \beta)^\gamma \le \alpha^\gamma + \beta^\gamma$ for any $\alpha, \beta > 0$, and the complexity is $n^{1+\gamma} \sum_{i=1}^{\delta} \left(\frac{c-1}{c}\right)^{i\gamma} + O(n \log n) = O(\mathsf{M}(n))$. □

2.3 In-place division of power series

Division of power series can be implemented easily as an inversion followed by a product. Yet, using in-place algorithms for these two steps is not enough to obtain an in-place division algorithm since the intermediate result must be stored. Karp and Markstein's trick, that includes the dividend in the last iteration of Newton iteration [13], cannot be used directly in our case since we replace the very last iteration by several ones. We thus need to build our in-place algorithm on the following generalization of their method.

Lemma 2.4. *Let F and G be two power series, G invertible, and $Q_{[0..k[}$ contain their quotient at precision k. Then for $0 < \ell \le k$, if we compute*

$$Q_{[k..k+\ell[} = \mathsf{SP}\left(G^{-1}_{[0..\ell[}, F_{[k..k+\ell[} - \mathsf{MP}(G_{[1..k+\ell[}, Q_{[0..k[})\right)$$

then $Q_{[0..k+\ell[}$ contains their quotient at precision $k + \ell$.

Proof. Let us write $F/G = Q_k + X^k Q_\ell + O(X^{k+\ell})$. We prove that $Q_\ell = G^{-1} \times ((F - GQ_k) \operatorname{div} X^k) \bmod X^\ell$. By definition, $F \equiv G(Q_k + X^k Q_\ell) \bmod X^{k+\ell}$. Hence $(F - GQ_k) \operatorname{div} X^k = GQ_\ell \bmod X^\ell$. Therefore, $Q_\ell = (G^{-1} \times ((F - GQ_k) \operatorname{div} X^k)) \bmod X^\ell$. Finally, since only the coefficients of degree k to $k + \ell - 1$ of GQ_k are needed, they can be computed as $\mathsf{MP}(G_{[1..k+\ell[}, Q_{[0..k[})$. □

Algorithm 2 is an in-place power series division algorithm based on Lemma 2.4, choosing at each step the appropriate value of ℓ so that all computations can be performed in place.

Algorithm 2 In-Place Power Series Division (InPlacePSDiv)

Input: $F, G \in \mathbb{K}[X]$ of size n, such that $G_{[0]}$ is invertible;
Output: $Q \in \mathbb{K}[X]$ of size n, such that $F/G = Q \mod X^n$.
Required: MP, SP, Inv alg. using extra space $\le c_m n,\ c_s n,\ c_i n$.

1: $k \leftarrow \lfloor n/\max(c_i + 1, c_s + 2) \rfloor$
2: $Q_{[n-k..n[} \leftarrow \mathrm{rev}(\mathsf{Inv}(G_{[0..k[}))$ ▷ WS: $Q_{[0..n-k[}$
3: $Q_{[0..k[} \leftarrow \mathsf{SP}(F_{[0..k[}, \mathrm{rev}(Q_{[n-k..n[}))$ ▷ WS: $Q_{[k..n-k[}$
4: $\ell \leftarrow \lfloor (n - k)/(3 + \max(c_m, c_s)) \rfloor$
5: **while** $\ell > 0$ **do**
6: $\quad Q_{[n-2\ell..n-\ell[} \leftarrow \mathsf{MP}(G_{[1..k+\ell[}, Q_{[0..k[})$ ▷ WS: $Q_{[k..n-2\ell[}$
7: $\quad Q_{[n-2\ell..n-\ell[} \leftarrow F_{[k..k+\ell[} - Q_{[n-2\ell..n-\ell[}$
8: $\quad$ *let us define* $Q^*_\ell = \mathrm{rev}(Q_{[n-\ell..n[})$
$\qquad Q_{[k..k+\ell[} \leftarrow \mathsf{SP}(Q_{[n-2\ell..n-\ell[}, Q^*_\ell)$ ▷ WS: $Q_{[k+\ell..n-2\ell[}$
9: $\quad k \leftarrow k + \ell$
10: $\quad \ell \leftarrow \lfloor (n - k)/(3 + \max(c_m, c_s)) \rfloor$
11: $tmp \leftarrow F_{[k..n[} - \mathsf{MP}(G_{[1..n[}, Q_{[0..k[})$ ▷ constant space
12: $Q_{[k..n[} \leftarrow \mathsf{SP}(tmp, \mathrm{rev}(Q_{[k..n[}))$ ▷ constant space

Theorem 2.5. *Algorithm 2 is correct. It uses $O(1)$ space, and either $\lambda_m \mathsf{M}(n) \log_{\frac{c_m+3}{c_m+2}}(n) + O(\mathsf{M}(n))$ operations in $\mathbb{K}$ when $\mathsf{M}(n)$ is quasi-linear or $O(\mathsf{M}(n))$ operations in $\mathbb{K}$ when $\mathsf{M}(n) = O(n^{1+\gamma})$, $0 < \gamma \le 1$.*

Proof. The correctness follows from Lemma 2.4. The inverse of G is computed once at Step 2, at precision $\lfloor n/\max(c_i + 1, c_s + 2) \rfloor$. Its coefficients are then progressively overwritten during the loop since Step 8 only requires ℓ coefficients of the inverse, and ℓ is decreasing. Since $c_i = \frac{1}{2}\max(c_m, c_s + 1)$, ℓ is always less than the initial precision. For simplicity of the presentation, we store the inverse in reversed order in $Q_{[n-k..n[}$. Step 2 requires space $c_i k$ while the free space has size $n - k$: Since $k \le \frac{n}{c_i+1}$, the free space is large enough. Similarly, the next step requires space $c_s k$ while the free space has size $n - 2k$, and $k \le \frac{n}{c_s+2}$. Step 6 needs $(c_m + 1)\ell$ space and the free space has size $n - k - 2\ell$, and Step 8 requires $c_s \ell$ space while the free space has size $n - k - 3\ell$. Since $\ell \le \frac{n-k}{3+\max(c_m, c_s)}$, these computations can also be performed in place.

The time complexity analysis is very similar to the one of Algorithm 1 given in Theorem 2.3. The main difference is Step 7 which adds a negligible term $O(n \log n)$ in the complexity. □

Corollary 2.6. *If it can erase its dividend, Algorithm 2 can be modified to improve its complexity to $\left(\lambda_m(\frac{c+1}{2} + \frac{1}{c}) + \lambda_s(1 + \frac{1}{c})\right)\mathsf{M}(n) + O(n)$ operations in $\mathbb{K}$ where $c = \max(c_m + 3, c_s + 2)$, still using $O(1)$ extra space.*

Proof. Once k coefficients of Q have been computed, $F_{[0..k[}$ is not needed anymore. This means that at Step 7, the result can be directly written in $F_{[k..k+\ell[}$ and that $F_{[0..k[}$ can be used as working space in the other steps of the loop. The free space at Steps 6 and 8

becomes $n-2\ell$ instead of $n-k-2\ell$ and $n-k-3\ell$ respectively. Therefore, ℓ can always be chosen as large as $\lfloor \frac{n}{c} \rfloor$ where $c = \max(c_m + 3, c_s + 2)$. Since ℓ stays positive, we also modify the algorithm to stop when all the coefficients of Q have been computed.

To simplify the complexity analysis, we further assume that k gets the same value $\lfloor \frac{n}{c} \rfloor$ at Step 1. Step 2 requires $(\lambda_s + \lambda_m)\mathsf{M}(\lfloor \frac{n}{c} \rfloor)$ operations in $\mathbb{K}$. The sum of the input sizes of all the short products in the algorithm is n. Their total complexity is thus $\lambda_s \mathsf{M}(n)$. At the i-th iteration of the loop, $k = (i+1)\ell$. Therefore Step 6 has complexity $i\lfloor \frac{n}{c} \rfloor$. Step 7 requires $\lfloor \frac{n}{c} \rfloor$ operations in $\mathbb{K}$. Altogether, the complexity of the modified algorithm is

$$\lambda_s \mathsf{M}(n) + (\lambda_s + \lambda_m)\mathsf{M}\left(\lfloor \tfrac{n}{c} \rfloor\right) + \sum_{i=1}^{c} \left(i\lambda_m \mathsf{M}\left(\lfloor \tfrac{n}{c} \rfloor\right) + \lfloor \tfrac{n}{c} \rfloor\right)$$

which is $\left(\lambda_m(\frac{c+1}{2} + \frac{1}{c}) + \lambda_s(1 + \frac{1}{c})\right)\mathsf{M}(n) + O(n)$. □

Using similar techniques, we get the following variant.

Remark 2.7. *Algorithm 2 can be easily modified to improve the complexity to $O(M(n))$ operations in $\mathbb{K}$ when a linear amount of extra space is available, say αn registers for some $\alpha \in \mathbb{R}_+$.*

2.4 In-place Euclidean division of polynomials

If A is a size-$(m+n-1)$ polynomial and B a size-n polynomial, one can compute their size-m quotient Q in place using Algorithm 2, in $O((\mathsf{M}(m)\log m))$ operations in $\mathbb{K}$. When Q is known, the remainder $R = A - BQ$, can be computed in-place using $O(\mathsf{M}(n))$ operations in $\mathbb{K}$ as it requires a single short product and some subtractions. As already mentioned, the exact size of the remainder is not determined by the size of the inputs. Given any tighter bound $r < n$ on $\deg(R)$, the same algorithm can compute R in place, in time $O(\mathsf{M}(r))$.

Altogether, we get in-place algorithms to compute the quotient of two polynomials in time $O(\mathsf{M}(m)\log m)$, or the quotient and size-r remainder in time $O(\mathsf{M}(m)\log m + \mathsf{M}(r))$. As suggested in Section 2.1 and in Remark 2.7, this complexity becomes $O(\mathsf{M}(m) + \mathsf{M}(r))$ whenever $m = O(r)$. Indeed, in that case the remainder space can be used to speed-up the quotient computation. We shall mention that computing only the remainder remains a harder problem as we cannot count on the space of the quotient while it is required for the computation. As of today, only the classical quadratic long division algorithm allows such an in-place computation.

We now provide a new in-place algorithm for computing both quotient and remainder that achieves a complexity of $O(\mathsf{M}(m) + \mathsf{M}(n))$ operation in $\mathbb{K}$ when $m \geq n$. Our algorithm requires an output space of size $n-1$ for the remainder since taking any smaller size $r < n-1$ would rebind to power series division.

Theorem 2.8. *Algorithm 3 is correct. It uses $O(1)$ extra space and $\left(\lambda_m(\frac{c+1}{2} + \frac{1}{c}) + \lambda_s(2 + \frac{1}{c})\right)\mathsf{M}(m) + O(m\log n)$ operations in $\mathbb{K}$ where $c = \max(c_m + 3, c_s + 2)$.*

Proof. Algorithm 3 is an adaptation of the classical *long division algorithm*, recalled in Section 2.1, where chunks of the quotient are computed iteratively *via* Euclidean division of size $(2n-1, n)$. The main difficulty is that the update of the dividend cannot be done on the input. Since we compute only chunks of size n from the quotient, the update of the dividend affects only $n-1$ coefficients. Therefore, it is possible to use the space of R for storing these new coefficients. As we need to consider n coefficients from the dividend to get a new chunk, we add the missing coefficient from A and consider the polynomial H as our new dividend.

Algorithm 3 In-Place Euclidean Division (InPlaceEuclDiv)

Input: $A, B \in \mathbb{K}[X]$ of sizes $(m+n, n)$, $m \geq n$, such that $B_{[0]} \neq 0$;
Output: $Q, R \in \mathbb{K}[X]$ of sizes $(m+1, n-1)$ such that $A = BQ + R$;
Required: In-place DivErase(F, G, n) computing $F/G \bmod X^n$ while erasing F; In-place SP;
For simplicity, H is a size-n polynomial such that $H_{[0..n-1[}$ is R and $H_{[n-1]}$ is an extra register

1: $H \leftarrow A_{[m..m+n[}$
2: $k \leftarrow m+1$
3: **while** $k > n$ **do**
4: $\quad Q_{[k-n..k[} \leftarrow \mathrm{rev}(\textsc{DivErase}(\mathrm{rev}(H), \mathrm{rev}(B), n))$
5: $\quad H_{[0..n-1[} \leftarrow \mathsf{SP}(Q_{[k-n..k-1[}, B_{[0..n-1[})$
6: $\quad H_{[1..n[} \leftarrow A_{[k-n..k-1[} - H_{[0..n-1[}$
7: $\quad H_{[0]} \leftarrow A_{[k-n-1]}$
8: $\quad k \leftarrow k-n$
9: $Q_{[0..k[} \leftarrow \mathrm{rev}(\textsc{DivErase}(\mathrm{rev}(H_{[n-k..n[}), \mathrm{rev}(B_{[n-k..n[})))$
10: $H_{[0..n-1[} \leftarrow \mathsf{SP}(Q_{[0..n-1[}, B_{[0..n-1[})$
11: $H_{[0..n-1[} \leftarrow A_{[0..n-1[} - H_{[0..n-1[}$
12: **return** $(Q, H_{[0..n-1[})$

By Corollary 2.6, Step 4 can be done in place while erasing H, which is not part of the original input. It is thus immediate that our algorithm is in-place. For the complexity, Steps 4 and 5 dominate the cost. Using the exact complexity for Step 4 given in Corollary 2.6, one can deduce easily that Algorithm 3 requires $\left(\lambda_m(\frac{c+1}{2} + \frac{1}{c}) + \lambda_s(2 + \frac{1}{c})\right)\mathsf{M}(m) + O(m\log n)$ operations in $\mathbb{K}$. □

Using time-efficient products with $\lambda_m = \lambda_s = 1$, $c_m = 4$ and $c_s = 3$ yields a complexity $\simeq 6.29\mathsf{M}(m)$, which is roughly $6.29/4 = 1.57$ times slower than the most time-efficient out-of-place algorithm.

3 MULTIPOINT EVALUATION AND INTERPOLATION

In this section, we present in-place algorithms for the two related problems of multipoint evaluation and interpolation. We first review both classical algorithms and their space-efficient variants.

3.1 Space complexity of classical algorithms

Multipoint evaluation. Given n elements $a_1, \ldots, a_n$ of $\mathbb{K}$ and a size-n polynomial $F \in \mathbb{K}[X]$, multipoint evaluation aims to compute $F(a_1), \ldots, F(a_n)$. While the naive approach using Horner scheme leads to a quadratic complexity, the fast approach of [15] reaches a quasi-linear complexity $O(\mathsf{M}(n)\log(n))$ using a divide-and-conquer approach and the fact that $F(a_i) = F \bmod (X - a_i)$. As proposed in [2] this complexity can be sharpened to $(\lambda_m + \frac{1}{2}\lambda_f)\mathsf{M}(n)\log(n) + O(\mathsf{M}(n))$ using the transposition principle.

The fast algorithms are based on building the so-called *subproduct tree* [6, Chapter 10] whose leaves contain the $(X - a_i)$'s and whose root contains the polynomial $\prod_{i=1}^{n}(X - a_i)$. This tree contains 2^i degree-$n/2^i$ monic polynomials at level i, and can be stored

in exactly $n \log n$ registers if n is a power of two. The fast algorithms then require $n\log(n) + O(n)$ registers as work space. Here, because the space complexity constants c_f, c_m, c_s do not appear in the leading term $n\log(n)$ of space usage, we can always choose the fastest underlying multiplication routines, so the computational cost for this approach is simply $\frac{3}{2}\mathsf{M}(n)\log(n) + O(\mathsf{M}(n))$.

As remarked in [7], one can easily derive a fast variant that uses only $O(n)$ extra space. In particular, [7, Lemma 2.1] shows that the evaluation of a size-n polynomial F on k points $a_1, \ldots, a_k$ with $k \leq n$ can be done at a cost $O(\mathsf{M}(k)(\frac{n}{k} + \log(k)))$ with $O(k)$ extra space.

We provide a tight analysis of this algorithm, starting with the *balanced case* $k = n$, i.e. the number of evaluation points is equal to the size of F. The idea of the algorithm is to group the points in $\lceil \log(n) \rceil$ groups of $\lfloor n/\log(n) \rfloor$ points each, and to use standard multipoint evaluation on each group, by first reducing F modulo the root of the corresponding subproduct tree. The complexity analysis of this approach is given in the following lemma. Observe that here too, the constants λ_s, c_s, etc., do not enter in since we can always use the fastest out-of-place subroutines without affecting the $O(n)$ term in the space usage.

Lemma 3.1. *Given $F \in \mathbb{K}[X]$ of size n and $a_1, \ldots, a_n \in \mathbb{K}$, one can compute $F(a_1), \ldots, F(a_n)$ using $\frac{7}{2}\mathsf{M}(n)\log(n) + O(\mathsf{M}(n))$ operations in $\mathbb{K}$ and $n + O(\frac{n}{\log(n)})$ extra registers.*

Proof. Computing each subproduct tree on $O(n/\log(n))$ points can be done in time $\frac{1}{2}\mathsf{M}(n/\log(n))\log(n) \leq \frac{1}{2}\mathsf{M}(n)$ and space $n + O(n/\log(n))$. The root of this tree is a polynomial of degree at most $n/\log(n)$. Each reduction of F modulo such a polynomial takes time $2\mathsf{M}(n) + O(n/\log(n))$ and space $O(n/\log(n))$ using the balanced Euclidean division algorithm from Section 2.1. Each multipoint evaluation of the reduced polynomial on $n/\log(n)$ points, using the pre-computed subproduct tree, takes $\mathsf{M}(n/\log(n))\log(n) + O(\mathsf{M}(n/\log(n)))$ operations in $\mathbb{K}$ and $O(n/\log(n))$ extra space [2].

All information except the evaluations from the last step — which are written directly to the output space — may be discarded before the next iteration begins. Therefore the total time and space complexity are as stated. □

When the number of evaluation points k is large compared to the size n of the polynomial F, we can simply repeat the approach of Lemma 3.1 $\lceil k/n \rceil$ times. The situation is more complicated when $k \leq n$, because the output space is smaller. The idea is to compute the degree-k polynomial M at the root of the product tree, reduce F modulo M and perform balanced k-point evaluation of F mod M.

Lemma 3.2. *Given $F \in \mathbb{K}[X]$ of size n and $a_1, \ldots, a_k \in \mathbb{K}$, one can compute $F(a_1), \ldots, F(a_k)$ using $2\lambda_s\mathsf{M}(n) + 4\mathsf{M}(k)\log(k) + O(n + \mathsf{M}(k)\,\mathrm{loglog}(k))$ operations in $\mathbb{K}$ and $(c_s + 2)k + O(k/\log(k))$ extra registers.*

Proof. Computing the root M of a product tree proceeds in two phases. For the bottom levels of the tree, we use the fastest out-of-place full multiplication algorithm that computes the product of two size-t polynomials in time $\mathsf{M}(t)$ and space $O(t)$. Then, only for the top $\mathrm{loglog}(n)$ levels, do we switch to an in-place full product algorithm from [8], which has time $O(\mathsf{M}(t))$ but only $O(1)$ extra space. The result is that M can be computed using $\frac{1}{2}\mathsf{M}(k)\log(k) + O(\mathsf{M}(k)\,\mathrm{loglog}(k))$ operations in $\mathbb{K}$ and $k + O(k/\log(k))$ registers.

Then, we reduce F modulo M. By Lemma 2.1, this is accomplished in time $2\lambda_s\mathsf{M}(n) + O(n + \mathsf{M}(k))$ and space $(c_s + 2)k$. Adding the cost of the k-point evaluation of Lemma 3.1 completes the proof. □

Interpolation. Interpolation is the inverse operation of multipoint evaluation, that is, to reconstruct a size-n polynomial F from its evaluations on n distinct points $F(a_1), \ldots, F(a_n)$. The classic approach using Lagrange's interpolation formula has a quadratic complexity [6, Chapter 5] while the fast approach of [15] has quasi-linear time complexity $O(\mathsf{M}(n)\log(n))$. We first briefly recall this fast algorithm.

Let $M(X) = \prod_{i=1}^{n}(X - a_i)$ and M' its derivative. Noting that $\frac{M}{X-a_i}(a_i) = M'(a_i)$ for $1 \leq i \leq n$, we have

$$F(X) = M(X) \sum_{i=1}^{n} \frac{F(a_i)/M'(a_i)}{X - a_i}. \tag{3}$$

Hence the fast algorithm of [15] consists in computing $M'(X)$ and its evaluation on each a_i through multipoint evaluation, and then to sum the n fractions using a divide-and-conquer strategy. The numerator of the result is then F by Equation (3).

If the subproduct tree over the a_i's is already computed, this gives all the denominators in the rational fraction sum. Using the same subproduct tree for evaluating M' and for the rational fraction sum gives the fastest interpolation algorithm, combining the textbook method [6] with the multi-point evaluation of [2]. The total computational cost is only $\frac{5}{2}\mathsf{M}(n)\log(n) + O(\mathsf{M}(n))$, while the space is dominated by the size of this subproduct tree, $n\log(n) + O(n)$.

A more space-efficient approach can be derived using linear-space multipoint evaluation. Since the subproduct must be essentially recomputed on the first and last steps, the total running time is $(2\lambda_f + \frac{7}{2})\mathsf{M}(n)\log(n) + O(\mathsf{M}(n))$, using $(2 + \frac{1}{2}c_f)n + O(n/\log(n))$ registers. This approach can be improved in two ways: first by again grouping the interpolation points and re-using the smaller subproduct trees for each group, and secondly by using an in-place full multiplication algorithm from [8] to combine the results of each group in the rational function summation. A detailed description of the resulting algorithm, along with a proof of the following lemma, can be found in the preprint version of this paper*.

Lemma 3.3. *Given $a_1, \ldots, a_n \in \mathbb{K}$ and $y_1, \ldots, y_n \in \mathbb{K}$, one can compute $F \in \mathbb{K}[X]$ of size n such that $F(a_i) = y_i$ for $1 \leq i \leq n$ using $5\mathsf{M}(n)\log(n) + O(\mathsf{M}(n)\,\mathrm{loglog}(n))$ operations in $\mathbb{K}$ and $2n + O(n/\log(n))$ extra registers.*

3.2 In-place multipoint evaluation

In order to derive an in-place algorithm we make repeated use of the unbalanced multi-point evaluation with linear space to compute only k evaluations of the polynomial F among the n original points. The strategy is to set k as a fraction of n to ensure that $n - k$ is large enough to serve as extra space. Applying this strategy on smaller and smaller values of k leads to Algorithm 4, which is an in-place algorithm with the same asymptotic time complexity $O(\mathsf{M}(n)\log(n))$ as out-of-place fast multipoint evaluation.

*Available under reference arXiv:2002.10304.

Algorithm 4 In-Place Multipoint Evaluation (InPlaceEval)

Input: $F \in \mathbb{K}[X]$ of size n and $(a_1, \ldots, a_n) \in \mathbb{K}^n$;
Output: $R = (F(a_1), \ldots, F(a_n))$
Required: Eval of space complexity $\leq (c_s + 2)k$ as in Lemma 3.2

1: $s \leftarrow 0, \quad k \leftarrow \lfloor n/(c_s + 3) \rfloor$
2: **while** $k > 0$ **do**
3: $\quad R_{[s..s+k[} \leftarrow \text{Eval}(F, a_s, \ldots, a_{s+k})$ ▷ WS: $R_{[s+k..n[}$
4: $\quad s \leftarrow s + k$
5: $\quad k \leftarrow \left\lfloor \frac{n-s}{c_s+3} \right\rfloor$
6: $R_{[s..n[} \leftarrow \text{Eval}(F, a_s, \ldots, a_n)$ ▷ constant space

Theorem 3.4. *Algorithm 4 is correct. It uses $O(1)$ extra space and $\left(4 + 2\lambda_s/\log(\frac{c_s+3}{c_s+2})\right)\mathsf{M}(n)\log(n) + O(\mathsf{M}(n)\log\log n)$ operations in $\mathbb{K}$.*

Proof. The correctness is obvious as soon as Eval is correct. By the choice of k and from the extra space bound of Eval from Lemma 3.2, Step 3 has sufficient work space, and therefore the entire algorithm is in-place. The sequence $k_i = \frac{(c_s+2)^{i-1}}{(c_s+3)^i} n$, for $i = 1, 2, \ldots$, gives the values of k in each iteration. Then $\sum_i k_i \leq n$ and the loop terminates after at most $\ell \log(n)$ iterations, where $\ell \leq 1/\log(\frac{c_s+3}{c_s+2})$. Applying Lemma 3.2, the cost of the entire algorithm is therefore dominated by $\sum_{1 \leq i \leq \ell} (2\lambda_s \mathsf{M}(n) + 4\mathsf{M}(k_i)\log(k_i))$, which is at most $(2\lambda_s \ell + 4)\mathsf{M}(n)\log(n)$. □

Using a time-efficient short product with $\lambda_s = 1$ and $c_s = 3$ yields a complexity $\simeq 11.61\mathsf{M}(n)\log n$, which is roughly $11.61/1.5 = 7.74$ times slower than the most time-efficient out-of-place algorithm.

3.3 In-place interpolation

Let $(a_1, y_1), \ldots, (a_n, y_n)$ be n pairs of evaluations, with the a_i's pairwise distinct. Our goal is to compute the unique size-n polynomial $F \in \mathbb{K}[X]$ such that $F(a_i) = y_i$ for $1 \leq i \leq n$, with an in-place algorithm. Our first aim is to provide a variant of polynomial interpolation that computes $F \bmod X^k$ using $O(k)$ extra space. Without loss of generality, we assume that k divides n. For $i = 1$ to n/k, let $T_i = \prod_{j=1+k(i-1)}^{ki}(X - a_j)$ and $S_i = M/T_i$ where $M = \prod_{i=1}^{n}(X - a_i)$. Note that $S_i = \prod_{j \neq i} T_j$. One can rewrite Equation (3) as

$$\begin{aligned} F(X) &= M(X) \sum_{i=1}^{n/k} \sum_{j=1+k(i-1)}^{ki} \frac{F(a_j)}{M'(a_j)} \frac{1}{(X - a_j)} \\ &= M(X) \sum_{i=1}^{n/k} \frac{N_i(X)}{T_i(X)} = \sum_{i=1}^{n/k} N_i(X) S_i(X) \end{aligned} \tag{4}$$

for some size-k polynomials $N_1, \ldots, N_{n/k}$. One may remark that the latter equality can also be viewed as an instance of the chinese remainder theorem where $N_i = F/S_i \bmod T_i$ (see [6, Chapter 5]). To get the first k terms of the polynomial F, we only need to compute

$$F \bmod X^k = \sum_{i=1}^{n/k} N_i (S_i \bmod X^k) \bmod X^k. \tag{5}$$

One can observe that $M'(a_j) = (S_i \bmod T_i)(a_j) T_i'(a_j)$ for $k(i-1) < j \leq ki$. Therefore, Equation (4) implies that N_i is the unique size-k polynomial satisfying $N_i(a_j) = (F/S_i \bmod T_i)(a_j)$ and can be computed using interpolation. One first computes $S_i \bmod T_i$, evaluates it at the a_j's, performs k divisions in $\mathbb{K}$ to get each $N_i(a_j)$ and finally interpolates N_i.

Our second aim is to generalize the previous approach when some initial coefficients of F are known. Writing $F = G + X^s H$ where G is known, we want to compute $H \bmod X^k$ from some evaluations of F. Since H has size at most $(n - s)$, only $(n - s)$ evaluation points are needed. Therefore, using Equation (4) with $M = \prod_{i=1}^{n-s}(X - a_i)$, we can write

$$H(X) = M(X) \sum_{i=1}^{(n-s)/k} \sum_{j=1+k(i-1)}^{ki} \frac{F(a_j) - G(a_j)}{a_j^s M'(a_j)} \frac{1}{(X - a_j)}. \tag{6}$$

This implies that $H \bmod X^k$ can be computed using the same approach described above by replacing $F(a_j)$ with $H(a_j) = (F(a_j) - G(a_j))/a_j^s$. We shall remark that the $H(a_j)$'s can be computed using multipoint evaluation and fast exponentation. Algorithm 5 fully describes this approach.

Algorithm 5 Partial Interpolation (PartInterpol)

Input: $G \in \mathbb{K}[X]$ of size s and $(y_1, \ldots, y_{n-s})$, $(a_1, \ldots, a_{n-s})$ in $\mathbb{K}^{n-s}$; an integer $k \leq n - s$
Output: $H \bmod X^k$ where $F = G + X^s H \in \mathbb{K}[X]$ is the unique size-n polynomial s.t. $F(a_i) = y_i$ for $1 \leq i \leq n - s$

1: **for** $i = 1$ to $(n - s)/k$ **do**
2: $\quad S_i^k \leftarrow 1, S_i^T \leftarrow 1$
3: $\quad T_i \leftarrow \prod_{j=1+k(i-1)}^{ki}(X - a_j)$ ▷ Fast divide-and-conquer
4: $\quad$ **for** $j = 1$ to $(n - s)/k, j \neq i$ **do**
5: $\quad\quad T_j \leftarrow \prod_{t=1+k(j-1)}^{kj}(X - a_t)$ ▷ Fast divide-and-conquer
6: $\quad\quad S_i^k \leftarrow S_i^k \times T_j \mod X^k$ ▷ $S_i^k = S_i \bmod X^k$
7: $\quad\quad S_i^T \leftarrow S_i^T \times T_j \mod T_i$ ▷ $S_i^T = S_i \bmod T_i$
8: $\quad G^T \leftarrow G \bmod T_i$
9: $\quad (b_1, \ldots, b_k) \leftarrow \text{Eval}(S_i^T, a_{1+k(i-1)}, \ldots, a_{ki})$
$\quad (z_1, \ldots, z_k) \leftarrow \text{Eval}(G^T, a_{1+k(i-1)}, \ldots, a_{ki})$
10: $\quad$ **for** $j = 1$ to k **do**
11: $\quad\quad b_j \leftarrow (y_{j+k(i-1)} - z_j)/(a^s_{j+k(i-1)} b_j)$
12: $\quad N_i \leftarrow \text{Interpol}((z_1, \ldots, z_k), (b_1, \ldots, b_k))$
13: $\quad H_{[0..k[} \leftarrow H_{[0..k[} + N_i S_i^k \bmod X^k$

Lemma 3.5. *Algorithm 5 is correct. It requires $6k + O(k/\log k)$ extra space and it uses $\left(\frac{1}{2}(\frac{n-s}{k})^2 + \frac{23}{2}\frac{n-s}{k}\right)\mathsf{M}(k)\log(k) + (n-s)\log(s) + O((\frac{n-s}{k})^2 \mathsf{M}(k)\log\log k)$ operations in $\mathbb{K}$.*

Proof. The correctness follows from the above discussion. In particular, note that the polynomials S_i^k and S_i^T at Steps 6 and 7 equal $S_i \bmod X^k$ and $S_i \bmod T_i$ respectively. Furthermore, $z_j = G(a_{j+k(i-1)})$ since $G(a_{j+k(i-1)}) = (G \bmod T_i)(a_{j+k(i-1)})$. Hence, Step 12 correctly computes the polynomial N_i and the result follows from Equations (5) and (6).

From the discussion in Section 3.1, we can compute each T_i in $1/2\mathsf{M}(k)\log(k) + O(M(k)\log\log k)$ operations in $\mathbb{K}$ and k extra space. Step 9 requires some care as we can share some computation among the two *equal-size* evaluations. Indeed, the subproduct trees induced

by this computation are identical and thus can be computed only once. Using Lemma 3.1, this amounts to $\frac{13}{2}\mathsf{M}(k)\log(k) + O(\mathsf{M}(k))$ operations in $\mathbb{K}$ using $k + O(k/\log k)$ extra space. Step 12 can be done in $5\mathsf{M}(k)\log(k) + O(M(k)\operatorname{loglog} k)$ operations in $\mathbb{K}$ and $2k + O(k/\log k)$ extra space using Lemma 3.3. Taking into account the $n - s$ exponentations a_j^s, and that other steps have a complexity in $O(\mathsf{M}(k))$, the cost of the algorithm is

$$\left(\frac{1}{2}\left(\frac{n-s}{k}\right)^2 + \frac{23}{2}\frac{n-s}{k}\right)\mathsf{M}(k)\log(k) + (n-s)\log(s) + O\left(\left(\frac{n-s}{k}\right)^2 \mathsf{M}(k)\operatorname{loglog} k\right).$$

We show that $6k + O(k/\log k)$ extra registers are enough to implement this algorithm. At Step 7, the polynomials T_i, T_j, S_i^k, S_i^T must be stored in memory. The computation involved at this step requires only $2k$ extra registers as $S_i^T \times T_j \bmod T_i$ can be computed with an in-place full product (stored in the extra registers) followed by an in-place division with remainder using the registers of S_i^T and T_j for the quotient and remainder storage. Using the same technique Step 8 requires only k extra space as for Steps 2 to 6. At Step 9, we need $3k$ registers to store G_T, S_i^T, S_i^k and $2k$ registers to store $(b_1, \ldots, b_k)$ and $(z_1, \ldots, z_k)$, plus $k + O(k/\log k)$ extra register for the computation. At Step 12 we re-use the space of G^T, S_i^T for N_i and the extra space of the computation which implies the claim. □

We can now provide our in-place variant for fast interpolation.

Algorithm 6 In-Place Interpolation (InPlaceInterpol)

Input: $(y_1, \ldots, y_n)$ and $(a_1, \ldots, a_n)$ of size n such that $a_i, y_i \in \mathbb{K}$;
Output: $F \in \mathbb{K}[X]$ of size n, such that $F(a_i) = y_i$ for $0 \le i \le n$.
Required: PartInterpol with space complexity $\le c_{pi}k$

1: $s \leftarrow 0$
2: **while** $s < n$ **do**
3: $\quad k \leftarrow \left\lfloor \frac{n-s}{c_{pi}+1} \right\rfloor$
4: $\quad$ **if** $k = 0$ **then** $k \leftarrow n - s$
5: $\quad Y, A \leftarrow (y_1, \ldots, y_{n-s}), (a_1, \ldots, a_{n-s})$
6: $\quad F_{[s..s+k[} \leftarrow \textsc{PartInterpol}(F_{[0..s[}, Y, A, k)$
7: $\quad s \leftarrow s + k$

Theorem 3.6. *Algorithm 6 is correct. It uses $O(1)$ extra space and at most $\frac{1}{2}(c^2 + 23c)\mathsf{M}(n)\log n + O(\mathsf{M}(n)\operatorname{loglog} n)$ operations in $\mathbb{K}$, where $c = 1 + c_{pi}$.*

Proof. The correctness is clear from the correctness of Algorithm PartInterpol. To ensure that the algorithm uses $O(1)$ extra space we notice that at Step 6, $F_{[s+k..n[}$ can be used as work space. Therefore, as soon as $c_{pi}k \le n - s - k$, that is, $k \le \frac{n-s}{c_{pi}+1}$, this free space is enough to run PartInterpol. Note that when $k = 0$, $n - s < c_{pi} + 1$ is a constant, which means that the final computation can be done with $O(1)$ extra space. Let $k_1, k_2, \ldots, k_t$ and $s_1, s_2, \ldots, s_t$ be the values of k and s taken during the course of the algorithm. Since $s_i = \sum_{j=1}^{i} k_j \le n$ with $s_0 = 0$, we have $k_i \le \lambda n(1-\lambda)^{i-1}$, and $s_i \ge n(1 - (1-\lambda)^i)$ where $\lambda = \frac{1}{c_{pi}+1}$. The time complexity $T(n)$ of the algorithm satisfies

$$T(n) \le \sum_{i=1}^{t}\left(\frac{c^2}{2} + \frac{23c}{2}\right)\mathsf{M}(k_i)\log(k_i) + \sum_{i=1}^{t}(n - s_{i-1})\log(s_{i-1}) + O(c^2\mathsf{M}(k_i)\operatorname{loglog} k_i)$$

since $\frac{n-s_{i-1}}{k_i} \le c = c_{pi} + 1$ by definition of k_i. Moreover, we have $\sum_{i=1}^{t}\mathsf{M}(k_i)\log(k_i) \le \mathsf{M}(\sum_i k_i)\log n \le \mathsf{M}(n)\log(n)$. By definition of s_i, we have $n - s_i \le n(1-\lambda)^i$ which gives

$$\sum_{i=1}^{t}(n - s_{i-1})\log(s_{i-1}) \le n\log(n)\sum_{i=1}^{t}(1-\lambda)^i \le (c_{pi}+1)n\log n.$$

This concludes the proof. □

Since $c_{pi} < 6 + \epsilon$ for any $\epsilon > 0$, the complexity can be approximated to $105\mathsf{M}(n)\log(n)$, which is 42 times slower than the fastest interpolation algorithm (see Table 1).

ACKNOWLEDGMENTS

We thank Grégoire Lecerf, Alin Bostan and Michael Monagan for pointing out the references [7, 16].

REFERENCES

[1] D.J. Bernstein. 2008. Fast multiplication and its applications. In *Algorithmic Number Theory*. MSRI Pub., Vol. 44. Cambridge University Press, 325–384.
[2] A. Bostan, G. Lecerf, and É. Schost. 2003. Tellegen's Principle into Practice. In *ISSAC'03*. ACM, 37–44. https://doi.org/10.1145/860854.860870
[3] D. G. Cantor and E. Kaltofen. 1991. On fast multiplication of polynomials over arbitrary algebras. *Acta Inform.* 28, 7 (1991), 693–701. https://doi.org/10.1007/BF01178683
[4] S. A. Cook. 1966. *On the minimum computation time of functions.* Ph.D. Dissertation. Harvard University.
[5] M. Frigo, C. E. Leiserson, H. Prokop, and S. Ramachandran. 1999. Cache-Oblivious Algorithms. In *FOCS'99*. IEEE, 285–297. https://doi.org/10.1109/SFFCS.1999.814600
[6] J. von zur Gathen and J. Gerhard. 2013. *Modern Computer Algebra* (3rd ed.). Cambridge University Press.
[7] J. von zur Gathen and V. Shoup. 1992. Computing Frobenius maps and factoring polynomials. *Comput. Complex.* 2, 3 (1992), 187–224. https://doi.org/10.1007/BF01272074
[8] P. Giorgi, B. Grenet, and D. S. Roche. 2019. Generic reductions for in-place polynomial multiplication. In *ISSAC'19*. ACM, 187–194. https://doi.org/10.1145/3326229.3326249
[9] G. Hanrot, M. Quercia, and P. Zimmermann. 2004. The Middle Product Algorithm I. *Appl. Algebr. Eng. Comm.* 14, 6 (2004), 415–438. https://doi.org/10.1007/s00200-003-0144-2
[10] D. Harvey and J. van der Hoeven. 2019. Polynomial multiplication over finite fields in time O(n log n). (2019). https://hal.archives-ouvertes.fr/hal-02070816/
[11] D. Harvey and D. S. Roche. 2010. An in-place truncated Fourier transform and applications to polynomial multiplication. In *ISSAC'10*. ACM, 325–329. https://doi.org/10.1145/1837934.1837996
[12] A. Karatsuba and Y. Ofman. 1963. Multiplication of Multidigit Numbers on Automata. *Sov. Phys. - Dok.* 7 (1963), 595–596.
[13] A. H. Karp and P. Markstein. 1997. High-precision division and square root. *ACM Trans. Math. Software* 23, 4 (1997), 561–589. https://doi.org/10.1145/279232.279237
[14] H. T. Kung. 1974. On computing reciprocals of power series. *Numer. Math.* 22, 5 (1974), 341–348. https://doi.org/10.1007/BF01436917
[15] R. Moenck and A. Borodin. 1972. Fast modular transforms via division. In *SWAT'72*. IEEE, 90–96. https://doi.org/10.1109/SWAT.1972.5
[16] M. Monagan. 1993. In-place arithmetic for polynomials over Zn. In *DISCO'93*, Vol. 721. Springer, 22–34. https://doi.org/10.1007/3-540-57272-4_21
[17] D. S. Roche. 2009. Space- and Time-efficient Polynomial Multiplication. In *ISSAC'09*. ACM, 295–302. https://doi.org/10.1145/1576702.1576743
[18] A. Schönhage. 1988. Probabilistic computation of integer polynomial GCDs. *J. Algorithms* 9, 3 (1988), 365–371. https://doi.org/10.1016/0196-6774(88)90027-2
[19] A. Schönhage and V. Strassen. 1971. Schnelle Multiplikation großer Zahlen. *Computing* 7, 3 (1971), 281–292. https://doi.org/10.1007/BF02242355
[20] E. Thomé. 2002. Karatsuba multiplication with temporary space of size ≤ n. (2002). https://hal.archives-ouvertes.fr/hal-02396734

Subdivisions for Macaulay Formulas of Sparse Systems

Friedemann Groh
Industrielle Steuerungstechnik GmbH
Stuttgart, Germany

ABSTRACT

In a seminal article [7], D'Andrea describes a method for determining Macaulay-type formulae for the resultants of sparse polynomial systems. His algorithm works recursive, reducing the dimension n of the problem at each step. In doing do, he applies a certain coherent mixed subdivision of the given Newton polytopes into cells, each representing a system with smaller dimension. To simplify this procedure, we insert an intermediate step in which these reduced systems are transferred to the n-dimensional domain of the complete cells. As a consequence, the input system of each iteration step need not contain an additional polytope and only one system per secondary cell has to be considered. The individual subdivisions determined in various steps of the algorithm are combined into a single subdivision of the whole problem. Only then, the matrix for calculating the resultant is determined. To prove our method, we generalize a theorem of [22] on the initial form of resultants with respect to coherent mixed subdivisions.

CCS CONCEPTS

• **Mathematics of computing → Solvers**.

KEYWORDS

Macaulay Formula, Resultants, Polynomial Systems, Polyhedral Subdivisions

ACM Reference Format:
Friedemann Groh. 2020. Subdivisions for Macaulay Formulas of Sparse Systems. In *International Symposium on Symbolic and Algebraic Computation (ISSAC '20), July 20–23, 2020, Kalamata, Greece.* ACM, New York, NY, USA, 8 pages. https://doi.org/10.1145/3373207.3403988

1 INTRODUCTION

Resultants are versatile instruments for solving algebraic problems, such as implicit representations [17]. For example, the equation of the intersection curve of two NURB surfaces is a resultant [2]. This opens up applications in computational geometry: In CAD systems the definition of edge loops in STEP format can be improved, since according to the state of the art such intersection curves are only approximated. [2] describe the implicit equation of the intersection curve of two surfaces as the determinant of a single Bézout matrix. Such a simple representation is not always possible. For general sparse systems (1) Macaulay-type formulas provide a sensible approach, since they allow to calculate resultants as a fraction of two determinants.

$$f_i(x_1, \cdots, x_n) = \sum_{a \in \mathcal{A}_i} c_{i,a} x^a \text{ with } c_{i,a} \neq 0 \text{ and } i = 0, \ldots, n. \quad (1)$$

Further, polynomial systems can be solved via resultants by hiding one of the unknowns in the field of coefficients [13] and [21]. [6] and [12] provide an overview of various methods for solving polynomial systems, and [23] describes numerous applications.

1.1 Previous Work

D'Andrea [7] was the first to prove that resultants of sparse polynomial systems (1) can be calculated via Macaulay style formulas as a fraction of two determinants [20]. These systems are specified by their family $\mathfrak{F} := \{\mathcal{A}_0, \ldots, \mathcal{A}_n\}$ of support sets. D'Andrea's algorithm stepwise reduces the dimension of the problem. For this, it is decomposed by a subdivision $\Delta_{\boldsymbol{b}}(\mathfrak{F})$, which is generated via lifting functions $\omega_i : \mathcal{A}_i \to \mathbb{Q}$ that vanish except a at single point $\boldsymbol{b}$ in one of the support sets. Moreover, he introduced additional auxiliary polytopes, which may have vertices with rational coordinates. It can be interpreted as a twist of the given system and does not change its resultant. Emiris and Konaxis [16] determine a single lifting function to obtain subdivisions suitable for Macaulay-Formulas of generalized unmixed problems. Their algorithm introduces additional points with rational coordinates. In a very recent article, D'Andrea, Jeronimo and Sombra [9] also determine a subdivision suitable for Macaulay formulae, which proves a conjecture of Canny and Emiris.

In view of their significance, there are different approaches for calculating resultants efficiently. They can be understood as determinants of a complex $\boldsymbol{V}^\bullet(\mathfrak{F})$ of finite dimensional vector spaces $V_r(\mathfrak{F})$ embedded in the Koszul complex of given polynomials, as in [18] and [5]. By restricting the differentials of this surrounding Koszul complex, we obtain mappings $D_r : V_r(\mathfrak{F}) \to V_{r-1}(\mathfrak{F})$ with Sylvester-like matrices. For systems of homogeneous polynomials [8] map the terms of this complex via Bézout matrices into suitable dual spaces. This way, they define new differentials in which Bézout and Sylvester-like matrices are combined, so that their dimensions can be reduced, see also [11].

1.2 Main result and open questions

The purpose of the present work is to simplify the algorithm of [7] so that additional auxiliary polytopes are no longer required. Moreover, the refined method yields a coherent subdivision $\Delta_{\mathrm{M}}(\mathfrak{F})$ of the mixed family $\mathfrak{F}$ of support sets $\mathcal{A}_i \subset \mathbb{Z}^n$. So the matrix $\mathbf{M}$ and its sub-matrix $\mathbf{E}$, required to calculate the resultant $\mathrm{Res}(\mathfrak{F})$ via Macaulay's formula (2), can be obtained in a single step at the end of the procedure. As with [7], we decompose the family $\mathfrak{F}$ into cells $\mathfrak{C}_\alpha$ by a subdivision $\Delta_{\boldsymbol{b}}(\mathfrak{F})$, specified by a point $\boldsymbol{b}$ in one

ISSAC '20, July 20–23, 2020, Kalamata, Greece

ACM ISBN 978-1-4503-7100-1/20/07…$15.00
https://doi.org/10.1145/3373207.3403988

of the support sets. Each of these cells contains an essential subfamily $\mathfrak{C}_{\alpha|\vartheta}$ representing a polynomial system with a dimension less than that of the surrounding space. We extend subdivisions $\Delta_{\mathrm{M}}(\mathfrak{C}_{\alpha|\vartheta})$ of this subfamily to the complete system of the cell $\mathfrak{C}_\alpha$. Our main result is Theorem 2.12, which states, that this mapping of subdivisions: $\Delta_{\mathrm{M}}(\mathfrak{C}_{\alpha|\vartheta}) \mapsto \Delta_{\mathrm{M}}(\mathfrak{C}_\alpha)$ preserves the suitability for Macaulay-Formulas.

To prove this theorem, the product formula in Theorem 4.1 of [22], with which the initial form $\mathrm{init}_\omega \mathrm{Res}(\mathfrak{F})$ of a resultant with respect to coherent subdivisions $\Delta_\omega(\mathfrak{F})$ can be calculated, is generalized for arbitrary families $\mathfrak{F}$ of support sets. For this, we apply the redefined resultant in [10], which may also contain multiplicities.

Theorem 12 allows the subdivision $\Delta_{\mathrm{M}}(\mathfrak{F})$ of the input system to be assembled step by step, as in [7] with descending dimensions. It might be more efficient to first determine a globally defined lifting function instead, as in [16]. The support sets' Cayley embedding determine the secondary polytope, which have a face lattice isomorphic to the poset of coherent mixed subdivisions of the family $\mathfrak{F}$, [22] and [15] for an example. It would be interesting to investigate properties of those faces on this secondary polytope, which are assigned to the subdivisions suitable for Macaulay formulae.

1.3 Summary

This article is divided in two parts: Section 2 describes the calculation of subdivisions $\Delta_{\mathrm{M}}(\mathfrak{F})$ suitable for Macaulay-style formulas, and the closing section 3 explains this algorithm using an example. The central section 2 in turn is organized into five parts: Subsection 2.1 first addresses coherent mixed subdivisions $\Delta_\omega(\mathfrak{F})$, then deals with affine lattices $\boldsymbol{L} \subset \mathbb{Z}^n$ generated by the support sets $\mathcal{A}_i$ formed by the polynomial's exponents vectors, the definition of essential families of these support sets and finally introduces the generalized Sylvester matrix $D[\mathfrak{F}]$; the greatest common divisor of all its maximum minors yields the redefined resultant. To prove the method presented, section 2.2 is concerned with initial resultants $\mathrm{init}_\omega(\mathrm{Res}(\mathfrak{F}))$, with respect to coherent subdivisions. They are formed by all terms on faces of the resultant's Newton polytope $\mathcal{N}_{\mathrm{Res}(\mathfrak{F})}$ and can be calculated with the product formula of [22], which we generalize here for arbitrary families $\mathfrak{F}$ of support sets. Therefore we extend the Geometric Lemma of [4] with a result of [7]. The following section 2.3 discusses the transfer $\Delta_{\mathrm{M}}(\mathfrak{F}_\vartheta) \mapsto \Delta_{\mathrm{M}}(\mathfrak{F})$ of a subdivision of an essential subfamily $\mathfrak{F}_\vartheta$ to the complete problem. If the first is suitable for Macaulay formulas, then it is also the second. The fourth subsection 2.4 describes the iteration step inspired by D'Andrea. In doing so, a particular subdivision $\Delta_{\boldsymbol{b}}(\mathfrak{F})$ decomposes the family of the n-dimensional system into smaller problems with reduced dimensions. According to the induction hypothesis, they have subdivisions suitable for Macaulay formulas. These are combined to form a subdivision $\Delta_{\mathrm{M}}(\mathfrak{F})$ of the given problem, whereby we avoid auxiliary polytopes. The concluding section 3 explains the previously described method with an example.

2 SUBDIVISION FOR MACAULAY-FORMULAS

To calculate resultants of algebraic systems (1), defined by the family $\mathfrak{F}$, via Macaulay-style formulas, a mixed subdivision $\Delta_{\mathrm{M}}(\mathfrak{F})$ of the Minkowski sum Q has to be specified, with the latter formed by Newton polytopes $Q_i := \mathrm{conv}(\mathcal{A}_i)$ of the support sets $\mathcal{A}_i$ in $\mathfrak{F}$. Such subdivisions are introduced in [1] and [18] in chapter 7. They consist of cells $\mathfrak{C}_\alpha = \{C_{\alpha,0}, \ldots, C_{\alpha,n}\}$, which in turn are families of subsets $C_{\alpha,i} \subset \mathcal{A}_i$ of the given supports. These cells component's Minkowski sums $\sum_{i=0}^n C_{\alpha,i}$ are disjoint and their union covers the sum of all support sets. Coherent mixed subdivisions (CMD) are determined by lifting functions $\omega_i : \mathcal{A}_i \to \mathbb{Q}$, [18]. They are called tight (TCMD), if in each cell, the dimensions of its individual components' convex hulls $F_{\alpha,i} := \mathrm{conv}(C_{\alpha,i})$ add up to that of the surrounding space. Moreover, the Minkowski sum of these hulls is designated as the domain $\boldsymbol{F}_\alpha$ of the cell.

Definition 2.1. Each cell $\mathfrak{C}_\alpha$, which contains a single vertex $\dim C_{\alpha,i} = 0$ and otherwise only edges $\dim C_{\alpha,k} = 1$ for $k \neq i$, is referred to as *mixed*. All other cells are called *non-mixed*.

In a recursive procedure, the resultants of these cells are also determined. To circumvent restrictions on the family of support sets in this step, we use the refined version of resultants, defined by [10]. Generally, it is a multiple of the sparse-resultants, which is also referred to as the eliminant of the system. For coherent tight subdivisions (TCMD), [4] define a regular sub-matrix $\mathbf{M}$ of the generalized Sylvester map $D[\mathfrak{F}]$, so that its determinant contains the resultant as a factor. Moreover, its maps onto the first space V_0, see also [14]. The non-mixed cells of the TCMD determine in turn a smaller sub-matrix $\mathbf{E}$. We will show that for each family $\mathfrak{F}$ there is a subdivision $\Delta_{\mathrm{M}}(\mathfrak{F})$, so that the resultant can be calculated by means of a Macaulay-style formula, as a fraction of determinants:

$$\mathrm{Res}(\mathfrak{F}) = \frac{\det(\mathbf{M})}{\det(\mathbf{E})}. \tag{2}$$

If the resultant of the sparse polynomial system (1) can be calculated with such an equation, the subdivision $\Delta_{\mathrm{M}}(\mathfrak{F})$ is referred to as suitable for a Macaulay formula. The presented algorithm to determine it is based on [7]. Accordingly, the given family $\mathfrak{F}$ is first decomposed into a series of cells $\mathfrak{C}_\alpha$, using a subdivision $\Delta_{\boldsymbol{b}}(\mathfrak{F})$ formed by lifting functions ω_i, which vanish except at a single point $\boldsymbol{b} \in \mathcal{A}_n$ in the last support. Each of these cells $\mathfrak{C}_\alpha$ contains an essential sub-family $\mathfrak{C}_{\alpha|\vartheta} := \{C_{\alpha|i}\}_{i\in\vartheta}$, representing a polynomial system of reduced dimension. According to the induction hypothesis they have subdivisions $\Delta_{\mathrm{M}}(\mathfrak{C}_{\alpha|\vartheta})$, which are suitable for Macaulay formulas. We introduce a mapping $\Delta_{\mathrm{M}}(\mathfrak{C}_{\alpha|\vartheta}) \mapsto \Delta_{\mathrm{M}}(\mathfrak{C}_\alpha)$ that preserves this property. This way, also the cells with n-dimensional domains are subdivided, and the union of these parts, yields the subdivision $\Delta_{\mathrm{M}}(\mathfrak{F})$ of the complete input system.

2.1 Definitions

Coherent mixed subdivisions of Minkowski sums Q with the family $\mathfrak{F}$ of support sets are determined by lifting functions $\omega_i : \mathcal{A}_i \to \mathbb{Q}$; they are referred to as $\Delta_\omega(\mathfrak{F})$. To obtain them, the given support sets are extended $\hat{\mathcal{A}}_i := \{(a, \omega_i(a)) : a \in \mathcal{A}_i\}$ with the values of the lifting functions. These points' convex hulls are accordingly designated $\hat{Q}_i := \mathrm{conv}(\hat{\mathcal{A}}_i)$. Finally, the cells of $\Delta_\omega(\mathfrak{F})$ are formed by the facets on the lower envelope $\partial_-(\hat{Q})$ of the Minkowski sum of all lifted polytopes, $\hat{Q} := \sum_{i=0}^n \hat{Q}_i$. These facets are determined by their inward normals $\hat{v}_\alpha = (\boldsymbol{v}_\alpha, \hat{v}_{\alpha,n+1}) \in \mathbb{Q}^n \times \mathbb{Q}$. The last coordinates $\hat{v}_{\alpha,n+1}$ are positive, since the lower envelope of the lifted sum $\hat{Q}$ is considered here, as in [19]. We refer to the face in direction of the inward normal $\hat{v}_\alpha$ as $\boldsymbol{F}_\alpha(\hat{Q}) := \mathrm{face}_{\hat{v}_\alpha}\hat{Q}$. Furthermore, the

canonical projection $\pi_1 : \mathbb{R}^n \times \mathbb{R} \to \mathbb{R}^n$ onto the first n-coordinates maps facets on $\partial_-(\hat{Q})$ to the domains of corresponding cells: $\boldsymbol{F}_\alpha = \pi_1(\boldsymbol{F}_\alpha(\hat{Q}))$.

Sub-families of supports selected by an index set $\vartheta \subset \{0, \ldots, n\}$ will be designated as $\mathfrak{F}_\vartheta$. The finest lattice of integers contained in the affine hull of the Minkowski sum $\mathcal{A}_\vartheta := \sum_{i\in\vartheta}^n \mathcal{A}_i$ of their support sets, is referred to as $\boldsymbol{L}_\vartheta := \mathbb{Z}^n \cap \mathrm{aff}_{\mathbb{R}}(\mathcal{A}_\vartheta)$. For the complete family $\mathfrak{F}$ this lattice is abbreviated with $\boldsymbol{L}$ in the following.

Definition 2.2. A family $\mathfrak{F}_\vartheta = \{\mathcal{A}_i\}_{i\in\vartheta}$ of support sets is *essential* if the rank of the affine lattice $\boldsymbol{L}_\vartheta$ equals $\#\vartheta - 1$ and moreover $\mathrm{rank}(\boldsymbol{L}_\theta) \geq \#\theta$ for all proper subsets θ of ϑ.

If there is a unique essential $\mathfrak{F}_\vartheta$ family contained in $\mathfrak{F}$, then according to Corollary 1.1. in [22], the resultant $\mathrm{Res}(\mathfrak{F})$ depends only on coefficients of the polynomials (1) with supports $\mathcal{A}_i$ in $\mathfrak{F}_\vartheta$. If there is no such family, the resultant is equal to one. In this article, $\mathrm{Res}(\mathfrak{F})$ denotes the *redefined* resultant of [10]. For non-essential families $\mathfrak{F}$ of supports it is a multiple of the *sparse* resultant, which in turn is irreducible or constant. As in [10], the latter is referred to as $\mathrm{Elim}(\mathfrak{F})$ here. If the family of support sets is essential and the affine integer lattice $\mathrm{aff}_{\mathbb{Z}}(\sum_{i=0}^n \mathcal{A}_i)$ generated by their Minkowski sum agrees with the generally finer grid $\boldsymbol{L}$, then the eliminant and resultant are identical. To determine resultants, we consider the Sylvester map, which assigns an element in the ideal $\langle f_0, \ldots, f_n\rangle$ to each $n+1$ tuple of polynomials. Here, $\boldsymbol{e}_i \in \mathbb{C}^n$ denote unit vectors.

$$D[\mathfrak{F}] : x^p \boldsymbol{e}_i \mapsto x^p\, f_i(x) \tag{3}$$

As in Canny and Emiris [3], [4] its image space $V_0(\mathfrak{F})$ is restricted to polynomials with exponent vectors $q \in \boldsymbol{L}$ inside the Minkowski sum Q of all Newton polytopes shifted by the generically chosen vector $\delta \in \mathbb{Q}^n$. The corresponding range space is denoted by $V_1(\mathfrak{F})$, where the sum $P_i := \sum_{k\in\{0,\ldots,n\}\setminus\{i\}} Q_k$ is required.

$$V_0(\mathfrak{F}) := \mathrm{span}\{x^q : q \in \boldsymbol{L} \cap (Q + \delta)\} \tag{4}$$

$$V_1(\mathfrak{F}) := \oplus_{i=0}^n \left(\mathrm{span}\{x^p \boldsymbol{e}_i : p \in \boldsymbol{L} \cap (P_i + \delta)\right)\} \tag{5}$$

These definitions apply to any family $\mathfrak{F}$ of support sets. If $q \in p + \mathcal{A}_i$ applies, the components of $D[\mathfrak{F}]$ are given by (6), otherwise they are zero.

$$D^q_{p,i} = c_{i,q-p} \quad \text{with} \quad q \in p + \mathcal{A}_i \tag{6}$$

Since the cells $\mathfrak{C}_\alpha$ of a subdivision are families of certain support sets $C_{\alpha,i} \subset \mathcal{A}_i$ as well, they determine via Definition (4) subspaces $V_r(\mathfrak{C}_\alpha) \subset V_r(\mathfrak{F})$. All the Minkowski sums $\sum_{i=0}^n C_{\alpha,i}$ of the cell components form a disjoint cover of the complete set $\mathcal{A}$. Therefore, the image and range of the Sylvester map decomposes into a direct sum of the subspaces, related to the cells $\mathfrak{C}_\alpha$ of the subdivision.

$$V_r(\mathfrak{F}) = \oplus_\alpha V_r(\mathfrak{C}_\alpha) \tag{7}$$

This subspaces are not invariant under actions of the Sylvester map.

2.2 Initial resultant of a coherent subdivision

D'Andrea [7] realised the importance of initial resultants to specify a submatrix **E**, whose determinant is the denominator of the Macaulay formula (2). To calculate them, we replace the coefficients of the algebraic system by functions $t \mapsto c_{i,a}t^{\omega_i(a)}$, wherein the exponents are given by the lifting functions $\omega_i : \mathcal{A}_i \to \mathbb{Q}$, as in [22] and [4]. The corresponding resultant of the polynomial system varied by the parameter t is denoted as $\mathrm{Res}(\mathfrak{F}_t)$ subsequently. As we consider the lower envelope of the lifted polytope $\hat{Q}$ to obtain subdivisions, the initial form $\mathrm{init}_\omega(\mathrm{Res}(\mathfrak{F}))$ of the resultant with respect to the given lifting functions, is that term of the t-dependent resultant $t \mapsto \mathrm{Res}(\mathfrak{F}_t)$ which has the lowest degree in the parameter t of variation. Resultants are polynomials, which in turn depend on the coefficients $c_{i,a}$ of the given system (1). To describe initial forms also by the respective Newton polytope $\mathcal{N}_{\mathrm{Res}}$ it is convenient to combine the values of each function ω_i on the support sets $\mathcal{A}_i$ in a single lifting vector $\omega \in \mathbb{Q}^m$ with $m := \sum_{i=0}^n \#\mathcal{A}_i$ denoting the total number of terms in the system.

Definition 2.3 (Initial Resultant). The initial form $\mathrm{init}_\omega(\mathrm{Res}(\mathfrak{F}))$ is the sum of all terms with exponent vectors that lie on $\mathrm{face}_\omega(\mathcal{N}_{\mathrm{Res}})$ in direction of the lifting vector $\omega \in \mathbb{Q}^m$, which defines the coherent subdivision $\Delta_\omega(\mathfrak{F})$. It agrees with the part of the t-dependent resultant $\mathrm{Res}(\mathfrak{F}_t)$ that has the smallest power with respect to the parameter t of variation.

The given lift functions define a convex function h which assigns a point in the shifted Minkowski's sum $Q + \delta$ to the height of the lower envelope of the lifted polytope $\hat{Q} + (\delta, 0)$.

$$h(q) := \min\left\{y \in \mathbb{R} : (q, y) \in \hat{Q} + (\delta, 0)\right\} \tag{8}$$

$$h_i(p) := \min\left\{y \in \mathbb{R} : (p, y) \in \hat{P}_i + (\delta, 0)\right\} \tag{9}$$

The t-dependent Sylvester maps, obtained when replacing the coefficients by $c_{i,a}t^{\omega_i(a)}$, are referred to as $D(t)$ subsequently. To determine the initial resultant $\mathrm{init}_\omega(\mathrm{Res}(\mathfrak{F}))$, we further scale the rows and columns of their matrices and therefore introduce the mappings $\boldsymbol{T}_0 : x^q \mapsto t^{h(q)}x^q$ and $\boldsymbol{T}_1 : x^p\boldsymbol{e}_i \mapsto t^{h_i(p)}x^p\boldsymbol{e}_i$, using the level functions (8), for the spaces $V_k(\mathfrak{F})$. These automorphisms have diagonal matrices. Finally, we define the scaled t-dependent Sylvester maps $\hat{D}(t) := \boldsymbol{T}_0{}^{-1}D(t)\boldsymbol{T}_1$, which have the components, with respect to the base vectors x^q and $x^p\boldsymbol{e}_i$:

$$\hat{D}^q_{p,i}(t) = t^{-h(q)}D^q_{p,i}(t)\, t^{h_i(p)} = t^{h_i(p)+\omega_i(q-p)-h(q)}\, D^q_{p,i}. \tag{10}$$

Their initial parts with respect to the lowest degree of the parameter t decomposes into a direct sum which is compatible with (7) and thus corresponds to the cells of the subdivision $\Delta_\omega(\mathfrak{F})$ under consideration. This statement will be proven by the following two lemmas. First we note, that the fourth point of Lemma 3.11 in [7] is valid for any coherent mixed subdivision $\Delta_\omega(\mathfrak{F})$. In a second step, we use it to generalize the Geometric Lemma 4.5 given in [4]. By definition, $P_i + Q_i = Q$ applies.

Lemma 2.4. *Let $\boldsymbol{v}_\alpha$ and $\boldsymbol{v}_b$ be directions generating two different cells of a coherent subdivision, further $\boldsymbol{F}_\alpha(\hat{P}_i)$ and $\boldsymbol{F}_b(\hat{Q})$ denote the facets they support on the lifted partial sums $\hat{P}_i$ and $\hat{Q}$. Then, for each point $\hat{p}$ inside the first facet $\boldsymbol{F}_\alpha(\hat{P}_i)$ the intersection $(\hat{p} + \hat{Q}_i) \cap \boldsymbol{F}_b(\hat{Q})$ is empty.*

Proof. Let $\boldsymbol{u}$ be the projection of the inner normal $\boldsymbol{v}_b$ of the facet $\boldsymbol{F}_b(\hat{Q})$ to the co-dimension one subspace, which is parallel to the first facet $\boldsymbol{F}_\alpha(\hat{P}_i)$. Further, suppose there was a point $\hat{q}$ in the intersection $(\hat{p}+\hat{Q}_i)\cap\boldsymbol{F}_b(\hat{Q})$. The points $\hat{q}_\lambda := \hat{q}+\lambda\boldsymbol{u}$ are outside the sum $\hat{Q}$ for negative parameters λ. Moreover, we consider the parallel line of points $\hat{p}_\lambda := \hat{p}+\lambda\boldsymbol{u}$ on the facet $\boldsymbol{F}_\alpha(\hat{P}_i)$. It is translated by the difference $\hat{q} - \hat{p} \in \hat{Q}_i$, so that each point $\hat{q}_\lambda$ is contained in the sum $\hat{p}_\lambda + \hat{Q}_i$ for any $\lambda \in \mathbb{R}$. Since $\hat{p}$ is an interior point on its facet, there

is a neighbourhood of $\lambda = 0$ so that $\hat{p}_\lambda \in \boldsymbol{F}_\alpha(\hat{P}_i)$ is also valid. The convexity of all Newton polytopes implies inclusion $\hat{p}_\lambda + \hat{Q}_i \subset \hat{Q}$ for these parameters. However, this contradicts $\hat{q}_\lambda \notin \hat{Q}$ for $\lambda < 0$, which proves the lemma. □

Lemma 2.5 (Geometric). *Let $\hat{p}$ be an interior point of the facet $\boldsymbol{F}_\alpha(\hat{P}_i)$, then $\hat{p} + \hat{Q}_i \subset \hat{Q}$ and its intersection with the lower convex hull $\partial_-(\hat{Q})$ equals the face: $\hat{p} + \boldsymbol{F}_\alpha(\hat{Q}_i) = (\hat{p} + \hat{Q}_i) \cap \partial_-(Q)$.*

Proof. The first inclusion follows from the convexity of Newton polytopes and the assumption $P_i + Q_i = Q$. The lower hull $\partial_-(\hat{Q})$ is the union of the cells $\boldsymbol{F}_b(\hat{Q})$, which in turn determine a subdivision of $(Q, \mathfrak{F})$. According to Lemma 2.4, intersections of the polytope $\hat{p}+\hat{Q}_i$ with facets, whose indices b differ from α, are empty. Thus, it is sufficient to examine case $(\hat{p} + \hat{Q}_i) \cap \boldsymbol{F}_\alpha(\hat{Q})$ only. To begin with, inclusion $\hat{p} + \boldsymbol{F}_\alpha(\hat{Q}_i) \subset \boldsymbol{F}_\alpha(\hat{Q})$ results from the additivity of the mapping to faces $\boldsymbol{F}_\alpha(\hat{P}_i) + \boldsymbol{F}_\alpha(\hat{Q}_i) = \boldsymbol{F}_\alpha(\hat{Q})$ and the assumption $\hat{p} \in \boldsymbol{F}_\alpha(\hat{P}_i)$. Let us suppose that there is a point $\hat{q}$ in the polytope $\hat{Q}_i$ which is not in its face in the direction of $\boldsymbol{v}_\alpha$, then there would be another point $\hat{q}_\alpha \in \boldsymbol{F}_\alpha(\hat{Q})$ so that the product $\boldsymbol{v}_\alpha^T \cdot (\hat{q} - \hat{q}_\alpha)$ is positive. Hence, $\hat{p}$ cannot lie in this facet, so that the identity $\hat{p} + \boldsymbol{F}_\alpha(\hat{Q}_i) = (\hat{p} + \hat{Q}_i) \cap \boldsymbol{F}_\alpha(\hat{Q})$ is valid. □

For tight coherent mixed subdivisions (TCMD), the sum of the dimensions of all surfaces in each cell $\mathfrak{C}_\alpha$ agrees with that of the ambient space: $\sum_{i=0}^n \dim F_{\alpha,i} = n$. Because we assume that both $\boldsymbol{F}_\alpha(\hat{P}_i)$ and $\boldsymbol{F}_\alpha(\hat{Q})$ are facets, $\boldsymbol{F}_\alpha(\hat{Q}_i)$ must be a vertex. Therefore, in this case, we obtain the Geometric Lemma of [4]. It also shows that, if the sum Q is n-dimensional, regular Newton matrices can be determined exactly as in [4]. With this algorithm $n+1$ of such matrices $\mathbf{M}_k$ can be determined, so that according to Theorem 7.4 in [4] the degree of homogeneity of their determinants with the mixed volume MV of all Newtonian polytopes except Q_k is given.

$$\deg_k(\det \mathbf{M}_k) = MV(Q_0, \ldots, Q_{k-1}, Q_{k+1}, \ldots, Q_n) \tag{11}$$

In this more general case, however, their greatest common divisor can be a multiple of the *eliminant*. In the recursive method presented, we need the Macaulay formula also for other systems formed by cells of decompositions. Here, its essentiality cannot be assumed. Hence, a generalized version of Theorem 3.1 in [22] suitable for the redefined resultant is required: Proposition 4.17 in [9] provides the desired statement.

$$\det \mathbf{M}_k = p_k \operatorname{res}(\mathfrak{F}) \quad \text{with} \quad \deg_k(p_k) = 0. \tag{12}$$

Since this equation applies to all indices $k = 0, \ldots, n$ the resultant can be specified as in Corollary 4.21, [9], via determinants of the related Newton-matrices $\mathbf{M}_k$ of Canny-Emiris type.

Proposition 2.6. *The redefined resultant $Res(\mathfrak{F})$ is the greatest common divisor of* $\det \mathbf{M}_0, \ldots, \det \mathbf{M}_n$ *with Newton matrices* $\mathbf{M}_k$.

Subsequently, the last polynomial $f_n(x)$ always has the particular meaning in the Canny Emiris algorithm, so that we only consider the Newton matrix $\mathbf{M} = \mathbf{M}_n$. The degree of homogeneity of the redefined resultant is given according to Proposition 3.4 in [10] by the mixed volume as in equation (11).

$$\deg_k(\operatorname{res}(\mathfrak{F})) = MV(Q_0, \ldots, Q_{k-1}, Q_{k+1}, \ldots, Q_n) \tag{13}$$

Since the Geometric lemma 2.5 does not presuppose the essentiality of $\mathfrak{F}$, the Theorem 6.4 in [4], which states that the Newton matrices are regular, can be extended to the general case. In the proof of this theorem a leading Newton matrix is defined as in Equation (10), which is diagonal if ω generates a *tight* subdivision. For general lifting functions ω, this results in a blockwise diagonal matrix as with Proposition 3.12. in [7].

Inspired by this, we will examine the leading terms of the t-dependent Sylvester map $\hat{D}(t)$, with matrix elements $\hat{D}^q_{p,i}(t)$ given in (10) where exponent vectors $p \in (P_i+\delta)\cap\mathbb{Z}^n$ and $q \in (Q+\delta)\cap\mathbb{Z}^n$. It is sufficient to regard only pairs with differences $a := q - p$ in the support set $\mathcal{A}_i$, since the other matrix elements vanish. Further, we consider the points $\hat{p} := (p, h_i(p))$ and $\hat{q} := (q, h(q))$, extended by the level functions introduced in (8). Due to the displacement with $\delta \in \mathbb{Q}^n$ we may assume, the first point $\hat{p}$ is located in the interior of some facet $\boldsymbol{F}_\alpha(\hat{P}_i) + \hat{\delta}$, which is moved by the vector $\hat{\delta} := (\delta, 0)$ here. The second point $\hat{q}$ is on the lower envelope $\partial_-(\hat{Q})$ of the polytope $\hat{Q}$ and the vector $\hat{a} := (a, \omega_i(a))$ finally is element of the corresponding polytope $\hat{Q}_i$. Consequently, Lemma 2.5 can be reformulated via the level functions (8), whereby the first inclusion $\hat{p} + \hat{Q}_i \subset \hat{Q}$ implies the inequality:

$$h_i(p) + \omega_i(q - p) \geq h(q) \tag{14}$$

The key statement says, however, that (14) becomes an equation, if and only if vector $\hat{a}$ lies in the face $\boldsymbol{F}_\alpha(\hat{Q}_i)$; projected down to the first $n+1$ coordinates, this condition is equivalent to $a \in \boldsymbol{F}_{\alpha,i}$. By definition of subdivisions, the latter polytope $\boldsymbol{F}_{\alpha,i}$ is the convex hull of the point set $C_i \subset \mathcal{A}_i$ which is included in the cell $\mathfrak{C}_\alpha$. This allows us to decompose the leading part of $\hat{D}(t)$.

Proposition 2.7. *The initial part of $\hat{D}(t)$ act on the subspaces formed by the cells $\mathfrak{C}_\alpha$ of the considered subdivision, $init_\omega D|_{V_1(\mathfrak{C}_\alpha)} : V_1(\mathfrak{C}_\alpha) \to V_0(\mathfrak{C}_\alpha)$.*

Proof. To verify this statement, we consider the matrix components $\hat{D}^q_{p,i}(t)$ of each t-dependent scaled Sylvester map with respect to a base element $x^p \boldsymbol{e}_i \in V_1(\mathfrak{C}_\alpha)$ of the subspace determined the cell $\mathfrak{C}_\alpha$ and another base vector x^q in its target space. Correspondingly, the relation $a = q - p \in \mathcal{A}_i$ is valid. Because of inequality (14), none of the t-exponents of this matrix components (10) is negative. The extended exponent vector $\hat{p} = (p, h_i(p))$ of the argument $x^p \boldsymbol{e}_i$ is located inside the shifted facet $\boldsymbol{F}_\alpha(\hat{P}_i)$. As a consequence of Lemma 2.5 an exponent of t vanish exactly, when the difference $a \in C_{\alpha,i}$ is contained in the i-th component of the cell $\mathfrak{C}_\alpha$. Hence, the exponent vector $q \in (Q + \delta) \cap \mathbb{Z}^n$ is element of the sum $\sum_{k=0}^n C_{\alpha,k}$. Since $\Delta_\omega(\mathfrak{F})$ is a subdivision, we conclude that there is no other cell $\mathfrak{C}_b$ with a domain $F_b + \delta$ containing this vector. Therefore, the initial parts $\operatorname{init}_\omega(D)$ map the subspaces $V_1(\mathfrak{C}_\alpha)$ to $V_0(\mathfrak{C}_\alpha)$. □

According to Theorem 4.1 of [22], the initial form $\operatorname{init}_\omega(\operatorname{Res}(\mathfrak{F}))$ is a product of smaller resultants $\operatorname{Res}(\mathfrak{C}_\alpha)$ of systems whose support sets are just the cells $\mathfrak{C}_\alpha = \{C_{\alpha,0}, \ldots, C_{\alpha,n}\}$ of the decomposition $\Delta_\omega(\mathfrak{F})$. Because re-defined resultants already include the required multiplicities, we may state this theorem in simplified form, without requirements to the family $\mathfrak{F} = \{\mathcal{A}_0, \ldots, \mathcal{A}_n\}$ of support sets.

Theorem 2.8. *For any lifting functions $\omega_i : \mathcal{A}_i \to \mathbb{Q}$ defined on the support sets, the initial resultant $init_\omega Res(\mathfrak{F})$ is given by the*

product (15) of the cell's resultants, where the index α runs through all cells $\mathfrak{C}_\alpha$ of the subdivision defined by the considered lifting ω_i:

$$init_\omega\,(Res(\mathfrak{F})) = \prod_\alpha Res(\mathfrak{C}_\alpha). \tag{15}$$

Proof. The t-dependent automorphisms $\boldsymbol{T}_r$ have diagonal matrices. So, it can be derived that the greatest common divisors of $D(t)$ and $\hat{D}(t) = \boldsymbol{T}_0{}^{-1}D(t)\boldsymbol{T}_1$ differ by only a factor t^η with some rational exponent. Consequently, their terms coincide with the respective smallest powers in the parameter t. Theorem 2.6 and the decomposition given in Proposition 2.7 finally yields the product formula of the resultant's initial form. □

2.3 Extend essential subdivisons

In order to reduce the dimension of mixed sparse polynomial systems in each iteration step, as with [7], resultants of systems with an essential subfamily of support sets must be considered. We refer to the set of indexes of this supports as $\vartheta \subset \{0,\dots,n\}$ and accordingly denote the essential subfamily with $\mathfrak{F}_\vartheta$; further, $\mathcal{A}_\vartheta := \sum_{i\in\vartheta}\mathcal{A}_i$ designates the Minkowski sum of supports selected by the index set ϑ. The aim of this paragraph is to extend a mixed subdivision $\Delta_{\mathrm{M}}(\mathfrak{F}_\vartheta)$ of the essential system $(Q_\vartheta, \mathfrak{F}_\vartheta)$ to the complete problem $\Delta_{\mathrm{M}}(\mathfrak{F})$, in such a way that if the Macaulay formula holds in $\Delta_{\mathrm{M}}(\mathfrak{F}_\vartheta)$, it can be transferred to $\Delta_{\mathrm{M}}(\mathfrak{F})$. The finest lattice of integers contained in the affine hull of the Minkowski sum $\mathcal{A}_\vartheta$ of all essential support sets, is referred to as $\boldsymbol{L}_\vartheta := \mathbb{Z}^n \cap \mathrm{aff}(\mathcal{A}_\vartheta)$. Its dimension equals $n_\vartheta := \#\vartheta - 1$. Moreover, $\bar{\vartheta} := \{0,\dots,n\}\setminus\vartheta$ denotes the complement of the index set ϑ of the essential family. The algorithm extending subdivisions of this essential subfamily to the complete system relies on Theorem 1.1 in [22], of which we only need the following special case here:

Theorem 2.9 (Sturmfels). *The resultant $Res(\mathfrak{F})$ of a family $\mathfrak{F}$ of support sets is non-constant if and only if for all subsets $\vartheta \subset \{0,1,\dots,n\}$ the maximum of the differences between the cardinality $\#\vartheta$ and the rank of the lattice $\boldsymbol{L}_\vartheta$ fulfills:* $\max_\vartheta(\#\vartheta - rk(\boldsymbol{L}_\vartheta)) = 1$. □

We now consider a system with the support sets $\mathcal{A}_0,\dots,\mathcal{A}_n$ in which the resultant $\mathrm{Res}(\mathfrak{F})$ is not constant and also the Minkowski sum Q of all Newton polytopes Q_i is n-dimensional. Moreover, we assume the subfamily $\mathfrak{F}_\vartheta := \{\mathcal{A}_i : i\in\vartheta\}$ to be essential. As a consequence, the sum of these sets generates a n_ϑ-dimensional affine subspace. According to Theorem 2.9, the dimension of the affine hull of $\mathcal{A}_i + \sum_{k\in\vartheta}\mathcal{A}_k$ must increase by one, if we add one of the support sets $\mathcal{A}_i$ which is not included in the essential family $i \in \bar{\vartheta}$. Therefore, there must be an edge E_i in the face lattice of Q_i that is not in the subspace parallel to $\mathrm{aff}(\mathcal{A}_\vartheta)$. This procedure continues iteratively to the last support set. In doing so, we obtain edges $E_i \subset Q_i$ for each Newton polytope not included in the essential family. Their direction vectors are linearly independent. We define the parallelotope $E_{\bar{\vartheta}} := \sum_{i\in\bar{\vartheta}} E_i$, which is $\#\bar{\vartheta}$-dimensional.

In the next step, we determine a coherent mixed subdivision $\Delta_\omega(\mathfrak{F})$ of $(Q,\mathfrak{F})$, which contains the polytope $Q_\vartheta + E_{\bar{\vartheta}}$ as one of the cell-domains. For this purpose, the lift functions $\omega_i : \mathcal{A}_i \to \mathbb{Q}$ are chosen so that they vanish for each index $i\in\vartheta$ in the essential family. For the other support sets, they have a constant negative value at the grid points on each edge E_i with $i\in\bar{\vartheta}$ and they are zero at all remaining points:

$$\begin{aligned} &i\in\vartheta:\ \omega_i = 0\\ &i\in\bar{\vartheta}:\ \omega_i(\mathcal{A}_i\cap E_i) = -1 \ \text{ and } \omega_i(\mathcal{A}_i\setminus E_i) = 0\ . \end{aligned} \tag{16}$$

To obtain the different cells that produce this lifting function, we examine the facets which are supported by different direction vectors $\hat{v}_\alpha = (\boldsymbol{v}_\alpha, \hat{v}_{\alpha,n+1}) \in \mathbb{Q}^n\times\mathbb{Q}$ on the lower envelope of the lifted polytope $\hat{Q}$. The vector $\hat{v}_0 = (0,\dots,0,1)$ in the direction of the additional coordinate supports the cell: $\mathfrak{C}_0 = \{C_{0,0},\dots,C_{0,n}\}$ with $C_{0,i} = \mathcal{A}_i$ for $i\in\vartheta$ and $C_{0,i} = \mathcal{A}_i\cap E_i$ for the other indexes $i\in\bar{\vartheta}$, which we will call the *primary cell* of the subdivision $\Delta_\omega(\mathfrak{F})$ in the following. Its domain is as intended the sum $\boldsymbol{F}_0 = Q_\vartheta + E_{\bar{\vartheta}}$.

For any other direction at least one of the vector's $\boldsymbol{v}_\alpha$ coordinates does not vanish. The cells $\mathfrak{C}_\alpha$ they determine are referred to as *secondary cells* of the subdivision, with $\alpha > 0$. Since the lifted Newton polytopes $\hat{Q}_i$ of the essential family lie in the plane with zero extra coordinate, the vector $\boldsymbol{v}_\alpha \in \mathbb{Q}^n$ defines their components $C_{\alpha,i}$ with $i\in\vartheta$ in the cell $\mathfrak{C}_\alpha$. Consequently, the sum of their faces $F_{\alpha,i} = \mathrm{conv}(C_{\alpha,i})$ is less than n_ϑ-dimensional.

$$\dim\sum_{i\in\vartheta} F_{\alpha,i} < n_\vartheta \tag{17}$$

Proposition 2.10. *The resultant of the complete system is a multiple of that obtained by the essential system: $Res(\mathfrak{F}) = Res(\mathfrak{F}_\vartheta)^{d_E}$. The exponent d_E agrees with the number of lattices parallel to $\boldsymbol{L}_\vartheta$, for which the intersection with the domain of the shifted primary cell $\boldsymbol{F}_0 + \delta$ is not empty.*

Proof. Both the resultant of the essential sub-system subsystem $\mathrm{Res}(\mathfrak{F}_\vartheta)$ and that of the complete system $\mathrm{Res}(\mathfrak{F})$ are divisible by the same eliminant $\mathrm{Elim}(\mathfrak{F})$, which implies that the Newton polytopes $\mathcal{N}_{\mathrm{Res}(\mathfrak{F}_\vartheta)}$ and $\mathcal{N}_{\mathrm{Res}(\mathfrak{F})}$ of either polynomial are proportional to each other in a rational ratio. Since the lifting vector ω vanishes on each support $\mathcal{A}_i$ in the essential sub-family $i\in\vartheta$, it is orthogonally on these polytopes, so that the resultants coincide with their initial terms with respect to the lifting. Consequently, the resultant $\mathrm{Res}(\mathfrak{F})$ of the complete polynomial system can be examined with the product formula in Theorem 2.8 of Sturmfels, as in equation (18) below.

For secondary cells $\mathfrak{C}_\alpha$ with $\alpha > 0$ the rank of the affine lattice generated by the sum $C_{\alpha|\vartheta}$ is less than the dimension $n_\vartheta = \#\vartheta - 1$ of the affine subspace, which contains the supports of the essential family $\mathcal{A}_\vartheta$. Thus, according to Theorem 2.9 the related resultants must be constant one: $\mathrm{Res}(\mathfrak{C}_\alpha) = 1$, since the maximum of the differences $\#\vartheta - \mathrm{rk}(\vartheta) > 1$ is greater than 1. In this way we obtain the identity

$$\mathrm{Res}(\mathfrak{F}) = \mathrm{init}_\omega(\mathrm{Res}(\mathfrak{F})) = \prod_\alpha \mathrm{Res}(\mathfrak{C}_\alpha) = \mathrm{Res}(\mathfrak{C}_0)\ . \tag{18}$$

In the remainder we calculate the resultant $\mathrm{Res}(\mathfrak{C}_0)$ of the system associated with the primary cell. To determine the space $V_1(\mathfrak{C}_0)$ according to equation (5), we consider the Newton polytopes of the primary cell's components: They are Q_i for $i\in\vartheta$ and E_k for $k\in\bar{\vartheta}$. The space $V_1(\mathfrak{C}_0)$ is spanned by $x^p\boldsymbol{e}_i$, where $i\in\vartheta$, since each edge E_k is one-dimensional and therefore must be a summand of the n-dimensional polytope containing the exponent vectors. Hence, the Sylvester map $D[\mathfrak{C}_0]$ associated with the primary cell contains only polynomials of the essential sub-family. As it turns out, this

map decomposes into smaller independent parts, all of which are isomorphic to $D[\mathfrak{F}_\vartheta]$.

To verify this statement, we move the Minkowski sum Q_ϑ by vectors $\xi_s \in \delta + E_{\bar\vartheta}$ within the shifted parallelotope, so that the intersections $(\xi_s + Q_\vartheta) \cap \mathbb{Z}^n$ are not empty. The number of such translated polytopes is denoted by d_E and their index is s. Accordingly, the two vector spaces $V_r(\mathfrak{C}_0)$ can be decomposed into a direct sum of d_E subspaces.

$$V_{s,0}(\mathfrak{C}_0) := \text{span}\left(x^q : q \in (\xi_s + Q_\vartheta) \cap \mathbb{Z}^n\right) \tag{19}$$

$$V_{s,1}(\mathfrak{C}_0) := \text{span}\left(x^p \boldsymbol{e}_i : p \in (\xi_s + Q_{\vartheta\setminus\{i\}}) \cap \mathbb{Z}^n, i \in \vartheta\right) \tag{20}$$

Since $Q_{\vartheta\setminus\{i\}} + Q_i = Q_\vartheta$ for each $i \in \vartheta$, the Sylvester map of the primary cell acts on these subspaces, $D[\mathfrak{C}_0] : V_{s,1}(\mathfrak{C}_0) \to V_{s,0}(\mathfrak{C}_0)$. To link these systems with the polynomial system of the essential subfamily $\mathfrak{F}_\vartheta$, we split the shift vector δ into a part δ_ϑ parallel to the affine subspace $\text{aff}(\mathcal{A}_\vartheta)$ and a vector δ_E within the sub-space spanned by the direction vectors of the parallelotope $E_{\bar\vartheta}$. Each intersection $(Q_\vartheta + \xi_s) \cap \mathbb{Z}^n$ is in bijection to the set $(Q_\vartheta + \delta_\vartheta) \cap \boldsymbol{L}_\vartheta$ of points, via the affine mapping

$$\phi_s : q' \mapsto q = q' + \xi_s - \delta_\vartheta. \tag{21}$$

In addition, the restrictions of $D[\mathfrak{C}_0]$ to the subspaces $V_{s,1}(\mathfrak{C}_0)$ are isomorphic to the Sylvester map $D[\mathfrak{F}_\vartheta]$ of the essential subsystem, which is a consequence of the identity $D[\mathfrak{C}_0] \circ \phi_s = \phi_s \circ D[\mathfrak{F}_\vartheta]$. This proves the proposition, since there are d_E such sub-matrices and by definition, d_E is the number of lattices $\phi_s(\boldsymbol{L}_\vartheta)$ for which the intersection with the shifted domain of the primary cell $\boldsymbol{F}_0 + \delta$ is not empty. □

Subsequently, we show how the Macaulay formula is transferred from the system of the essential subfamily $\mathfrak{F}_\vartheta$ to its completion $\mathfrak{F}$. For this we consider the decomposition of the vector spaces $V_r(\mathfrak{F})$ regarding the subdivision $\Delta_\omega(\mathfrak{F})$ into a primary and secondary component: $V_r(\mathfrak{C}_0)$ and $\tilde V_r(\mathfrak{F}) := \oplus_{\alpha>0} V_r(\mathfrak{C}_\alpha)$, respectively. The first subspace is further split up into parts, which are isomorphic to $V_r(\mathfrak{F}_\vartheta)$ via mapping (21).

$$V_r(\mathfrak{F}) = V_r(\mathfrak{C}_0) \oplus \tilde V_r(\mathfrak{F}) = \left(\oplus_s V_{s,r}(\mathfrak{C}_0)\right) \oplus \tilde V_r(\mathfrak{F}) \tag{22}$$

Let $\Delta_\text{M}(\mathfrak{F}_\vartheta)$ be a mixed subdivision of the essential family with cells $\mathfrak{c}_b$, formed by faces $G_{b,i}$, which we numerate with the index set of the essential supports $i \in \vartheta$. By adding the edges E_i to these cells, this subdivision can be extended to the primary cell. This way, we can define the *composite subdivision* of the complete system:

Definition 2.11. In the composite subdivision $\Delta_\text{M}(\mathfrak{F})$ the primary cell $\mathfrak{C}_0$ is replaced by the extended cells $\mathfrak{C}_{0,b} = \{G_{b,0}, \dots, G_{b,n}\}$. For the indexes $i \in \vartheta$ its faces agree with those of the smaller cell $\mathfrak{c}_b$, and for any other index $k \notin \vartheta$ in the complement, the face $G_{b,k} = E_k$ is an edge. The secondary cells $\mathfrak{C}_\alpha$ of the coherent mixed subdivision $\Delta_\omega(\mathfrak{F})$ with $\alpha > 0$ are contained without modification.

Because the additional faces E_k of the extended cells are one-dimensional, the classification into mixed and non-mixed cells of the essential family's subdivision $\Delta_\text{M}(\mathfrak{F}_\vartheta)$ is transferred to the extended cells $\mathfrak{C}_{0,b}$ of the composite subdivision $\Delta_\text{M}(\mathfrak{F})$. The latter's domains cover the primary cell $\mathfrak{C}_0$. In secondary cells $\mathfrak{C}_\alpha$ the dimension of the sum of faces $F_{\alpha,i}$ with indices $i \in \vartheta$ is less than that of the affine hull of the support sets of the essential family (17). Since the sum of all faces is n-dimensional, at least one of the faces $F_{\alpha,j}$ in the complement $j \in \bar\vartheta$ must be neither vertex nor edge, so that all of these secondary cells are non-mixed.

Theorem 2.12. *The mapping $\Delta_M(\mathfrak{F}_\vartheta) \mapsto \Delta_M(\mathfrak{F})$ of the essential family's subdivision $\Delta_M(\mathfrak{F}_\vartheta)$ to a composite subdivision $\Delta_M(\mathfrak{F})$ of the complete system preserves the suitability for Macaulay formulas.*

Proof. The Sylvester map $D[\mathfrak{F}]$ falls into several blocks according to the decomposition (23) of its range and image space. As the proof of Proposition 2.10 shows, it acts on subspace $V_1(\mathfrak{C}_0)$.

$$D[\mathfrak{F}]|_{V_1(\mathfrak{C}_0)} = D[\mathfrak{C}_0] \tag{23}$$

This block structure is transferred to the regular matrix $\mathbf{M}$, which is obtained by selecting certain columns [3]. Thereby, the index sets $I_{s,0}$ and $I_{s,1}$ indicate the related basis elements in the subspaces $V_{s,0}(\mathfrak{C}_0)$ respective $V_{s,1}(\mathfrak{C}_0)$. Accordingly, $\tilde I_0$ and $\tilde I_1$ select base vectors in the subspaces $\tilde V_0(\mathfrak{F})$ and $\tilde V_1(\mathfrak{F})$ formed by the secondary cells. Because of (23), the sub-matrices $\mathbf{M}(\tilde I_0, I_{s,1})$ vanish as well as $\mathbf{M}(I_{s,0}, I_{s',1})$ for different index vectors $s \neq s'$. Thus $\mathbf{M}$ is block-wise triangular matrix.

By means of the bijection ϕ_s, defined in (21), the Sylvester map $D[\mathfrak{F}_\vartheta]$ of the essential sub-family's complex is linked to the restricted maps $D[\mathfrak{F}]|_{V_{s,1}(\mathfrak{C}_0)}$, the matrices of all these mappings coinciding. As a consequence, the matrix $\mathbf{M}|_\vartheta$ of the essential subsystem coincide with each $\mathbf{M}(I_{s,0}, I_{s,1})$ so that the determinant of $\mathbf{M}$ factorizes

$$\det \mathbf{M} = \det \mathbf{M}(\tilde I_0, \tilde I_1) \prod_s \det \mathbf{M}(I_{s,0}, I_{s,1}) \tag{24}$$

$$= \det \mathbf{M}(\tilde I_0, \tilde I_1) \det(\mathbf{M}|_\vartheta)^{d_E}. \tag{25}$$

The sub-matrix $\mathbf{E}$ is formed by the columns and rows of $\mathbf{M}$ that belong to base elements whose exponent vectors are located in the shifted domains of the unmixed cells of $\Delta_\text{M}(\mathfrak{F})$. Further, the sub-matrix $\mathbf{E}|_\vartheta$ of $\mathbf{M}|_\vartheta$ is defined accordingly with respect to the subdivision $\Delta_\text{M}(\mathfrak{F}_\vartheta)$, which we assumed to admit the Macaulay formula: Their fraction yields the resultant $\text{Res}(\mathfrak{F}_\vartheta)$ of the essential sub-system.

By definition ϕ_s is a bijection between the polytopes $Q_\vartheta + \delta_\vartheta$ and $Q_\vartheta + \xi_s$. It transfers the categorization of base elements into mixed and non-mixed cells from $V_r(\mathfrak{F}_\vartheta)$ to each subspace $V_{s,r}(\mathfrak{C}_0)$. This defines index sets $J_{s,0}$ and $J_{s,1}$ which select the rows and columns of the matrix $\mathbf{E}$ according to the subspaces $V_{s,r}(\mathfrak{C}_0)$, so that $\mathbf{E}|_\vartheta = \mathbf{E}(J_{s,0}, J_{s,1})$ is valid for each $s \in \{1 \dots d_E\}$. Furthermore, the union of the ϕ_s-images of all lattice points $\boldsymbol{L}_\vartheta \cap (Q_\vartheta + \delta_\vartheta)$ in the shifted Minkowski sum of the essential family's Newton polytopes gives the set of all integer points $\mathbb{Z}^n \cap (\boldsymbol{F}_0 + \delta)$ in the shifted domain of the primary cell $\mathfrak{C}_0$ of the subdivision $\Delta_\omega(\mathfrak{F})$. Since all secondary cells $\mathfrak{C}_\alpha$ are non-mixed, the whole sub-block of $\mathbf{M}$ selected by $\tilde I_0$ and $\tilde I_1$ agrees with the corresponding $\mathbf{E}$-matrix, so that the fraction of their determinants is identical to one. Thus, we obtain from equation (24) and Property 2.10 the resultant of the complete system as a fraction of two determinants.

$$\frac{\det(\mathbf{M})}{\det(\mathbf{E})} = \prod_{s=1}^{d_E} \frac{\det \mathbf{M}(I_{s,0}, I_{s,1})}{\det \mathbf{E}(J_{s,0}, J_{s,1})} = \left(\frac{\det \mathbf{M}|_\vartheta}{\det \mathbf{E}|_\vartheta}\right)^{d_E} \tag{26}$$

$$= \text{Res}(\mathfrak{F}_\vartheta)^{d_E} = \text{Res}(\mathfrak{F}) \tag{27}$$

Hence, the composite subdivision $\Delta_{\mathrm{M}}(\mathfrak{F})$ admits the Macaulay formula, as claimed. □

2.4 The Iteration Step of D'Andrea

[7] discovered an recursive procedure in which the dimension of the problem is reduced at every step. In doing so, he considered lifting functions $\omega_i : \mathcal{A}_i \to \mathbb{Q}$ that vanish except at a single point $\boldsymbol{b}$ in one of the support sets, say the last one $\mathcal{A}_n$. The subdivision of $(Q, \mathfrak{F})$ defined in this way will be designated as $\Delta_{\boldsymbol{b}}(\mathfrak{F})$. Moreover, the function value at the particular point $\boldsymbol{b} \in \mathcal{A}_n$ should be negative, since we look at the lower envelope of the lifted polytope $\hat{Q}$ here, to obtain a subdivision.

$$\begin{aligned} &i < n: \ \omega_i = 0 \\ &i = n: \ \omega_i(\boldsymbol{b}) = -1 \ \text{ and } \omega_i(\mathcal{A}_i \setminus \{\boldsymbol{b}\}) = 0 \end{aligned} \tag{28}$$

To determine the cells of the D'Andrea subdivision $\Delta_{\boldsymbol{b}}(\mathfrak{F})$ generated by these lifting functions, we consider the faces of the lifted polytope $\hat{Q}$ supported by the pairs $\hat{v}_\alpha = (\boldsymbol{v}_\alpha, \hat{v}_{\alpha,n+1}) \in \mathbb{Q}^n \times \mathbb{Q}$. As in section 2.3 the vector $\hat{v}_0 = (0, \ldots, 0, 1)$, in direction of the additional coordinate, defines a particular facet, which forms the *primary cell*:

$$\mathfrak{C}_0 = \{\mathcal{A}_0, \ldots, \mathcal{A}_{n-1}, \{\boldsymbol{b}\}\}. \tag{29}$$

The remaining *secondary cells* are generated by the inward normals $\boldsymbol{v}_\alpha$ on all the facets of the Minkowski sum Q, that do not contain the particular point $\boldsymbol{b}$ of the subdivision, as explained in [7]. For this, we have to assume essential families $\mathfrak{F}$ as input of each iteration step, which however does not mean any restriction, due to Theorem 2.12. Further, we refer the subset of the support $\mathcal{A}_i$ on the face of its convex hull Q_i toward the normal $\boldsymbol{v}_\alpha$ as $\boldsymbol{F}_\alpha(\mathcal{A}_i) \coloneqq \mathrm{face}_{\boldsymbol{v}_\alpha}(Q_i) \cap \mathcal{A}_i$; this allows us to specify the secondary cells of the D'Andrea subdivision:

$$\mathfrak{C}_\alpha = \{\boldsymbol{F}_\alpha(\mathcal{A}_0), \ldots, \boldsymbol{F}_\alpha(\mathcal{A}_{n-1}), \boldsymbol{F}_\alpha(\mathcal{A}_n) \cup \{\boldsymbol{b}\}\}. \tag{30}$$

The key point of this decomposition is the following property, as it allows to reduce the dimension of the problem in each iteration step of the procedure.

Proposition 2.13. *Each cell $\mathfrak{C}_\alpha$ of D'Andrea's decomposition $\Delta_{\boldsymbol{b}}(\mathfrak{F})$ contains either an essential subfamily representing a system whose dimension is smaller than n or no such family.*

Proof. Since we assume that $\mathfrak{F}$ is essential, the family of components of the primary cell $\mathfrak{C}_0$ contains a single vertex, namely the chosen point $\boldsymbol{b} \in \mathcal{A}_n$. The faces $\boldsymbol{F}_\alpha(\mathcal{A}_i)$ and with it the first the first n components $C_{\alpha,i}$ of a secondary cell (30) can be moved into the hyperplane H_α orthogonal to the direction $\boldsymbol{v}_\alpha$ which specifies this cell $\mathfrak{C}_\alpha$. Let us assume, there was an essential family $\mathfrak{C}_{\alpha|\vartheta}$ containing the last support set: $n \in \vartheta$. Since the selected point $\boldsymbol{b}$ does not lie in the face $\boldsymbol{F}_\alpha(\mathcal{A}_n)$, the the component $C_{\alpha,n}$ is never a vertex, so that the intersection $\theta \coloneqq \vartheta \setminus \{n\}$ would not be empty and the rank of the grid of ϑ should exceed that of its subset θ : $\mathrm{rank}(\boldsymbol{L}_\vartheta) > \mathrm{rank}(\boldsymbol{L}_\theta)$. As we supposed the set ϑ indexes an essential family, the inequality $\mathrm{rank}(\boldsymbol{L}_\theta) \geq \#\theta$ is also valid. This, however, implies $\mathrm{rank}(\boldsymbol{L}_\vartheta) > \#\vartheta - 1$, which is a contradiction. Therefore, the last component of a secondary cell cannot be part of the essential family. It thus represents a system with support sets in the hyperplane H_α, whose dimension is smaller that of the surrounding space. □

Hence, if the cell $\mathfrak{C}_\alpha$ contains an essential subfamily $\mathfrak{C}_{\alpha|\vartheta}$, it forms a system with a smaller dimension which, according to the induction hypothesis, has a subdivision $\Delta_{\mathrm{M}}(\mathfrak{C}_{\alpha|\vartheta})$ suitable for Macaulay formulas. According to Theorem 2.12, we can extend this subdivision to the complete cell: $\Delta_{\mathrm{M}}(\mathfrak{C}_{\alpha|\vartheta}) \mapsto \Delta_{\mathrm{M}}(\mathfrak{C}_\alpha)$, so that the Macaulay formula (31) continues to apply.

$$\mathrm{Res}(\mathfrak{C}_\alpha) = \frac{\det \mathbf{M}_\alpha}{\det \mathbf{E}_\alpha} \tag{31}$$

The iterative procedure ends, in the case of one-dimensional systems consisting of two univariate polynomials, a vertex or alternatively, if no unique essential sub-family exists. The first case is already described in [7]. If the essential family is a vertex $\boldsymbol{a} \in C_{\alpha,i}$ in one of the cell components, its resultant is simply a monomial: $\mathrm{Res}(\{\boldsymbol{a}\}) = c_{i,a}$. For such polynomial systems we set the two matrices $\mathbf{M} = c_{i,a}$ and $\mathbf{E} = 1$, so that the Macaulay formula is fulfilled, here. Definition 2.11 can be applied in this particular case as well, resulting in the combined subdivision $\Delta_{\mathrm{M}}(\mathfrak{C}_\alpha)$, which contains a single mixed cell $\mathfrak{C}'_0 = \{E_0, \ldots, \{\boldsymbol{a}\}, \ldots, E_n\}$ with the vertex at the i-th place. Thus, Proposition 2.10 implies $\mathrm{Res}(\mathfrak{C}_\alpha) = (c_{i,a})^{d_E}$ where the exponent d_E is the number of lattice points inside the shifted cell domain of $\mathfrak{C}'_0$, here. It is equal to the volume of the parallelotope $E_{\bar{\vartheta}}$ with $\vartheta = \{i\}$, which in agreement with Theorem 2.4 in [19] corresponds to the mixed volume $d_E = MV(Q_0, \ldots, Q_{i-1}, Q_{i+1}, \ldots, Q_{n-1})$.

If there is no unique essential family contained in the cell $\mathfrak{C}_\alpha$, its resultant is: $\mathrm{Res}(\mathfrak{C}_\alpha) = 1$; in this instance, there are no mixed cells in the combined decomposition $\Delta_{\mathrm{M}}(\mathfrak{C}_\alpha)$, so that both matrices $\mathbf{M}_\alpha$ and $\mathbf{E}_\alpha$ coincide and the Macaulay formula (31) is also valid here.

Equation (12), applied to our order of equations $k = n$, states that the quotient of the minor $\det \boldsymbol{M}$ and the resultant $\mathrm{Res}(\mathfrak{F})$ on the left side of (32) does not depend on coefficients of the last polynomial. It corresponds to its initial term in the direction of $\omega \in \mathbb{Q}^m$. According to Proposition 3.12 in [7] the initial forms of the determinant of the matrices $\mathbf{M}$ and $\mathbf{E}$ factor into the minors $\det \mathbf{M}_\alpha$ and $\det \mathbf{E}_\alpha$ assigned to the cells $\mathfrak{C}_\alpha$. This is consistent with Proposition 2.7, since each of the matrices $\mathbf{M}_\alpha$ and $\mathbf{E}_\alpha$ are submatrices of the Sylvester maps $D[\mathfrak{C}_\alpha]$ resulting from the cells of D'Andreas's subdivision $\Delta_{\boldsymbol{b}}(\mathfrak{F})$.

$$\frac{\det \mathbf{M}}{\mathrm{Res}(\mathfrak{F})} = \mathrm{init}_\omega\left(\frac{\det \mathbf{M}}{\mathrm{Res}(\mathfrak{F})}\right) = \prod_\alpha \frac{\det \mathbf{M}_\alpha}{\mathrm{Res}(\mathfrak{C}_\alpha)} = \cdots \tag{32}$$

As a consequence of Proposition 2.13 and the induction hypothesis there are subdivisions $\Delta_{\mathrm{M}}(\mathfrak{C}_\alpha)$ for each of these cells, so that the Macaulay formula (31) applies. Since the determinant of the submatrix $\mathbf{E}$ does not depend on the coefficient $c_{n,\boldsymbol{b}}$, it agrees with its initial term:

$$\cdots = \prod_\alpha \det \mathbf{E}_\alpha = \mathrm{init}_\omega \det(\mathbf{E}) = \det \mathbf{E}. \tag{33}$$

A comparison of equations (32) and (33) shows that the Macaulay formula (2) is valid for the given n-dimensional system, if we insert the combined subdivisions $\Delta_{\mathrm{M}}(\mathfrak{C}_\alpha)$ into the individual cells $\mathfrak{C}_\alpha$ and thus assemble them to form a subdivision $\Delta_{\mathrm{M}}(\mathfrak{F})$ of the complete family.

3 EXAMPLE

In this section, we present a polynomial system that illustrates the different cases that can occur when reducing the dimension

in each iteration step. For this purpose, the system should not be too small, which means, however, that its resultant contains a large number of terms. Therefore we consider only the denominator $\det \mathbf{E}$ of the Macaulay-style formula here. It can be compared with the corresponding result of the Cayley formula.

$$\begin{aligned}\mathcal{A}_0 &= \{(0,0,0,1),(0,0,1,2),(2,0,1,0),(2,1,0,0)\}\\ \mathcal{A}_1 &= \{(0,2,0,1),(0,2,1,0),(1,0,2,0),(2,1,0,0)\}\\ \mathcal{A}_2 &= \{(0,0,0,0),(0,1,1,0),(1,0,1,0)\}\\ \mathcal{A}_3 &= \{(0,0,1,0),(0,0,1,1),(1,0,0,1)\}\\ \mathcal{A}_4 &= \{(0,1,0,1),(0,1,1,0),(1,0,0,0)\}\end{aligned} \tag{34}$$

In the first iteration step, we select the last element of the support $\mathcal{A}_n$ as particular point $\boldsymbol{b}$ of D'Andrea's subdivision $\Delta_{\boldsymbol{b}}(\mathfrak{F})$. Thus, in this example, we obtain 34 normal vectors $\boldsymbol{v}_\alpha$ defining facets which do not contain this point. For 19 of these directions the polynomial system formed by the cells $\mathfrak{C}_\alpha$ do not contain an essential family, 12 result in a vertex. Only the three secondary cells shown in Table 1 yield a non-trivial polynomial system.

With the vector $\delta = (1/8, 1/3, 1/5, 1/7)$ displacing the Newton polytopes, we obtain a resultant complex $V^\bullet(\mathfrak{F})$ of four terms, having the dimensions: $\dim V_r(\mathfrak{F}) = \{188, 306, 127, 9\}$. The matrix $\mathbf{M}$ in the numerator of Macaulay's formula (2) is correspondingly 188-dimensional, while its sub-matrix $\mathbf{E}$ in the denominator turns out to be of order 109. As stated in equation (33), its determinant is a product of factors determined by the cells $\mathfrak{C}_\alpha$ of subdivision $\Delta_{\boldsymbol{b}}(\mathfrak{F})$.

$$\det \mathbf{E} = \prod_{\alpha=0}^{34} \det \mathbf{E}_\alpha = \boldsymbol{c}_0^{p_0}\boldsymbol{c}_1^{p_1}\boldsymbol{c}_2^{p_2}\boldsymbol{c}_3^{p_3} \sum_{\boldsymbol{q}\in\mathcal{B}} \rho_{\boldsymbol{q}}\, \boldsymbol{c}_0^{q_0}\boldsymbol{c}_1^{q_1}\boldsymbol{c}_2^{q_2}\boldsymbol{c}_3^{q_3} \tag{35}$$

The integer vectors $p_i, q_i \in \mathbb{Z}^{m_i}$ specify the exponents of the coefficients $\boldsymbol{c}_i := (c_{i,a})_{a\in\mathcal{A}_i} \in \mathbb{C}^{m_i}$ of the polynomials (1) in the determinant (35) above; for this, we define the cardinalities $m_i := \#\mathcal{A}_i$.

$$p_0 = (1,6,0,4),\ p_1 = (14,8,5,2),\ p_2 = (18,9,5),\ p_3 = (6,1,2) \tag{36}$$

The minors of the secondary cells yield monomials, only the first factor $\det \mathbf{E}_0$ contributes to the sum in equation (35).

The last column of Table 1 contains the indexes ϑ_α of the respective essential sub-families. By conception, the last support set never contributes to the essential sub-family, as it contains the particular point of the subdivision $\Delta_{\boldsymbol{b}}(\mathfrak{F})$. In the two systems determined by the cells $\mathfrak{C}_2$ and $\mathfrak{C}_3$ it even includes less than 4 elements.

Table 1: These four cells $\mathfrak{C}_\alpha = (C_{\alpha|0}, \cdots, C_{\alpha|3})$ of D'Andrea's subdivision $\Delta_{\boldsymbol{b}}(\mathfrak{F})$ contain essential families ϑ_α with more than one element. Their components are given by the index sets in columns 2 to 6 : $C_{\alpha|i} = \{a_{i,k} \in \mathcal{A}_i : k \in \theta_{\alpha|i}\}$.

α	$\boldsymbol{v}_\alpha$	$\theta_{\alpha\|0}$	$\theta_{\alpha\|1}$	$\theta_{\alpha\|2}$
0	(0,0,0,1)	$\{1,2,3,4\}$	$\{1,2,3,4\}$	$\{1,2,3\}$
1	-(1,1,1,1)	$\{2,3,4\}$	$\{1,2,3,4\}$	$\{2,3\}$
2	-(1,1,1,0)	$\{3,4\}$	$\{2,3,4\}$	$\{2,3\}$
3	(1,0,0,0)	$\{1,2\}$	$\{1,2\}$	$\{1,2\}$

α	$\theta_{\alpha\|3}$	$\theta_{\alpha\|4}$	ϑ_α	
0	$\{1,2,3\}$	$\{3\}$	$\{4\}$	
1	$\{2,3\}$	$\{1,2\}$	$\{0,1,2,3\}$	
2	$\{1,2,3\}$	$\{2,3\}$	$\{0,1,2\}$	
3	$\{1,2\}$	$\{1,2,3\}$	$\{0,1,3\}$	

REFERENCES

[1] Louis J. Billera and Bernd Sturmfels. 1992. Fiber Polytopes. *Annals of Mathematics* 135, 3 (1992), 527–549.

[2] Laurent Buse, Mohamed Elkadi, and André Galligo. 2008. Intersection and self-intersection of surfaces by means of Bezoutian matrices. *Computer Aided Geometric Design* 25, 2 (2008), 53 – 68.

[3] John Canny and Ioannis Emiris. 1993. An efficient algorithm for the sparse mixed resultant. In *Applied Algebra, Algebraic Algorithms and Error-Correcting Codes*, Gérard Cohen, Teo Mora, and Oscar Moreno (Eds.). Springer Berlin Heidelberg, Berlin, Heidelberg, 89–104.

[4] John F. Canny and Ioannis Z. Emiris. 2000. A Subdivision-based Algorithm for the Sparse Resultant. *J. ACM* 47, 3 (May 2000), 417–451.

[5] Marc Chardin. 1993. The Resultant via a Koszul Complex. In *Computational Algebraic Geometry*, Frédéric Eyssette and André Galligo (Eds.). Birkhäuser Boston, Boston, MA, 29–39.

[6] David Cox, John Little, and Donal O'Shea. 2005. *Using Algebraic Geometry*. Springer New York.

[7] Carlos D'Andrea. 2002. Macaulay style formulas for sparse resultants. *Trans. Am. Math. Soc.* 354, 7 (2002), 2595–2629.

[8] Carlos D'Andrea and Alicia Dickenstein. 2001. Explicit formulas for the multivariate resultant. *Journal of Pure and Applied Algebra* 164, 1 (2001), 59 – 86. Effective Methods in Algebraic Geometry.

[9] Carlos D'Andrea, Gabriela Jeronimo, and Martín Sombra. 2020. The Canny-Emiris conjecture for the sparse resultant. arXiv:2004.14622 [math.AC]

[10] Carlos D'Andrea and Martín Sombra. 2015. A Poisson formula for the sparse resultant. *Proc. Lond. Math. Soc. (3)* 110, 4 (2015), 932–964.

[11] Alicia Dickenstein and Ioannis Z. Emiris. 2003. Multihomogeneous resultant formulae by means of complexes. *Journal of Symbolic Computation* 36, 3 (2003), 317 – 342. ISSAC 2002.

[12] Alicia Dickenstein and Ioannis Z. Emiris. 2006. *Solving Polynomial Equations: Foundations, Algorithms, and Applications*. Springer Berlin Heidelberg.

[13] Ioannis Z. Emiris. 2012. A General Solver Based on Sparse Resultants. *CoRR* abs/1201.5810 (2012). arXiv:1201.5810

[14] Ioannis Z. Emiris and John F. Canny. 1995. Efficient Incremental Algorithms for the Sparse Resultant and the Mixed Volume. *Journal of Symbolic Computation* 20, 2 (1995), 117 – 149.

[15] Ioannis Z. Emiris, Vissarion Fisikopoulos, Christos Konaxis, and Luis Peñaranda. 2013. An Oracle-based, Output-sensitive Algorithm for Projections of Resultant Polytopes. *International Journal of Computational Geometry and Applications* 23, 04n05 (2013), 397–423.

[16] Ioannis Z. Emiris and Christos Konaxis. 2011. Single-lifting Macaulay-type formulae of generalized unmixed sparse resultants. *J. Symb. Comput.* 46, 8 (2011), 919–942.

[17] Ioannis Z. Emiris, Christos Konaxis, Ilias S Kotsireas, and Clément Laroche. 2017. Matrix Representations by Means of Interpolation. In *ISSAC '17 - International Symposium on Symbolic and Algebraic Computation*. Kaiserslautern, Germany, 149–156.

[18] Israel M. Gelfand, Mikhail Kapranov, and Andrei Zelevinsky. 1994. *Discriminants, Resultants, and Multidimensional Determinants*. Birkhäuser Boston.

[19] Birkett Huber and Bernd Sturmfels. 1995. A Polyhedral Method for Solving Sparse Polynomial Systems. 64 (10 1995), 1541–1555.

[20] Francis S. Macaulay. 1902. Some Formulæ in Elimination. *Proceedings of the London Mathematical Society* s1-35, 1 (1902), 3–27.

[21] Dinesh Manocha. 1994. Solving Systems of Polynomial Equations. *IEEE Comput. Graph. Appl.* 14, 2 (March 1994), 46–55.

[22] Bernd Sturmfels. 1994. On the Newton Polytope of the Resultant. *Journal of Algebraic Combinatorics* 3, 2 (01 Apr 1994), 207–236.

[23] Bernd Sturmfels. 2002. *Solving Systems of Polynomial Equations*. Number no. 97 in CBMS Regional Conferences Series. American Mathematical Society, Providence Rhode Island.

On the Uniqueness of Simultaneous Rational Function Reconstruction

Eleonora Guerrini, Romain Lebreton, Ilaria Zappatore
guerrini,lebreton,zappatore@lirmm.fr
LIRMM, Université de Montpellier, CNRS
Montpellier, France

ABSTRACT

This paper focuses on the problem of reconstructing a vector of rational functions given some evaluations, or more generally given their remainders modulo different polynomials. The special case of rational functions sharing the same denominator, *a.k.a.* Simultaneous Rational Function Reconstruction (SRFR), has many applications from linear system solving to coding theory, provided that SRFR has a unique solution. The number of unknowns in SRFR is smaller than for a general vector of rational function. This allows one to reduce the number of evaluation points needed to guarantee the existence of a solution, possibly losing its uniqueness. In this work, we prove that uniqueness is guaranteed for a generic instance.

CCS CONCEPTS

• **Mathematics of computing** → **Coding theory**; • **Computing methodologies** → **Algebraic algorithms**; *Linear algebra algorithms.*

ACM Reference Format:
Eleonora Guerrini, Romain Lebreton, Ilaria Zappatore. 2020. On the Uniqueness of Simultaneous Rational Function Reconstruction. In *International Symposium on Symbolic and Algebraic Computation (ISSAC '20), July 20–23, 2020, Kalamata, Greece.* ACM, New York, NY, USA, 8 pages. https://doi.org/10.1145/3373207.3404051

1 INTRODUCTION

Vector Rational Function Reconstruction (VRFR) is the problem of reconstructing a vector $\boldsymbol{v}/\boldsymbol{d} = (v_1/d_1, \ldots, v_n/d_n)$ of rational functions given their remainders $u_i = v_i/d_i \bmod a_i$ and bounds on their degrees. VRFR generalizes *interpolation* problems by taking $a_1 = \cdots = a_n = \prod(x - \alpha_j)$ for some distinct α_j because the modular equations become then equations on evaluations $u_i(\alpha_j) = (v_i/d_i)(\alpha_j)$. *Simultaneous Rational Function Reconstruction* (SRFR) is the particular case of VRFR where all the rational functions share the same denominator (see Section 2.1). The common denominator constraint of SRFR reduces the number of unknowns w.r.t. VRFR, lowering the number of equations (or the number of evaluations in the interpolation case) required to ensure existence of a non-trivial solution. This consideration has interesting consequences for several applications: SRFR appears in polynomial linear system solving via evaluation-interpolation which may be done with less evaluation points. Also, SRFR is related to the decoding of interleaved Reed-Solomon codes and previous consideration can improve the error correction capability of this code (see Section 2.2). However, having a unique solution is fundamental for these applications and there are SRFR instances where the number of equations required to ensure existence does not lead to a unique solution (see Example 2.2). This work studies SRFR instances leading to uniqueness.

A uniqueness result for instances of SRFR coming from polynomial linear system solving can be found in [OS07]. However, this result requires the solution to have a specific degree. We have reasons to believe that we can generalize this result: we conjecture that for almost all $(\boldsymbol{v}, d)$ the SRFR problem admits a unique solution (see Conjecture 2.5).

We can learn more about conditions of uniqueness by looking at results coming from error correcting codes. Interleaved Reed Solomon codes (IRS) can be seen as the evaluation of a vector of polynomials $\boldsymbol{v}$. The problem of decoding IRS codes consists in the reconstruction of the vector of polynomials $\boldsymbol{v}$ given its evaluations, some possibly erroneous. A classic approach to decode IRS codes is the application of SRFR (in its interpolation version) for instances $\boldsymbol{u} = \boldsymbol{v} + \boldsymbol{e}$ where $\boldsymbol{e}$ are the errors. Results from coding theory show that for all $\boldsymbol{v}$ and almost all errors $\boldsymbol{e}$, we get the uniqueness of SRFR for the corresponding instance $\boldsymbol{u}$ (provided that there are not too many errors) [BKY03, BMS04, SSB09]. There is a natural extension of SRFR when errors occur (SRFRwE, see Section 2.2), which can be related to a fractional generalization of IRS [GLZ19, GLZ20]. We conjecture that we can decode almost all codeword $\boldsymbol{v}/d$ and almost all errors $\boldsymbol{e}$ of this fractional code (Conjecture 2.9).

In this paper we present a result which is a step towards Conjectures 2.5 and 2.9. We prove that uniqueness is guaranteed for a generic instance $\boldsymbol{u}$ of SRFR (Theorem 2.4). Our result is valid not only given evaluations, but also in the general context of any moduli a. Our approach to prove Theorem 2.4 is to study the degrees of a relation module. Solutions of SRFR are related to generators of a particular basis of this $\mathbb{K}[x]$-module which have a negative shifted-row degree. Shifts are necessary to integrate degree constraints. We show that for generic instances, there is only one generator with negative row degree, hence uniqueness of SRFR solutions.

Previous works studied generic degrees of different but related modules: *e.g.* for the module of generating polynomials of a scalar matrix sequence [Vil97], for the kernel of a polynomial matrix of specific dimensions [JV05]. Both cases do not consider any shift. The generic degrees also appear as dimensions of blocks of a shifted Hessenberg form [PS07]. However, the link with the degrees of a

ISSAC '20, July 20–23, 2020, Kalamata, Greece

ACM ISBN 978-1-4503-7100-1/20/07…$15.00
https://doi.org/10.1145/3373207.3404051

module is unclear and no shift is discussed (shifted Hessenberg is not related to our shift). We prove our result for any shift and any matrix dimension by adapting some of their techniques to the specific relation module related to SRFR.

In Section 2 we introduce the motivations of our work, starting from the classic SRFR to the extended version with errors. We also show their respective applications in polynomial linear system solving and in error correcting algorithms. In Section 3, we define the algebraic tools that we will use to prove our technical results of the Section 4. In Section 5 we explain how these results are linked to the uniqueness of the solution of SRFR and we finally prove Theorem 2.4 about the generic uniqueness.

2 MOTIVATIONS

2.1 Rational Function Reconstruction

In this section we recall standard definitions and we state our problem, starting from rational function reconstruction and its application to linear algebra. Let $\mathbb{K}$ be a field, $a, u \in \mathbb{K}[x]$ with $\deg(u) < \deg(a)$. The *Rational Function Reconstruction* (RFR) is the problem of reconstructing rational functions $v/d \in \mathbb{K}(x)$ verifying

$$\gcd(d, a) = 1, \frac{v}{d} = u \bmod a, \deg(v) < N, \deg(d) < D. \quad (1)$$

Since the *gcd* equation is not linear, it is customary to focus on the weaker homogeneous linear equation in the polynomial pair (v, d)

$$v = du \bmod a, \deg(v) < N, \deg(d) < D. \quad (2)$$

RFR generalizes many problems including the *Padé approximation* if $a = x^f$ and the *Cauchy interpolation* if $a = \prod_{i=1}^{f}(x - \alpha_i)$, where the α_i are pairwise distinct elements of the field $\mathbb{K}$. The homogeneous linear system related to (2) has $\deg(a)$ equations and $N + D$ unknowns. If $\deg(a) = N + D - 1$, the dimension of the solution space of (2) is at least 1 and it always admits a non-trivial solution. Moreover, such a solution is unique in the sense that all solutions are polynomial multiples of a unique one, $(v_{\min}, d_{\min})$ (see *e.g.* [GG13, Theorem 5.16]). On the other hand, (1) does not always have a solution, but when a solution exists, it is unique and must be $v_{\min}/d_{\min}$, which can be computed using the *Extended Euclidean Algorithm*. Throughout this paper, we will focus on (2).

RFR can be naturally extended to the vector case as follows. Let $a_1, \ldots, a_n \in \mathbb{K}[x]$ with degrees $f_i = \deg(a_i)$ and $\boldsymbol{u} = (u_1, \ldots, u_n) \in \mathbb{K}[x]^n$ where $\deg(u_i) < f_i$. Given $0 < N_i, D_i \leq f_i$, the *Vector Rational Function Reconstruction* (VRFR) is the problem of reconstructing (v_i, d_i) for $1 \leq i \leq n$ such that $v_i = d_i u_i \bmod a_i, \deg(v_i) < N_i, \deg(d_i) < D_i$. We can apply RFR componentwise and so, if $f_i = N_i + D_i - 1$, we can uniquely reconstruct the solution.

SRFR is then the problem of reconstructing a vector of rational functions with the same denominator.

Definition 2.1 (SRFR). Given $\boldsymbol{u} = (u_1, \ldots, u_n) \in \mathbb{K}[x]^n$ where $\deg(u_i) < f_i$, and degree bounds $0 < N_i < f_i$ and $0 < D < \min f_i$, we want to reconstruct the tuple $(\boldsymbol{v}, d) = (v_1, \ldots, v_n, d)$ such that

$$v_i = du_i \bmod a_i, \deg(v_i) < N_i, \deg(d) < D. \quad (3)$$

We denote $\mathcal{S}_{\boldsymbol{u}}$ the set of solutions.

Since solutions of SRFR are solutions of VRFR, SRFR has a unique solution (if it exists) whenever $f_i = N_i + D - 1$ for all i. On the other hand, if the number of equations of (3) is equal to the number of unknowns minus one, that is if

$$\sum_{i=1}^{n} f_i = \sum_{i=1}^{n} N_i + D - 1 \quad (4)$$

then (3) always admits a non-trivial solution. This number of equations is always smaller than before, possibly up to a factor 2. However, the uniqueness is not anymore guaranteed.

Example 2.2. Let $\mathbb{K} = \mathbb{F}_{11}$, $n = 2$, $N_1 = N_2 = 4$, $D = 5$ and $a_1 = a_2 = \prod_{i=1}^{6}(x - 2^i) = x^6 + 6x^5 + 5x^4 + 7x^3 + 2x^2 + 8x + 2$. Let $\boldsymbol{u} = \left(5x^5 + 5x^3 + x^2 + 4x + 4, 8x^5 + 9x^4 + 8x^3 + 8x^2 + 4x + 6\right)$. Then SRFR has two $\mathbb{K}[x]$-linearly independent solutions $(\boldsymbol{v}, d)$: $\left(8x^3 + 5x^2 + x + 6, 7x^3 + 9x^2 + 8x + 9, 7x^3 + 7x^2 + 8x + 9\right)$ and $\left(2x^3 + 2x^2 + 8x, 10x^2 + 10x + 10, 6x^4 + 7x^3 + 8x^2 + 5x + 5\right)$.

Uniqueness is a central property for the applications of SRFR: unique decoding algorithms are essential in error correcting codes, and it is also a widespread condition to use evaluation interpolation techniques in computer algebra. The number of equations which guarantees uniqueness of SRFR has also repercussion on the complexity. Indeed, the complexity of decoding algorithms or evaluation interpolation techniques depends on this number of equations. Since SRFR decreases this number up to a factor 2, this implies a constant factor speedup for applications, like in [OS07].

We denote by s the rank of the $\mathbb{K}[x]$-module spanned by the solutions $\mathcal{S}_{\boldsymbol{u}}$. All solutions can be written as a linear combination $\sum_{i=1}^{s} c_i p_i$ of s polynomials p_i with polynomial coefficients c_i. The case $s = 1$ corresponds to what we call uniqueness of the solution. In [OS07], the authors studied the particular case where $a_1 = \ldots = a_n = a$ and $N_1 = \ldots = N_n = N$. They proved the following,

Theorem 2.3 ([OS07, Theorem 4.2]). *Let k be minimal such that $\deg(a) \geq N + (D-1)/k$, then the rank s of the solution space $\mathcal{S}_{\boldsymbol{u}}$ satisfies $s \leq k$.*

Note that if $k = 1$, the solution is always unique ($s = 1$). This matches the uniqueness condition on $\deg(a)$ of VRFR. On the other hand, if $k = n$ and $\deg(a) \geq N + (D-1)/n$ then $s \leq n$, which does not provide any new information about the solution space. Theorem 2.3 represents a connection between the classic bound $\deg(a) \geq N + D - 1$ which guarantees the uniqueness and the *ideal* one $\deg(a) \geq N + (D-1)/n$ (see (4)), which exploits the common denominator property.

Our main contribution is the following

Theorem 2.4. *If $\sum_{i=1}^{n} f_i = \sum_{i=1}^{n} N_i + D - 1$ then for almost all instances $\boldsymbol{u}$, SRFR admits a unique solution,* i.e. *it has rank $s = 1$.*

Moreover, if $\mathbb{K}$ is a finite field of cardinality q, the proportion of instances leading to non-uniqueness is $\leq (D-1)/q$.

Note that when $D = 1$, rational functions become polynomials and $N_i = f_i$ so that SRFR has always a unique solution $(\boldsymbol{v}, d) = (\boldsymbol{u}, 1)$. Theorem 2.4 will be proved in Section 5. We say that a certain property $\mathcal{P}$ is verified by a generic instance $\boldsymbol{u}$ (or interchangeably for almost all instances $\boldsymbol{u}$) if and only if there exists a nonzero polynomial C such that C does not vanish on $\boldsymbol{u}$ implies that $\mathcal{P}$ is true. In our case, the property is the uniqueness of SRFR and the indeterminates of C are the polynomial coefficients $u_{j,k}$ of the components $u_j = \sum_{k=0}^{f_j - 1} u_{j,k} x^k$ of $\boldsymbol{u}$.

In terms of complexity, [OS07] computes a complete basis of the solution space using $O(nk^{\omega-1}B(\deg(a)))$ operations in $\mathbb{K}$ where $2 \le \omega \le 3$ is the exponent of the matrix multiplication and $B(t) \coloneqq M(t)\log t$ where M is the classic polynomial multiplication arithmetic complexity (see [GG13] for instance). In [RNS16] the complexity was improved: they compute the solution space (in the general case of different moduli a_i) in complexity $O(n^{\omega-1}B(f)\log(f/n)^2)$ where $f = \max_i \deg(a_i)$.

Application to polynomial linear system solving. SRFR has a natural application in linear algebra. Suppose that we want to compute the solution $\boldsymbol{y} = A^{-1}\boldsymbol{b} \in \mathbb{K}(x)$ of a full rank polynomial linear system $A \in \mathbb{K}[x]^{n\times n}$, $\boldsymbol{b} \in \mathbb{K}[x]^{n\times 1}$, from its image modulo a polynomial a. We will refer to this problem as *Polynomial Linear System solving* (PLS). We remark that, by Cramer's rule, $\boldsymbol{y}$ is vector of rational functions with the same denominator: PLS is then a special case of SRFR. In [OS07, Theorem 5.1], the authors proved that the solution space is uniquely generated ($s = 1$) when $\deg(a) \ge N + (D-1)/n$ in the special case of $D = N = n\deg(A)+1$ and $\deg(A) = \deg(b)$. For this purpose, they exploited another bound on the degree of a based on [Cab71].

In view of Theorem 2.4 and as our experiments suggest, we could hope for the following,

Conjecture 2.5. *If* (4) *is satisfied then for almost all* $(\boldsymbol{v}, d)$ *with* $\gcd(d, a_i) = 1$, *SRFR with* $\boldsymbol{u} = \frac{\boldsymbol{v}}{d}$ *as input admits a unique solution.*

Since we have proved the uniqueness for generic instances $\boldsymbol{u}$, it would be sufficient to show the existence of an instance $\boldsymbol{u}$ of the form $\boldsymbol{v}/d$ for any N_i, D, a_i to prove the conjecture.

2.2 Reconstruction with Errors

In this section we introduce the problem of the Simultaneous Rational Function with Errors, *i.e.* SRFR in a scenario where errors may occur in some evaluations [BK14, KPSW17, GLZ19, Per14, GLZ20]. Throughout this section we suppose that $\mathbb{K}$ is a finite field of cardinality q, we fix $\boldsymbol{\alpha} = \{\alpha_1, \dots, \alpha_f\}$ pairwise distinct evaluation points in $\mathbb{K}$ and we consider the polynomial $a = \prod_{i=1}^{f}(x - \alpha_i)$.

Definition 2.6 (SRFR with Errors). Fix $0 < N, D, \varepsilon < f \le q$. An instance of SRFR with errors (SRFRwE) is a matrix $\boldsymbol{\omega} \in \mathbb{K}^{n\times f}$ whose columns are $\boldsymbol{\omega}_j = \boldsymbol{v}(\alpha_j)/d(\alpha_j) + \boldsymbol{e}_j$ for some reduced $\boldsymbol{v}/d \in \mathbb{K}(x)^{n\times 1}$ and some error matrix $\boldsymbol{e}$. The reduced vector must satisfy $\deg(\boldsymbol{v}) < N$, $\deg(d) < D$ and $d(\alpha_i) \ne 0$. The error matrix must have its *error support* $E \coloneqq \{1 \le j \le f \mid \boldsymbol{e}_j \ne \boldsymbol{0}\}$ which satisfies $|E| \le \varepsilon$. Then SRFRwE is the problem of finding a solution $(\boldsymbol{v}, d)$ given an instance $\boldsymbol{\omega}$.

SRFRwE as Reed-Solomon decoding. Observe that if $n = 1$ and $D = 1$, $\boldsymbol{v}/d$ becomes a polynomial. Then SRFRwE is the problem of recovering a polynomial v given evaluations, some of which possibly erroneous; that is decoding an instance of a *Reed-Solomon code*. Its vector generalization, that is $n > 1$ and $D = 1$, coincides with the decoding of an *homogeneous Interleaved Reed-Solomon (IRS) code*. Indeed, an IRS codeword can be seen as the evaluation of a vector of polynomials $\boldsymbol{v}$ on $\boldsymbol{\alpha}$. Thus decoding IRS codes is the problem of recovering $\boldsymbol{v}$ from $\boldsymbol{\omega}_j = \boldsymbol{v}(\alpha_j) + \boldsymbol{e}_j$.

Let us now detail how we can solve SRFRwE using SRFR. We use the same technique of decoding RS and IRS codes [BW86, BKY03, PRN17]. We introduce the *Error Locator Polynomial* $\Lambda = \prod_{j\in E}(x - \alpha_j)$. Its roots are the erroneous evaluations so $\deg(\Lambda) = |E| \le \varepsilon$. We consider the *Lagrangian polynomials* $u_i \in \mathbb{K}[x]$ such that $u_i(\alpha_j) = \omega_{ij}$ for any $1 \le i \le n$. The classic approach is to remark that $(\boldsymbol{\varphi}, \psi) = (\Lambda\boldsymbol{v}, \Lambda d)$ is a solution of $\boldsymbol{\varphi} = \psi\boldsymbol{u} \bmod \prod_{i=1}^{f}(x - \alpha_i)$ such that $\deg(\boldsymbol{\varphi}) < N + \varepsilon$ and $\deg(\psi) < D + \varepsilon$. In this way we reduce SRFRwE to SRFR. If the unique $(\boldsymbol{\varphi}, \psi)$ satisfying latter conditions is $(\Lambda\boldsymbol{v}, \Lambda d)$, then we can reconstruct $(\boldsymbol{v}, d)$ and solve SRFRwE. Uniqueness can be obtained by taking VRFR constraints $f = (N + \varepsilon) + (D + \varepsilon) - 1 = N + D + 2\varepsilon - 1$ [BK14, KPSW17].

It is possible to reduce the number of evaluations w.r.t. the maximal number of errors ε in the setting of IRS decoding ($D = 1$).

Theorem 2.7 ([BKY03, BMS04, SSB09]). *Fix* $0 < N, \varepsilon < f \le q$ *and* E *such that* $|E| \le \varepsilon$. *If* $f = N + \varepsilon + \varepsilon/n$, *then for all* $(\boldsymbol{v}, 1)$ *and almost all error matrices* $\boldsymbol{e}$ *of support* E, *SRFRwE admits a unique solution on the instance* $\boldsymbol{\omega}$ *where* $\boldsymbol{\omega}_j = \boldsymbol{v}(\alpha_j)/d(\alpha_j) + \boldsymbol{e}_j$.

We proved a similar result in the rational function case,

Theorem 2.8 ([GLZ19, GLZ20]). *Fix* $0 < N, D, \varepsilon < f \le q$ *and* E *such that* $|E| \le \varepsilon$. *If* $f = N + D - 1 + \varepsilon + \varepsilon/n$, *then for all* $(\boldsymbol{v}, d)$ *and almost all error matrices* $\boldsymbol{e}$ *of support* E, *SRFRwE admits a unique solution on the instance* $\boldsymbol{\omega}$ *where* $\boldsymbol{\omega}_j = \boldsymbol{v}(\alpha_j)/d(\alpha_j) + \boldsymbol{e}_j$.

Since the problem of SRFRwE reduces to SRFR, there always exists a non-trivial $(\boldsymbol{\varphi}, \psi)$ whenever $f = N + \varepsilon + (D + \varepsilon - 1)/n$. Our ideal result would be to prove a uniqueness result also in this case. Our experiments suggest the following,

Conjecture 2.9. *Fix* $0 < N, D, \varepsilon < f \le q$ *and* E *such that* $|E| \le \varepsilon$. *If* $f = N + \varepsilon + (D + \varepsilon - 1)/n$, *then for almost all* $(\boldsymbol{v}, d)$ *and almost all error matrices* $\boldsymbol{e}$ *of support* E, *SRFRwE admits a unique solution on the instance* $\boldsymbol{\omega}$ *where* $\boldsymbol{\omega}_j = \boldsymbol{v}(\alpha_j)/d(\alpha_j) + \boldsymbol{e}_j$.

Note that Conjecture 2.9 is for almost all fractions $(\boldsymbol{v}, d)$ whereas Theorems 2.7 and 2.8 are for all fractions. This difference is due to Example 2.2, which shows that we can not have uniqueness for all instances $\boldsymbol{u}$ of the form $\boldsymbol{u} = \boldsymbol{v}/d$ when $f = N + (D-1)/n$. This latter number of evaluations matches the one of Conjecture 2.9 in the situation without errors $\varepsilon = 0$. Remark that this obstruction does not affect Theorems 2.7 and 2.8 because their number of evaluations f becomes $N + D - 1$ when $\varepsilon = 0$.

Our result Theorem 2.4 is a first step towards Conjecture 2.9: Since uniqueness of SRFR is true for generic instance $\boldsymbol{\omega}$, it remains to prove the existence of an instance of the form $\boldsymbol{\omega}_j = \boldsymbol{v}(\alpha_j)/d(\alpha_j) + \boldsymbol{e}_j$ for any N, D, ε, E to prove the conjecture.

Polynomial linear system solving with errors. SRFRwE was first introduced by [BK14] as a special case of Polynomial Linear System Solving with Errors (PLSwE), that we now introduce. Suppose that we want to compute the unique solution $\boldsymbol{y} = \boldsymbol{v}/d = A^{-1}\boldsymbol{b} \in \mathbb{K}[x]^{n\times n}$ of a PLS in a scenario where some errors occur [BK14, KPSW17, GLZ19]. Suppose a black box gives us solutions $\boldsymbol{y}_i = A(\alpha_i)^{-1}\boldsymbol{b}(\alpha_i)$ of evaluated systems, where α_i are f distinct evaluations points such that $d(\alpha_i) \ne 0$. This black box could make some errors in the computations; an evaluation α_j is *erroneous* if $\boldsymbol{y}_j \ne \boldsymbol{v}(\alpha_j)/d(\alpha_j)$ and we denote by $E \coloneqq \{j \mid \boldsymbol{y}_j \ne \boldsymbol{v}(\alpha_j)/d(\alpha_j)\}$ the set of erroneous positions. We observe that if $j \in E$, then there exists a nonzero $\boldsymbol{e}_j \in \mathbb{K}^{n\times f}$ such that $\boldsymbol{y}_j = \boldsymbol{v}(\alpha_j)/d(\alpha_j) + \boldsymbol{e}_j$.

Hence, this problem is a special case of SRFRwE. Here we want to reconstruct a vector of rational functions which is a solution of a polynomial linear system. Therefore, all the results about uniqueness of the previous sections hold. Finally, we mention that there exists another bound on f which guarantees the uniqueness in the context of PLSwE; this bound depends on the degree of the polynomial matrix A and the vector $\boldsymbol{b}$ [KPSW17].

3 PRELIMINARIES

In this section we will give some definitions and set out the notation that we will use throughout this paper. We refer to [Nei16] for proofs of lemmas, examples and historical references.

3.1 Row degrees of a $\mathbb{K}[x]$-module

Let $\mathbb{K}$ be a field and $\mathbb{K}[x]$ its ring of polynomials. We start by defining the row degree of a vector, then of a matrix. Let $\boldsymbol{p} = (p_1, \ldots, p_\nu) \in \mathbb{K}[x]^\nu = \mathbb{K}[x]^{1\times\nu}$ and $\boldsymbol{s} = (s_1, \ldots, s_\nu) \in \mathbb{Z}^\nu$ a shift.

Definition 3.1 (Shifted row degree). Let $r_i = \deg(p_i) + s_i$ for $1 \le i \le \nu$. The $\boldsymbol{s}$*-row degree* of $\boldsymbol{p}$ is $\operatorname{rdeg}_{\boldsymbol{s}}(\boldsymbol{p}) = \max r_i$. We also denote $\boldsymbol{p} = ([r_1]_{s_1}, \ldots, [r_\nu]_{s_\nu})$ a vector of polynomials with these degrees.

We can extend this definition to polynomial matrices. In fact, let $P \in \mathbb{K}[x]^{\rho\times\nu}$ be a polynomial matrix, with $\rho \le \nu$. Let $P_{i,*}$ be the i-th row of P for $1 \le i \le \rho$. We can define the $\boldsymbol{s}$*-row degrees* of the matrix P as $\operatorname{rdeg}_{\boldsymbol{s}}(P) := (r_1, \ldots, r_\rho)$ where $r_i := \operatorname{rdeg}_{\boldsymbol{s}}(P_{i,*})$.

Let $\mathcal{N}$ be a $\mathbb{K}[x]$-submodule of $\mathbb{K}[x]^\nu = \mathbb{K}[x]^{1\times\nu}$. Since $\mathbb{K}[x]$ is a principal ideal domain, $\mathcal{N}$ is *free* of *rank* $\rho := \operatorname{rank}(\mathcal{N})$ less than ν [DF03, Section 12.1, Theorem 4]. Hence, we can consider a *basis* $P \in \mathbb{K}[x]^{\rho\times\nu}$, *i.e.* a full rank polynomial matrix, such that $\mathcal{N} = \mathbb{K}[x]^{1\times\rho}P = \{\boldsymbol{\lambda}P \mid \boldsymbol{\lambda} \in \mathbb{K}[x]^{1\times\rho}\}$.

Our goal is to define a notion of row degrees of $\mathcal{N}$ in order to study later the $\mathbb{K}$-vector space $\mathcal{N}_{<r} := \left\{\boldsymbol{p} \in \mathcal{N} \;\middle|\; \operatorname{rdeg}_{\boldsymbol{s}}(\boldsymbol{p}) < r\right\}$ for some $r \in \mathbb{Z}$. Different bases P of $\mathcal{N}$ have different $\boldsymbol{s}$-row degrees so we need more definitions. We start with row reduced bases.

Let $\boldsymbol{t} = (t_1, \ldots, t_\nu) \in \mathbb{Z}^\nu$. We denote by $X^{\boldsymbol{t}}$ the diagonal matrix whose entries are $x^{t_1}, \ldots, x^{t_\nu}$. The $\boldsymbol{s}$*-leading matrix* $LM_{\boldsymbol{s}}(P)$ of P is a matrix in $\mathbb{K}^{\rho\times\nu}$, whose entries are the coefficient of degree zero of $X^{-\operatorname{rdeg}_{\boldsymbol{s}}(P)}PX^{\boldsymbol{s}}$. A basis $P \in \mathbb{K}[x]^{\rho\times\nu}$ of $\mathcal{N}$ is $\boldsymbol{s}$*-row reduced* (shortly $\boldsymbol{s}$-reduced) if $LM_{\boldsymbol{s}}(P)$ has full rank. This definition is equivalent to [Nei16, Definition 1.10], which implies that all $\boldsymbol{s}$-reduced basis of $\mathcal{N}$ have the same row degrees, up to permutation. We now focus on the following crucial property.

Lemma 3.2 (Predictable degree property). *P is $\boldsymbol{s}$-reduced if and only if for all $\boldsymbol{\lambda} = (\lambda_1, \ldots, \lambda_\rho) \in \mathbb{K}[x]^{1\times\rho}$,*

$$rdeg_{\boldsymbol{s}}(\boldsymbol{\lambda}P) = \max_{1\le i\le\rho}(\deg(\lambda_i) + rdeg_{\boldsymbol{s}}(P_{i,*})) = rdeg_{rdeg_{\boldsymbol{s}}(P)}(\boldsymbol{\lambda}).$$

The proof of this classic proposition can be found for instance in [Nei16, Theorem 1.11]. This latter proposition is useful because it implies that $\dim_{\mathbb{K}} \mathcal{N}_{<r} = \sum_{\{i \mid r_i < r\}}(r - r_i)$ where $(r_1, \ldots, r_\rho)$ are the $\boldsymbol{s}$-row degrees of any $\boldsymbol{s}$-reduced basis of $\mathcal{N}$.

Since we will need to define the $\boldsymbol{s}$-row degrees of $\mathcal{N}$ uniquely, not just up to permutation, we need to introduce ordered weak Popov form, which relies on the notion of pivot. The $\boldsymbol{s}$*-pivot index* of $\boldsymbol{p} \in \mathbb{K}[x]^{1\times\nu}$ is $\max\{j \mid \operatorname{rdeg}_{\boldsymbol{s}}(\boldsymbol{p}) = \deg(p_j) + s_j\}$. Moreover the corresponding p_j is the $\boldsymbol{s}$*-pivot entry* and $\deg(p_j)$ is the $\boldsymbol{s}$*-pivot degree* of $\boldsymbol{p}$. We naturally extend the notion of pivot to polynomial matrices. A basis P of $\mathcal{N}$ is in $\boldsymbol{s}$*-weak Popov form* if the $\boldsymbol{s}$-pivot indices of its rows are pairwise distinct. On the other hand, it is in $\boldsymbol{s}$*-ordered weak Popov form* if the sequence of the $\boldsymbol{s}$-pivot indices of its rows is strictly increasing. A basis in $\boldsymbol{s}$-weak Popov form is $\boldsymbol{s}$-reduced. Indeed, $LM_{\boldsymbol{s}}(P)$ becomes, up to row permutation, a lower triangular matrix with non-zero entries on the diagonal. Hence it is full-rank.

Assume from now on that $\mathcal{N}$ is a submodule of $\mathbb{K}[x]^\nu$ of rank ν and that P is a basis of $\mathcal{N}$ in $\boldsymbol{s}$-ordered weak Popov form. Then its pivot indices must be $\{1, \ldots, \nu\}$. Weak Popov bases have a strong degree minimality property, stated in the following lemma.

Lemma 3.3 ([Nei16, Lemma 1.17]). *Let $\boldsymbol{s} \in \mathbb{Z}^\nu$, P be a basis of $\mathcal{N}$ in $\boldsymbol{s}$-weak Popov form with $\boldsymbol{s}$-pivot degrees $(d_1, \ldots, d_\nu)$. Let $\boldsymbol{p} \in \mathcal{N}$ whose pivot index is $1 \le i \le \nu$. Then the $\boldsymbol{s}$-pivot degree of $\boldsymbol{p}$ is $\ge d_i$ or equivalently $rdeg_{\boldsymbol{s}}(\boldsymbol{p}) \ge rdeg_{\boldsymbol{s}}(P_{i,*})$.*

As it turns out, ordered weak Popov bases are reduced bases for which the $\boldsymbol{s}$-row degrees is unique. The following lemma is a consequence of Lemma 3.3.

Lemma 3.4 ([Nei16, Lemma 1.25]). *Let $\boldsymbol{s} \in \mathbb{Z}^\nu$ and assume $\mathcal{N}$ is a submodule of $\mathbb{K}[x]^\nu$ of rank ν. Let P and Q be two bases of $\mathcal{N}$ in $\boldsymbol{s}$-ordered weak Popov form. Then P and Q have the same $\boldsymbol{s}$-row degrees and $\boldsymbol{s}$-pivot degrees.*

3.2 Link between pivot and leading term

In this section, we will focus on the relation between pivots of weak Popov bases and leading terms w.r.t. a specific monomial order, as in Gröbner basis theory (see for instance [CLO98]).

Let $\mathbb{K}[\boldsymbol{x}] := \mathbb{K}[x_1, \ldots, x_n]$ be the ring of multivariate polynomials. Recall that a *monomial in* $\mathbb{K}[\boldsymbol{x}]$ is a product of powers of the indeterminates $\boldsymbol{x}^{\boldsymbol{i}} := x_1^{i_1} \cdots x_n^{i_n}$ for some $\boldsymbol{i} := (i_1, \ldots, i_n) \in \mathbb{N}^n$. On the other hand, a *monomial in* $\mathbb{K}[\boldsymbol{x}]^n$ is $\boldsymbol{x}^{\boldsymbol{i}}\boldsymbol{\varepsilon}_j$, where $\boldsymbol{\varepsilon}_1, \ldots, \boldsymbol{\varepsilon}_n$ is the canonical basis of the $\mathbb{K}[\boldsymbol{x}]$-module $\mathbb{K}[\boldsymbol{x}]^n$.

A *monomial order* on $\mathbb{K}[\boldsymbol{x}]^n$ is a *total order* $\prec$ on the monomials of $\mathbb{K}[\boldsymbol{x}]^n$ such that, for any monomials $\varphi\boldsymbol{\varepsilon}_i, \psi\boldsymbol{\varepsilon}_j \in \mathbb{K}[\boldsymbol{x}]^n$ and any monomial $\tau \ne 1$, $\tau \in \mathbb{K}[\boldsymbol{x}]$, $\varphi\boldsymbol{\varepsilon}_i \prec \psi\boldsymbol{\varepsilon}_j \implies \varphi\boldsymbol{\varepsilon}_i \prec \tau\varphi\boldsymbol{\varepsilon}_i \prec \tau\psi\boldsymbol{\varepsilon}_j$. Given a monomial order $\prec$ on $\mathbb{K}[\boldsymbol{x}]^n$ and $f \in \mathbb{K}[\boldsymbol{x}]^n$, the $\prec$*-initial term* $in_\prec(f)$ of f is the term of f whose monomial is the greatest with respect to the order $\prec$. We remark that in the case of $\mathbb{K}[x]$, the only monomial order is the natural degree order $x^a \prec x^b \Leftrightarrow a < b$.

We now define the *shifted* $\boldsymbol{s}$*-TOP order* (Term Over Position) on $\mathbb{K}[\boldsymbol{x}]^n$ related to a monomial order $\prec$ on $\mathbb{K}[\boldsymbol{x}]$ and a choice of *shifting monomials* $\gamma_1, \ldots, \gamma_n$ in $\mathbb{K}[\boldsymbol{x}]$:

$$\varphi\boldsymbol{\varepsilon}_i \prec_{\boldsymbol{s}-TOP} \psi\boldsymbol{\varepsilon}_j \iff (\varphi\gamma_i \prec \psi\gamma_j) \text{ or } (\varphi\gamma_i = \psi\gamma_j \text{ and } i < j)$$

for any pairs of monomials $\varphi\boldsymbol{\varepsilon}_i$ and $\psi\boldsymbol{\varepsilon}_j$ of $\mathbb{K}[\boldsymbol{x}]^n$. In the univariate case $\mathbb{K}[x]^n$, the only monomial order $\prec$ on $\mathbb{K}[x]$ is the *natural* one and the shifting monomials are $\gamma_i = x^{s_i}$ for $\boldsymbol{s} = (s_1, \ldots, s_n) \in \mathbb{N}^n$, so that the $\boldsymbol{s}$-TOP order on $\mathbb{K}[x]^n$ is

$$x^a\boldsymbol{\varepsilon}_i \prec_{\boldsymbol{s}\text{-TOP}} x^b\boldsymbol{\varepsilon}_j \iff (a + s_i, i) \prec_{lex} (b + s_j, j). \tag{5}$$

We can now state the link between this monomial order and the pivot's definition: let $\boldsymbol{p} \in \mathbb{K}[x]^{1\times n}$ and write $in_{\prec_{\boldsymbol{s}\text{-TOP}}}(\boldsymbol{p}) = \alpha x^d\boldsymbol{\varepsilon}_i$, then the $\boldsymbol{s}$-pivot index, entry, and degree are respectively i, p_i and d. This will be useful later on, in *e.g.* Proposition 4.3.

4 ROW DEGREE OF THE RELATION MODULE

Fix $m \geq n \geq 0$, and $M \in \mathbb{K}[x]^{m\times n}$. We consider a $\mathbb{K}[x]$-submodule $\mathcal{M}$ of $\mathbb{K}[x]^n$. We define the $\mathbb{K}[x]-$*module homomorphism*

$$\varphi_{\hat{M}}: \begin{array}{ccc} \mathbb{K}[x]^m & \longrightarrow & \mathbb{K}[x]^n/\mathcal{M} \\ \boldsymbol{p} & \longmapsto & \boldsymbol{p}M \end{array}.$$

Set $\mathcal{A}_{\mathcal{M},M} := \ker(\varphi_{\hat{M}})$ to get the injection

$$\varphi_M : \mathbb{K}[x]^m/\mathcal{A}_{\mathcal{M},M} \hookrightarrow \mathbb{K}[x]^n/\mathcal{M}.$$

We call $\mathcal{A}_{\mathcal{M},M}$ the *relation module* because $p \in \mathcal{A}_{\mathcal{M},M} \Leftrightarrow \varphi_M(\boldsymbol{p}) = \boldsymbol{p}M = 0 \bmod \mathcal{M}$, *i.e.* $\boldsymbol{p}$ is a relation between rows of M.

Let $\boldsymbol{\varepsilon}_1, \ldots, \boldsymbol{\varepsilon}_m$ be the *canonical basis* of $\mathbb{K}[x]^m$, $\boldsymbol{\varepsilon}'_1, \ldots, \boldsymbol{\varepsilon}'_n$ the *canonical basis* of $\mathbb{K}[x]^n$ and $\boldsymbol{e}_i = \boldsymbol{\varepsilon}_i \bmod \mathbb{K}[x]^m/A_{\mathcal{M},M}$ for $1 \leq i \leq m$.

Remark 4.1. We observe that by the *Invariant Factor Form of modules over Principal Ideal Domains* (*cf.* [DF03, Theorem 4, Chapter 12]), $\mathcal{K} := \mathbb{K}[x]^n/\mathcal{M} \simeq \mathbb{K}[x]^n/\langle a_i\boldsymbol{\varepsilon}'_i\rangle_{1\leq i\leq n}$ for nonzero $a_i \in \mathbb{K}[x]$ such that $a_n|a_{n-1}|\ldots|a_1$. The polynomials a_i are the *invariants* of the module $\mathcal{M}$. We also denote $f_i := \deg(a_i)$ and we observe that $f_1 \geq f_2 \geq \ldots \geq f_n$.

From now on we will assume that $\mathcal{M} = \langle a_i\boldsymbol{\varepsilon}'_i\rangle_{1\leq i\leq n}$. It means that any $\boldsymbol{q} \in \mathcal{K}$ can be seen as $(q_1 \bmod a_1, \ldots, q_n \bmod a_n)$. Using the result of Lemma 3.4, we can define the row and pivot degrees of the relation module $\mathcal{A}_{\mathcal{M},M}$.

Definition 4.2 (Row and pivot degrees of the relation module). Let $\boldsymbol{s} \in \mathbb{Z}^m$ be a shift and P be any basis of $\mathcal{A}_{\mathcal{M},M}$ in ordered weak Popov form. The $\boldsymbol{s}$-row degrees of the relation module $\mathcal{A}_{\mathcal{M},M}$ are $\boldsymbol{\rho} := \mathrm{rdeg}_{\boldsymbol{s}}(P) = (\rho_1, \ldots, \rho_m)$ and the $\boldsymbol{s}$-pivot degrees are $\boldsymbol{\delta} := (\delta_1, \ldots, \delta_m)$ where $\delta_i = \rho_i - s_i$.

Throughout this paper we will also denote $\boldsymbol{\rho}_M$ and $\boldsymbol{\delta}_M$ when we want to stress out the matrix dependency.

4.1 Row degrees as row rank profile

In this section, we will see that the row degrees of the relation module can be deduced from the row rank profile of a matrix associated to $\varphi_{\hat{M}}$. We start by associating the pivot degree of $\boldsymbol{p} \in \mathcal{A}_{\mathcal{M},M}$ to linear dependency relation.

Proposition 4.3. *There exists $\boldsymbol{p} \in \mathcal{A}_{\mathcal{M},M}$ with $\boldsymbol{s}$-pivot index i and $\boldsymbol{s}$-pivot degree d if and only if $x^d\boldsymbol{e}_i \in B_M^{\prec x^d\boldsymbol{\varepsilon}_i}$ where $B_M^{\prec x^d\boldsymbol{\varepsilon}_i} := \langle x^n\boldsymbol{e}_j \mid x^n\boldsymbol{\varepsilon}_j \prec_{s-TOP} x^d\boldsymbol{\varepsilon}_i\rangle$.*

Proof. Fix $i, d \in \mathbb{N}$ and let $\boldsymbol{p} \in \mathbb{K}[x]^n$ with $\boldsymbol{s}$-pivot index i and $\boldsymbol{s}$-pivot degree d, so $r := \mathrm{rdeg}_{\boldsymbol{s}}(\boldsymbol{p}) = d + s_i$. Then $\boldsymbol{p} = ([\leq r]_{s_1}, \ldots, [\leq r]_{s_{i-1}}, [r]_{s_i}, [< r]_{s_{i+1}}, \ldots, [< r]_{s_m})$ (see Definition 3.1) and we can write $\boldsymbol{p} = cx^d\boldsymbol{\varepsilon}_i + \boldsymbol{p}'$ where $c \in \mathbb{K}^*$ and $\boldsymbol{p}' = ([\leq r]_{s_1}, \ldots, [\leq r]_{s_{i-1}}, [< r]_{s_i}, [< r]_{s_{i+1}}, \ldots, [< r]_{s_m})$. So $\boldsymbol{p} \in \mathcal{A}_{\mathcal{M},M}$ has s-pivot index i and degree $d \Leftrightarrow x^d\boldsymbol{\varepsilon}_i = -1/c\ \boldsymbol{p}' \bmod \mathcal{A}_{\mathcal{M},M} \Leftrightarrow$

$$x^d\boldsymbol{e}_i \in \left\langle x^n\boldsymbol{e}_j \;\middle|\; \begin{array}{ll} n + s_j \leq d + s_i, & \text{for } 1 \leq j \leq i-1 \\ n + s_j < d + s_i, & \text{for } i \leq j \leq m \end{array} \right\rangle = B_M^{\prec x^d\boldsymbol{\varepsilon}_i}. \quad \square$$

Theorem 4.4. *Let $\boldsymbol{\delta}$ be the $\boldsymbol{s}$-pivot degrees of the relation module $\mathcal{A}_{\mathcal{M},M}$. Then $\delta_j = min\{d \mid x^d\boldsymbol{e}_j \in B_M^{\prec x^d\boldsymbol{\varepsilon}_j}\}$ for any $1 \leq j \leq m$.*

Proof. Fix $1 \leq j \leq m$. During this proof we denote $\overline{\delta}_j := min\{d \mid x^d\boldsymbol{e}_j \in B_M^{\prec x^d\boldsymbol{\varepsilon}_j}\}$. We want to prove that $\delta_j = \overline{\delta}_j$. Recall that by Proposition 4.3, $x^{\delta_j}\boldsymbol{e}_j \in B_M^{\prec x^{\delta_j}\boldsymbol{\varepsilon}_j}$. Hence, by the minimality of $\overline{\delta}_j$, $\delta_j \geq \overline{\delta}_j$. On the other hand, $x^{\overline{\delta}_j}\boldsymbol{e}_j \in B_M^{\prec x^{\overline{\delta}_j}\boldsymbol{\varepsilon}_j}$ so by Proposition 4.3 there exists $\boldsymbol{p} \in \mathcal{A}_{\mathcal{M},M}$ of $\boldsymbol{s}$-pivot index j and degree $\overline{\delta}_j$. Finally, by Lemma 3.3 we can conclude that $\overline{\delta}_j \geq \delta_j$. $\square$

We now define the *ordered matrix* O_M as the matrix of $\hat{\varphi}_M$ w.r.t. particular $\mathbb{K}$-vector space bases: the rows of O_M from top to bottom are the monomials of $\mathbb{K}[x]^m$ sorted increasingly for the $\prec_{s-TOP}$ order (see (5)). The columns of O_M are written *w.r.t.* the basis $\{x^i\boldsymbol{\varepsilon}'_j\}_{\substack{1\leq j\leq n \\ 0\leq i<f_j}}$ of $\mathbb{K}[x]^n/\mathcal{M}$. Therefore, O_M has finite rank $\mathrm{rank}(O_M) = \mathrm{rank}(\hat{\varphi}_M) = \mathrm{rank}(\varphi_M)$, infinite number of rows and $(\sum_{i=1}^n f_i) = \dim_{\mathbb{K}}(\mathbb{K}[x]^n/\mathcal{M})$ columns.

Monomial row rank profile. Our goal is to relate the row rank profile of O_M to the row degrees of the relation module. The classic definition of row rank profile of a rank r polynomial matrix is the lexicographically smallest sequence of r indices of linearly independent rows (*cf.* [DPS15] for instance). Since the rows of our ordered matrix O_M correspond to monomials, we will transpose the previous definition to monomials instead of indices.

Let Mon_r be the sets of r monomials of $\mathbb{K}[x]^m$. We define the lexicographical ordering on Mon_r by comparing lexicographically the sorted monomials for $\prec_{s-TOP}$. In detail, $\mathcal{F} \prec_{lex} \mathcal{F}'$ iff there exists $1 \leq t \leq r$ s.t. $x^{i_l}\boldsymbol{\varepsilon}_{j_l} = x^{u_l}\boldsymbol{\varepsilon}_{v_l}$ for $l < t$ and $x^{i_t}\boldsymbol{\varepsilon}_{j_t} \prec_{s-TOP} x^{u_t}\boldsymbol{\varepsilon}_{v_t}$ where $\mathcal{F} = \{x^{i_l}\boldsymbol{\varepsilon}_{j_l}\}_{1\leq l\leq r}$ and $\mathcal{F}' = \{x^{u_l}\boldsymbol{\varepsilon}_{v_l}\}_{1\leq l\leq r}$ and both $\{x^{i_l}\boldsymbol{\varepsilon}_{j_l}\}$ and $\{x^{u_l}\boldsymbol{\varepsilon}_{v_l}\}$ are increasing for the $\prec_{s-TOP}$ order.

We will use this lexicographic order on monomials to define the row rank profile of O_M. Let $r = \mathrm{rank}(O_M)$.

Definition 4.5 (Row rank profile). For any matrix $M \in \mathbb{K}[x]^{m\times n}$, we define the *row rank profile* of O_M (shortly RRP_M) as the family of monomials of $\mathbb{K}[x]^m$ defined by $RRP_M := min_{\prec_{lex}}\mathcal{P}_M$ where

$$\mathcal{P}_M := \left\{\mathcal{F} \in \mathsf{Mon}_r \;\middle|\; \{mM\}_{m\in\mathcal{F}} \text{ are linearly independent in } \mathcal{K}\right\}.$$

We now introduce a particular family of monomials, that we will frequently use: we will denote $\mathcal{F}_{\boldsymbol{d}} := \{x^i\boldsymbol{\varepsilon}_j\}_{\substack{i<d_j \\ 1\leq j\leq m}}$ for any $\boldsymbol{d} = (d_1, \ldots, d_m) \in \mathbb{N}^m$.

This family allows us to finally relate the row rank profile of O_M to the row degrees of the relation module.

Proposition 4.6. *The row rank profile of the ordered matrix O_M is given by the pivot degrees $\boldsymbol{\delta}_M$ of the relation module $\mathcal{A}_{\mathcal{M},M}$,* i.e. *$RRP_M = \mathcal{F}_{\boldsymbol{\delta}_M}$.*

Proof. We fix the matrix M in order to simplify notations. We define $\delta'_j = min\left\{\delta \mid x^\delta\boldsymbol{\varepsilon}_j \notin RRP\right\}$ and $\boldsymbol{\delta}' = (\delta'_1, \ldots, \delta'_m)$. By properties of row rank profile, we have that $x^{\delta_j}\boldsymbol{e}_j \in B^{\prec x^{\delta_j}\boldsymbol{\varepsilon}_j}$ (otherwise we could create a smaller family of linearly independent monomial with $x^{\delta_j}\boldsymbol{e}_j$). Using Theorem 4.4, we deduce that $\delta'_j \geq \delta_j$. Therefore $\mathcal{F}_{\boldsymbol{\delta}} \subset \mathcal{F}_{\boldsymbol{\delta}'} \subset RRP$. Since the families of monomials $\mathcal{F}_{\boldsymbol{\delta}}$ and RRP have the same cardinality $r = \mathrm{rank}(O_M)$, they are equal so $\mathcal{F}_{\boldsymbol{\delta}} = RRP$. $\square$

4.2 Constraints on relation's row degrees

We will now focus on integer tuples $\boldsymbol{\delta}_M$ which can be achieved. For this matter, in the light of Proposition 4.6, we need to understand which families $\mathcal{F}_{\boldsymbol{d}}$ of monomials can be linearly independent in the ordered matrix, *i.e.* belong to $\mathcal{P}_M$ (see Definition 4.5).

Recall that $\mathcal{K} = \mathbb{K}[x]^n/\mathcal{M} = \mathbb{K}[x]^n/\langle a_i\boldsymbol{\varepsilon}'_i\rangle_{1\le i\le n}$ and $f_i = \deg(a_i)$ are non-increasing as in Remark 4.1. Recall also from Definition 4.5 that $\mathcal{P}_M$ is the set of families $\mathcal{F}$ of r monomials in $\mathbb{K}[x]^m$ such that $\{mM\}_{m\in\mathcal{F}}$ are linearly independent in $\mathbb{K}[x]^n/\mathcal{M}$.

Theorem 4.7. *Let $\boldsymbol{d} \in \mathbb{N}^m$ be non-increasing. We can extend $\boldsymbol{f} \in \mathbb{N}^m$ by $f_{n+1} = \ldots = f_m = 0$. Then $\exists M \in \mathbb{K}[x]^{m\times n}$ such that $\mathcal{F}_{\boldsymbol{d}} \in \mathcal{P}_M$ if and only if $\sum_{i=1}^{l} d_i \le \sum_{i=1}^{l} f_i$ for all $1 \le l \le m$.*

The non-increasing property of $\boldsymbol{d}$ can be lifted: let $\boldsymbol{d}$ be non-increasing and $\boldsymbol{d}'$ be any permutation of $\boldsymbol{d}$. Then $\exists M \in \mathbb{K}[x]^{m\times n}$ such that $\mathcal{F}_{\boldsymbol{d}} \in \mathcal{P}_M$ if and only if $\exists M' \in \mathbb{K}[x]^{m\times n}$ such that $\mathcal{F}_{\boldsymbol{d}'} \in \mathcal{P}_{M'}$. Indeed, permuting $\boldsymbol{d}$ amounts to permuting the components of $\boldsymbol{p}$, *i.e.* permuting the rows of M. This does not affect the existence property.

Theorem 4.7 is an adaptation of [Vil97, Proposition 6.1] and its derivation [PS07, Theorem 3]. Even if the statements of these two papers are in a different but related context, their proof can be applied almost straightforwardly. We will still provide the main steps of the proof, for the sake of clarity and also because we will have to adapt it later in the proof of Theorem 2.4. Note also that we complete the 'if' part of the proof because it was not detailed in earlier references. For this purpose, we introduce the following

Lemma 4.8. *Let $\mathcal{N}$ be a $\mathbb{K}[x]$-submodule of $\mathcal{K}$ of rank l. Then the dimension of $\mathcal{N}$ as $\mathbb{K}$-vector space is at most $f_1 + \cdots + f_l$.*

Proof. First, remark that if $\boldsymbol{q} \in \mathcal{N}$ has its first non-zero element at index p then $a_p\boldsymbol{q} = 0$. Now since $\mathcal{N}$ has rank l, we can consider the matrix B whose rows are the l elements of a basis of $\mathcal{N}$. We operate on the rows of B to obtain the *Hermite normal form* B' of B. The rows $(\boldsymbol{b}'_i)_{1\le i\le l}$ of B' have first non-zero elements at distinct indices $k_1, \ldots, k_l$. Therefore $a_{k_j}\boldsymbol{b}'_j = 0$ and $\{x^i\boldsymbol{b}'_j\}_{\substack{0\le i<f_{k_j}\\ 1\le j\le l}}$ is a generating set of $\mathcal{N}$ and so $\dim_{\mathbb{K}} \mathcal{N} \le f_{k_1} + \cdots + f_{k_l} \le f_1 + \cdots + f_l$ since (f_i) are non increasing and (k_j) pairwise distinct. □

Corollary 4.9. *Let $r \ge 0$, $\boldsymbol{d} \in \mathbb{N}^l$ and $v_1, \ldots, v_l \in \mathcal{K}$ such that $\{x^j\boldsymbol{v}_i\}_{\substack{0\le j<d_i\\ 1\le i\le l}}$ are linearly independent then $\sum_{i=1}^{l} d_i \le \sum_{i=1}^{l} f_i$.*

Proof. We consider $\mathcal{N}$ the $\mathbb{K}[x]$-module spanned by $\{v_1, \ldots, v_l\}$, and we observe that $d_1 + \cdots + d_l \le \dim \mathcal{N} \le f_1 + \cdots + f_l$ by Lemma 4.8. □

Proof of Theorem 4.7. We observe that if $m > n$, we can write $\mathcal{K} = \mathbb{K}[x]^n/\langle a_i\boldsymbol{\varepsilon}'_i\rangle_{1\le i\le n} = \mathbb{K}[x]^m/\langle a_i\boldsymbol{\varepsilon}_i\rangle_{1\le i\le m}$ where $a_j = 1$ for $n+1 \le j \le m$. Hence, we can suppose *w.l.o.g.* that $m = n$.

$\Rightarrow$) By the hypotheses, there exists a matrix $M \in \mathbb{K}[x]^{m\times n}$ such that $\{x^i\boldsymbol{\varepsilon}_jM\}_{x^i\boldsymbol{\varepsilon}_j\in\mathcal{F}_{\boldsymbol{d}}} = \{x^i\boldsymbol{v}_j\}_{0<i<d_j}$ are linearly independent in $\mathcal{K}$ where $\boldsymbol{v}_j := \boldsymbol{\varepsilon}_jM$. Hence, for all $1 \le l \le m$, $\boldsymbol{v}_1, \ldots, \boldsymbol{v}_l$ satisfy the conditions of the Corollary 4.9 and so $\sum_{i=1}^{l} d_i \le \sum_{i=1}^{l} f_i$.

$\Leftarrow$) Set $\boldsymbol{u}_i = \boldsymbol{\varepsilon}_i$ for $1 \le i \le m$ so that $\{x^i\boldsymbol{u}_j\}_{\substack{i<f_j\\ 1\le j\le m}}$ are linearly independent in $\mathcal{M}$. We now consider the matrix $K := [K_1|\ldots|K_m]$ where $K_j \in \mathbb{K}[x]^{m\times f_j}$ is in *Krylov* form, that is $K_j = K(\boldsymbol{u}_j, f_j) := [\boldsymbol{u}_j|x\boldsymbol{u}_j|\ldots|x^{f_j-1}\boldsymbol{u}_j]$ by considering $\boldsymbol{u}_j$ as a column vector. Note that K is full column rank by construction. Our goal is to find vectors $\boldsymbol{v}_1, \ldots, \boldsymbol{v}_m$ such that $[K(\boldsymbol{v}_1, d_1)|\ldots|K(\boldsymbol{v}_m, d_m)]$ is full column rank (see $\widetilde{K}$ later).

For this purpose, we first need to consider the matrix $\overline{K}$ made of columns of K so that it remains full column rank. It is defined as $\overline{K} := [\overline{K}_1|\ldots|\overline{K}_m]$ where for $1 \le j \le m$, $\overline{K}_j \in \mathbb{K}[x]^{m\times d_j}$ are defined iteratively by

$$\overline{K}_j := [K(\boldsymbol{u}_j, \min(f_j, d_j))|K(x^{s_1}\boldsymbol{u}_{j_1}, t_1)|\ldots|K(x^{s_k}\boldsymbol{u}_{j_k}, t_k)]$$

and $K(x^{s_l}\boldsymbol{u}_{j_l}, t_l)$ derives from previously unused columns in K, which we add from left to right, *i.e.* (j_l) are increasing. Since $\sum_{i=1}^{j} d_i \le \sum_{i=1}^{j} f_i$, we will only pick from previous blocks, *i.e.* $j_k < j$. Since we must have depleted a block K_{i_l} before going to another one, we can observe that $s_l + t_l = f_l$ for $l < k$. The last block K_{i_k} is the only one that may not be exhausted, *i.e.* $s_k + t_k \le f_k$. Conversely, $s_l = d_l$ for $l > 1$ because no columns have been picked yet from the blocks j_l, except maybe the first block j_1 where $s_1 \ge d_1$.

We want to transform $\overline{K}_j$ into a Krylov matrix $\widetilde{K}_j$, working block by block. First we extend $[K(\boldsymbol{u}_j, \min(f_j, d_j))|0|\ldots|0]$ to the right to $K(\boldsymbol{u}_j, d_j)$. Then we extend all blocks $[0|\ldots|0|K(x^{s_l}\boldsymbol{u}_{j_l}, t_l)|0|\ldots|0]$ to the left and the right to $K(x^{s'_l}\boldsymbol{u}_{j_l}, d_l)$ where s'_l equals s_l minus the number of columns of the left extension. In this way, the extension matches the original matrix on its non-zero columns. Now we can define $\widetilde{K} := [\widetilde{K}_1|\ldots|\widetilde{K}_m]$, where $\widetilde{K}_j := K(\boldsymbol{v}_j, d_j)$ with $\boldsymbol{v}_j := \boldsymbol{u}_j + \sum_{l=1}^{k} x^{s'_l}\boldsymbol{u}_{j_l}$.

A crucial point of the proof is to show that $s'_k \ge 0$. But since d_i are-non increasing, j_l are increasing and $j_k < j$, we get $s_l \ge d_{j_l} \ge d_{j_k} \ge d_j$. As the number of columns of the left extension is at most d_j, we can conclude $s'_k \ge 0$.

In [Vil97] and [PS07] it is proved that there exist an upper triangular matrices T such that $\widetilde{K} = \overline{K}T$. So we can conclude that $\widetilde{K}$, which is in the desired block Krylov form, is full column rank as is $\overline{K}$, which concludes the proof. □

Example 4.10. We illustrate the construction of the proof of Theorem 4.7 with example. Let $m = 4$, $n = 3$, $\boldsymbol{f} = (8, 4, 4)$ extended to $f_4 = 0$ and $\boldsymbol{d} = (5, 5, 3, 3)$. Remark that $\sum_{i=1}^{l} d_i \le \sum_{i=1}^{l} f_i$ for all $1 \le l \le m$. Then $\overline{K}_1 = K(\boldsymbol{u}_1, d_1)$, $\overline{K}_2 = [K(\boldsymbol{u}_2, f_2)|K(x^{d_1}\boldsymbol{u}_1, d_2 - f_2)]$ picks its missing column from the first unused column of K_1, $\overline{K}_3 = K(\boldsymbol{u}_3, d_3)$, and $\overline{K}_4 = [K(\boldsymbol{u}_4, f_4) = \varnothing|K(x^{d_1+1}\boldsymbol{u}_1, f_1 - (d_1 + 1)|K(x^{d_3}\boldsymbol{u}_3, f_3 - d_3)]$ picks its 3 missing columns first from the 2 unused of K_1, then from the remaining one of K_3. Then the construction extends $\overline{K}$ to $\widetilde{K} = K(\boldsymbol{v}_i, d_i)$ where $\boldsymbol{v}_1 = \boldsymbol{u}_1 = [1, 0, 0]$, $\boldsymbol{v}_2 = \boldsymbol{u}_2 + x^{d_2-(d_1-1)}\boldsymbol{u}_1 = [x, 1, 0]$, $\boldsymbol{v}_3 = \boldsymbol{u}_3 = [0, 0, 1]$ and $\boldsymbol{v}_4 = x^{d_1+1}\boldsymbol{u}_1 + x^{d_3-(f_1-(d_1+1))}\boldsymbol{u}_3 = [x^6, 0, x]$. Finally the matrix M of the statement of Theorem 4.7 has its j-th row $M_{j,*}$ equal to $\boldsymbol{v}_j$. ◇

We now have all the cards in our hand to state the principal constraint on the pivot degrees $\boldsymbol{\delta}_M$ of the relation module $\mathcal{A}_{\mathcal{M},M}$ when M varies in the set of matrices $\mathbb{K}[x]^{m\times n}$ such that $\operatorname{rank}(O_M) = \operatorname{rank}(\varphi_M)$ is fixed. We will denote by $\mathbf{d}_r$ the pivot degrees corresponding to the constraint.

Theorem 4.11. *Recall that $\boldsymbol{f} = (f_1, \ldots, f_m)$ are the degrees of the invariants of $\mathcal{M}$ where $f_i = 0$ for $n+1 \le i \le m$, and let $r = \operatorname{rank}(O_M)$.*

Then $\mathcal{F}_{\boldsymbol{\delta}_M} \geq_{lex} \mathcal{F}_{\mathbf{d}_r}$ *where*

$$\mathcal{F}_{\mathbf{d}_r} = min_{<_{lex}} \left\{ \mathcal{F}_{\boldsymbol{d}} \in \mathsf{Mon}_r \;\middle|\; \forall 1 \leq l \leq m,\ \sum_{i=1}^{l} d_i \leq \sum_{i=1}^{l} f_i \right\} \quad (6)$$

Proof. We know from Proposition 4.6 that $RRP_M = \mathcal{F}_{\boldsymbol{\delta}_M}$ so $\{x^i \boldsymbol{\varepsilon}_j M\}_{\substack{i<\delta_{j,M} \\ 1\leq j\leq m}}$ are linearly independent and $\sum_{i=1}^{m} \delta_{i,M} = r$. Using Theorem 4.7, we get that $\sum_{i=1}^{l} \delta_{i,M} \leq \sum_{i=1}^{l} f_i$ for all $1 \leq l \leq m$. This means that $\mathcal{F}_{\boldsymbol{\delta}_M}$ belongs to the set whose minimum is $\mathcal{F}_{\mathbf{d}_r}$, which implies our result. □

We observe that $r = \mathrm{rank}(O_M)$ must satisfy $0 \leq r \leq \Sigma := \sum_{i=1}^{m} f_i = \dim_{\mathbb{K}} \mathbb{K}[x]^n/\mathcal{M}$ and that $r = \Sigma$ is reachable since $m \geq n$. Note also that $\mathbf{d}_r$ is well-defined in Theorem 4.11 as long as $0 \leq r \leq \Sigma := \sum_{i=1}^{m} f_i$ because it is related to the minimum of a non-empty set.

4.3 Generic row degrees of relation module

We will now show that this pivot degrees constraint $\mathbf{d}_\Sigma$ is attainable by $\boldsymbol{\delta}_M$ for matrices M such that $\mathrm{rank}(O_M) = \mathrm{rank}(\varphi_M) = \dim_{\mathbb{K}} \mathbb{K}[x]^n/\mathcal{M}$ in which case φ_M becomes a bijection. More specifically, we will show that this is the case for almost all matrices $M \in \mathbb{K}[x]^{m\times n}$.

Corollary 4.12. *For a generic matrix* $M \in \mathbb{K}[x]^{m\times n}$*, the pivot degrees* $\boldsymbol{\delta}_M$ *of the relation module* $A_{\mathcal{M},M}$ *satisfy* $\boldsymbol{\delta}_M = \mathbf{d}_\Sigma$ *where* $\Sigma = \sum_{i=1}^{n} f_i$.

Proof. Our goal is to prove that there exists a non-zero polynomial C in the coefficients $m_{i,j,k}$ of the polynomial entries $m_{i,j}$ of M such that $C(m_{i,j,k}) \neq 0$ implies that $\boldsymbol{\delta}_M = \mathbf{d}_\Sigma$.

Since $\sum_{i=1}^{l} \mathrm{d}_{\Sigma,i} \leq \sum_{i=1}^{l} f_i$ for all $1 \leq l \leq m$, we deduce from Theorem 4.7 that there exists $M \in \mathbb{K}[x]^{m\times n}$ such that $\{mM\}_{m\in\mathcal{F}_{\mathbf{d}_\Sigma}}$ are linearly independent. So the Σ-minor of the ordered matrix O_M of M corresponding to those lines is non-zero. We now consider this Σ-minor as a function C in the coefficients $m_{i,j,k}$ of the polynomial entries $m_{i,j}$ of M. Note that $C \in \mathbb{K}[m_{i,j,k}]$ since the entries of O_M are linear combinations of $m_{i,j,k}$. Indeed, we can write $m_{i,j} = \sum_{k=0}^{f_j-1} m_{i,j,k} x^k$ because $m_{i,j}$ is only considered modulo a_j, and the coefficient of O_M w.r.t. line $x^u\boldsymbol{\varepsilon}_i$ and column $x^v\boldsymbol{\varepsilon}'_j$ is $\sum_{k=0}^{f_j-1} m_{i,j,k} c_{j,k,u,v}$ where $c_{j,k,u,v} \in \mathbb{K}$ is the coefficient of $(x^{k+u} \bmod a_j)$ in x^v. We have seen that C admits a nonzero evaluation so is a non-zero polynomial.

Now for any matrix M such that $C(m_{i,j,k}) \neq 0$, the vectors $\{mM\}_{m\in\mathcal{F}_{\mathbf{d}_\Sigma}}$ must be linearly independent, so $\mathrm{rank}(O_M) = \Sigma$. We have $RRP_M \leq_{lex} \mathcal{F}_{\mathbf{d}_\Sigma}$ because $\mathcal{F}_{\mathbf{d}_\Sigma} \in \mathcal{P}_M$ (see Definition 4.5). Theorem 4.11 gives the other inequality, so $\mathcal{F}_{\mathbf{d}_\Sigma} = RRP_M = \mathcal{F}_{\boldsymbol{\delta}_M}$ and $\boldsymbol{\delta}_M = \mathbf{d}_\Sigma$. □

4.3.1 Special cases. In this section, we will see that our definition of the generic pivot degrees $\mathbf{d}_\Sigma$ in (6) has a simplified expression in a wide range of settings. Set the notation $\overline{s} = \max(\boldsymbol{s})$. We will see that under some assumptions the expected row degrees $\mathbf{p}_\Sigma := \mathbf{d}_\Sigma + \boldsymbol{s}$ has a nice form. Define p and u be the quotient and remainder of the Euclidean division $\sum_{i=1}^{m}(f_i + s_i) = p \cdot m + u$. The expected nice form of the row degrees will be

$$\boldsymbol{p} := (\underbrace{p+1, \ldots, p+1}_{u \text{ times}}, \underbrace{p, \ldots, p}_{m-u \text{ times}}). \quad (7)$$

This nice form will appear if the following conditions on $\boldsymbol{f}$ and $\boldsymbol{s}$ hold:

$$p \geq \overline{s} \quad (8)$$

$$\forall 1 \leq l \leq m-1,\ \sum_{i=1}^{l} p_i \leq \sum_{i=1}^{l} (f_i + s_i) \quad (9)$$

Theorem 4.13. *Let* $\boldsymbol{p}$ *as in* (7)*, and let* $\boldsymbol{f}$ *be non-increasing such that* (8) *and* (9) *hold. Then* $\mathbf{p}_\Sigma = \boldsymbol{p}$.

This nice form of row degree was already observed in different but related settings. To the best of our knowledge, it can be found in [Vil97, Proposition 6.1] for row degrees of minimal generating matrix polynomial but with no shift, in [PS07, Corollary 1] for dimensions of blocks in a shifted Hessenberg form but the link to row degree is unclear and no shift is discussed (shifted Hessenberg is not related to our shift $\boldsymbol{s}$), and in [JV05, after (2)] for kernel basis were $m = 2n$ with no shifts.

Proof. Denote again $\Sigma = \sum_{i=1}^{n} f_i$. Let $\overline{\mathcal{F}}$ be the first Σ monomials of $\mathbb{K}[x]^m$ for the $\prec_{\boldsymbol{s}-TOP}$ ordering. Let $\boldsymbol{p} = (p+1, \ldots, p+1, p, \ldots, p)$ be the candidate row degrees as in the theorem statement and $\boldsymbol{d} = \boldsymbol{p} - \boldsymbol{s}$ be the corresponding pivot degrees. Note that (8) implies that $p \geq \overline{s}$ so $\boldsymbol{d} \in \mathbb{N}^m$.

First we show that (8) implies $\overline{\mathcal{F}} = \mathcal{F}_{\boldsymbol{d}}$. For the first part, in order to prove $\overline{\mathcal{F}} = \mathcal{F}_{\boldsymbol{d}}$, we need to show that $d_i = \min\{d \in \mathbb{N} \mid x^d\boldsymbol{\varepsilon}_i \notin \overline{\mathcal{F}}\}$. We already know that $d_i \in \mathbb{N}$. We will need to study the row degrees of the first monomials to conclude. The monomials of $\mathbb{K}[x]^m$ of $\boldsymbol{s}$-row degree r ordered increasingly for $\prec_{\boldsymbol{s}-TOP}$ are $[x^{r-s_i}\boldsymbol{\varepsilon}_i]$ for increasing $1 \leq i \leq m$ such that $s_i \leq r$. There are m such monomials when $r \geq \overline{s}$. The monomials of $\boldsymbol{s}$-row degree less than $\overline{s}$ are $\{x^i\boldsymbol{\varepsilon}_j\}_{i+s_j<\overline{s}}$ and their number is $\sum_{i=1}^{m}(\overline{s} - s_i)$. From this we can deduce that the row degree of the n-th smallest monomial is $\left\lfloor (n-1-\sum_{i=1}^{m}(\overline{s}-s_i))/m \right\rfloor + \overline{s} = \left\lfloor (n-1+\sum_{i=1}^{m} s_i)/m \right\rfloor$ provided that $n \geq \sum_{i=1}^{m}(\overline{s}-s_i)+1$. We can now remark that the $(\Sigma+1)$-th smallest monomial has $\boldsymbol{s}$-row degree p. More precisely, the $(\Sigma+1)$-th smallest monomial is the $(u+1)$-th monomial of row-degree r, so $\overline{\mathcal{F}}$ is equal to all monomials of row degree less than p and the first u monomials of row degree p. This proves $d_i = \min\{d \in \mathbb{N} \mid x^d\boldsymbol{\varepsilon}_i \notin \overline{\mathcal{F}}\}$ and $\overline{\mathcal{F}} = \mathcal{F}_{\boldsymbol{d}}$.

Second we deduce from (9) that for all $1 \leq l \leq m$, $\sum_{i=1}^{l} d_i = \sum_{i=1}^{l}(p_i - s_i) \leq \sum_{i=1}^{l} f_i$, so $\mathcal{F}_{\mathbf{d}_r} \leq_{lex} \mathcal{F}_{\boldsymbol{d}}$ by Theorem 4.11 and finally $\mathcal{F}_{\mathbf{d}_r} = \mathcal{F}_{\boldsymbol{d}}$ because $\overline{\mathcal{F}}$ is the smallest set of Σ monomials. □

Example 4.14. Here we provide 3 examples of generic pivot degrees $\mathbf{d}_\Sigma$ and row degrees $\mathbf{p}_\Sigma$: Corollary 4.12 applies only to the first situation because the second and third situations are constructed so that (8) and respectively (9) are not satisfied. Let $m = n = 3$ and $\boldsymbol{s} = (0, 2, 4)$ so that $\overline{s} = 4$ and $\sum(\overline{s} - s_i) = 6$.

In the first situation $\boldsymbol{f} = (6, 1, 0)$, so $\sum(f_i + s_i) = 4m + 1$ and using Corollary 4.12 we get $\mathbf{p}_\Sigma = (5, 4, 4)$ from (7) and $\mathbf{d}_\Sigma = (5, 2, 0)$. In the second situation, $\boldsymbol{f} = (3, 0, 0)$ and (8) is not satisfied. We use Theorem 4.13 to get $\mathbf{d}_\Sigma = (3, 0, 0)$ from (6) and $\mathbf{p}_\Sigma = (3, 2, 4)$. Finally in the third situation, $\boldsymbol{f} = (3, 3, 1)$ and (9) is not satisfied. We

use Theorem 4.13 to get $\mathbf{d}_\Sigma = (3,3,1)$ from (6) and $\mathbf{p}_\Sigma = (3,5,5)$. Let $\mathcal{F}_1, \mathcal{F}_2, \mathcal{F}_3$ be the respective families of monomial of the three situations. We picture these families in the following table, where *Mon* are the first monomials for $\prec_{s-TOP}$

Mon	$\boldsymbol{\varepsilon}_1$	$x\boldsymbol{\varepsilon}_1$	$x^2\boldsymbol{\varepsilon}_1$	$\boldsymbol{\varepsilon}_2$	$x^3\boldsymbol{\varepsilon}_1$	$x\boldsymbol{\varepsilon}_2$	$x^4\boldsymbol{\varepsilon}_1$	$x^2\boldsymbol{\varepsilon}_2$	$\boldsymbol{\varepsilon}_3$
$\text{rdeg}_{\boldsymbol{s}}$	0	1	2		3		4		
$\mathcal{F}_1$	•	•	•	•	•	•	•		
$\mathcal{F}_2$	•	•	•						
$\mathcal{F}_3$	•	•	•	•		•		•	•

5 UNIQUENESS RESULTS ON SRFR

Let's recall SRFR defined in Section 2.1: let $a_1, \ldots, a_n \in \mathbb{K}[x]$ with degrees $f_i := \deg(a_i)$ and $\boldsymbol{u} := (u_1, \ldots, u_n) \in \mathbb{K}[x]^n$ such that $\deg(u_i) < f_i$ and $0 < N_i \leq f_i$ for $1 \leq i \leq n$, $0 < D \leq \min_{1\leq i\leq n}\{f_i\}$. We want to reconstruct $(\boldsymbol{v}, d) = (v_1, \ldots, v_n, d) \in \mathbb{K}[x]^{1\times(n+1)}$ such that $v_i = du_i \bmod a_i, \deg(v_i) < N_i, \deg(d) < D$. We consider $\mathcal{M} = \langle a_i\boldsymbol{\varepsilon}_i'\rangle$ and we denote by $S_{\boldsymbol{u}}$ the set of tuples which verify (3).

Lemma 5.1. *For the shift $\boldsymbol{s} = (-N_1, \ldots, -N_n, -D) \in \mathbb{Z}^{n+1}$, we have $(\boldsymbol{v}, d) \in S_{\boldsymbol{u}} \Leftrightarrow (\boldsymbol{v}, d) \in \mathcal{A}_{\mathcal{M},R_{\boldsymbol{u}}}$ with $rdeg_{\boldsymbol{s}}((\boldsymbol{v}, d)) < 0$, where*

$$R_{\boldsymbol{u}} := \begin{bmatrix} \mathrm{Id}_n \\ -\boldsymbol{u} \end{bmatrix} \in \mathbb{K}[x]^{(n+1)\times n} \tag{10}$$

Proof. Observe that $(\boldsymbol{v}, d) \in S_{\boldsymbol{u}}$ if and only if it satisfies the equation $\boldsymbol{v} - d\boldsymbol{u} = (\boldsymbol{v}, d)R_{\boldsymbol{u}} = 0 \bmod \mathcal{M}$, that is $(\boldsymbol{v}, d) \in \mathcal{A}_{\mathcal{M},R_{\boldsymbol{u}}}$, and if it satisfies the degree conditions equivalent to $\text{rdeg}_{\boldsymbol{s}}((\boldsymbol{v}, d)) = \max\{\deg(v_1) - N_1, \ldots, \deg(v_n) - N_n, \deg(d) - D\} < 0$ (see Def. 3.1). □

So in order to study the solutions of SRFR we introduce the $\boldsymbol{s}$-row degrees $\boldsymbol{\rho}_{\boldsymbol{u}} := \boldsymbol{\rho}_{R_{\boldsymbol{u}}}$ and the $\boldsymbol{s}$-pivot indices $\boldsymbol{\delta}_{\boldsymbol{u}} := \boldsymbol{\delta}_{R_{\boldsymbol{u}}}$ of $A_{\mathcal{M},R_{\boldsymbol{u}}}$ (see Definition 4.2). As remarked just after the *predictable degree property* (Lemma 3.2),

$$\dim_{\mathbb{K}} S_{\boldsymbol{u}} = \dim_{\mathbb{K}}(A_{\mathcal{M},R_{\boldsymbol{u}}})_{<0} = -\sum_{\rho_{\boldsymbol{u},i}<0} \rho_{\boldsymbol{u},i}. \tag{11}$$

We can now prove our main Theorem 2.4 about uniqueness in SRFR. Recall the theorem's statement: assuming $\sum_{i=1}^n f_i = \sum_{i=1}^n N_i + D - 1$ then the solution space $S_{\boldsymbol{u}}$ has dimension 1 as $\mathbb{K}$-vector space for generic $\boldsymbol{u} = (u_1, \ldots, u_n) \in \mathbb{K}[x]^{1\times n}$.

Proof of Theorem 2.4. By the previous considerations (see (11)) it is sufficient to prove that for generic $\boldsymbol{u} \in \mathbb{K}[x]^{n+1}, \boldsymbol{\rho}_{\boldsymbol{u}} = (0, \ldots, 0, -1)$.

First, we need to show that the generic $\boldsymbol{s}$-row degrees $\mathbf{p}_\Sigma$ have the expected nice form $\boldsymbol{p} = (0, \ldots, 0, -1)$ ($p = -1$ and $u = n = m - 1$ because $\sum(f_j + s_j) = -1 \cdot m + (m-1)$, see (7)). It remains to check that we verify the hypotheses of Theorem 4.13. By (8), $\bar{s} \leq -1 = p$. By (9), $\sum_{i=1}^l p_i \leq 0 \leq \sum_{i=1}^l (f_i + s_i)$ for all $0 \leq l \leq m-1$ since $f_i + s_i \geq 0 \geq p_i$ for all i.

We now show that there exists $\boldsymbol{u}$ such that $R_{\boldsymbol{u}}$ satisfies the genericity condition C of Corollary 4.12. This will prove that our new genericity condition $C'(u_{j,k})$ is not the zero polynomial, where C' is $C(m_{i,j,k})$ evaluated on matrices $R_{\boldsymbol{u}}$, and $u_{j,k}$ are the polynomial coefficients of u_j. Let's show that the construction of the proof of Theorem 4.7 provides a matrix of the form $R_{\boldsymbol{u}}$ in our case $(d_1, \ldots, d_{n+1}) = (N_1, \ldots, N_n, D-1)$ and $m = n+1$. In particular, by SRFR assumptions, for any $1 \leq i \leq n$, $d_i \leq f_i$ and so the matrices $\overline{K}_i = [K(\boldsymbol{u}_i, d_i)]$ are already in Krylov form. On the other hand, the last matrix is in the form $\overline{K}_{n+1} = [K(x^{d_j}\boldsymbol{u}_j, t_j)]_{1\leq j\leq n}$ where $d_j + t_j = f_j$ (here $f_{n+1} = 0$). Then $\widetilde{K}_{n+1} = [K(\sum_{j=1}^n x^{s'_j}\boldsymbol{u}_j, d_j)]$ and we need to prove that $s'_j \geq 0$ differently because we don't have the assumption about the non-increasing $\boldsymbol{d}$. Recall that s'_j is s_j minus the number of columns added to extend the matrix to the left. This number of columns is at most d_{n+1} minus the size t_l of the current block. So $s'_l \geq d_l - (d_{n+1} - t_l) = d_l - (d_{n+1} - (f_l - d_l)) = f_l - d_{n+1} \geq 0$ because $d_{n+1} = D - 1 \leq D \leq \min(f_i)$ and so the construction works.

When $\mathbb{K}$ is a finite field of cardinality q, we want to bound the number of $\boldsymbol{u}$ such that $C'(u_{j,k}) = 0$. Recall that $u_j = \sum_{k=0}^{f_j-1} u_{j,k}x^k$ and that $C' \in \mathbb{K}[u_{j,k}]$ is a constructed as a Σ-minor of the ordered matrix $O_{R_{\boldsymbol{u}}}$ where $\Sigma = \sum_{i=1}^n f_i$. The coefficients of $O_{R_{\boldsymbol{u}}}$ are in $\mathbb{K}$, except for the $D-1$ lines corresponding to $(x^u\boldsymbol{\varepsilon}_{n+1})_{0\leq u<D-1}$ which are linear combinations of $u_{j,k}$ as mentioned in the proof of Corollary 4.12. Therefore the total degree of C' is $\leq D-1$ and we can conclude using Schwartz-Zippel Lemma that the proportion of instances leading to non-uniqueness among all possible instances is $\leq (D-1)/q$. □

REFERENCES

[BK14] B. Boyer and E. Kaltofen. Numerical linear system solving with parametric entries by error correction. In *Proceedings of SNC'14*, 2014.

[BKY03] D. Bleichenbacher, A. Kiayias, and M. Yung. Decoding of interleaved reed solomon codes over noisy data. In *Proceedings of ICALP'03*, 2003.

[BMS04] A. Brown, L. Minder, and A. Shokrollahi. Probabilistic decoding of interleaved RS-codes on the q-ary symmetric channel. In *Proceedings of ISIT'04*, 2004.

[BW86] E. Berlekamp and L. Welch. Error correction of algebraic block codes., 1986. US Patent 4,633,470.

[Cab71] S. Cabay. Exact solution of linear equations. In *Proceedings of SYMSAC'71*, 1971.

[CLO98] D. Cox, J. Little, and D. O'Shea. *Using algebraic geometry*, volume 185 of *Graduate Texts in Mathematics*. Springer-Verlag, 1998.

[DF03] D. S. Dummit and R. M. Foote. *Abstract Algebra*. Wiley, 3rd edition, 2003.

[DPS15] J.-G. Dumas, C. Pernet, and Z. Sultan. Computing the Rank Profile Matrix. In *Proceedings of ISSAC'15*, 2015.

[GG13] J. von zur Gathen and J. Gerhard. *Modern Computer Algebra*. Cambridge University Press, 3rd edition, 2013.

[GLZ19] E. Guerrini, R. Lebreton, and I. Zappatore. Polynomial linear system solving with errors by simultaneous polynomial reconstruction of interleaved reed-solomon codes. In *Proceedings of ISIT'19*, 2019.

[GLZ20] E. Guerrini, R. Lebreton, and I. Zappatore. Enhancing simultaneous rational function recovery: adaptive error correction capability and new bounds for applications, 2020. Arxiv eprint 2003.01793.

[JV05] C.-P. Jeannerod and G. Villard. Essentially optimal computation of the inverse of generic polynomial matrices. *Journal of Complexity*, 21(1), 2005.

[KPSW17] E. L. Kaltofen, C. Pernet, A. Storjohann, and C. Waddell. Early termination in parametric linear system solving and rational function vector recovery with error correction. In *Proceedings of ISSAC'17*, 2017.

[Nei16] V. Neiger. *Bases of relations in one or several variables: fast algorithms and applications*. Phd thesis, ÉNS Lyon - University of Waterloo, 2016.

[OS07] Z. Olesh and A. Storjohann. The vector rational function reconstruction problem. In *Proceedings of the Waterloo Workshop*, 2007.

[Per14] C. Pernet. *High Performance and Reliable Algebraic Computing*. Habilitation à diriger des recherches, Université Joseph Fourier, Grenoble 1, 2014.

[PRN17] S. Puchinger and J. Rosenkilde né Nielsen. Decoding of interleaved reed-solomon codes using improved power decoding. In *Proceedings of ISIT'17*, 2017.

[PS07] C. Pernet and A. Storjohann. Faster Algorithms for the Characteristic Polynomial. In *Proceedings of ISSAC'07*, 2007.

[RNS16] J. Rosenkilde né Nielsen and A. Storjohann. Algorithms for simultaneous padé approximations. In *Proceedings of ISSAC'16*, 2016.

[SSB09] G. Schmidt, V. R. Sidorenko, and M. Bossert. Collaborative decoding of interleaved reed–solomon codes and concatenated code designs. *IEEE Transactions on Information Theory*, 55(7), 2009.

[Vil97] G. Villard. *A study of Coppersmith's block Wiedemann algorithm using matrix polynomials*. IMAG, 1997.

Efficient ECM Factorization in Parallel with the Lyness Map

Andrew Hone*
A.N.W.Hone@kent.ac.uk
University of Kent
Canterbury, UK

ABSTRACT

The Lyness map is a birational map in the plane which provides one of the simplest discrete analogues of a Hamiltonian system with one degree of freedom, having a conserved quantity and an invariant symplectic form. As an example of a symmetric Quispel-Roberts-Thompson (QRT) map, each generic orbit of the Lyness map lies on a curve of genus one, and corresponds to a sequence of points on an elliptic curve which is one of the fibres in a pencil of biquadratic curves in the plane.

Here we present a version of the elliptic curve method (ECM) for integer factorization, which is based on iteration of the Lyness map with a particular choice of initial data. More precisely, we give an algorithm for scalar multiplication of a point on an arbitrary elliptic curve over $\mathbb{Q}$, which is represented by one of the curves in the Lyness pencil. In order to avoid field inversion (**I**), and require only field multiplication (**M**), squaring (**S**) and addition, projective coordinates in $\mathbb{P}^1 \times \mathbb{P}^1$ are used. Neglecting multiplication by curve constants (assumed small), each addition of the chosen point uses 2**M**, while each doubling step requires 15**M**. We further show that the doubling step can be implemented efficiently in parallel with four processors, dropping the effective cost to 4**M**.

In contrast, the fastest algorithms in the literature use twisted Edwards curves (equivalent to Montgomery curves), which correspond to a subset of all elliptic curves. Scalar muliplication on twisted Edwards curves with suitable small curve constants uses 8**M** for point addition and 4**M**+4**S** for point doubling, both of which can be run in parallel with four processors to yield effective costs of 2**M** and 1**M** + 1**S**, respectively. Thus our scalar multiplication algorithm should require, on average, roughly twice as many multiplications per bit as state of the art methods using twisted Edwards curves. In our conclusions, we discuss applications where the use of Lyness curves may provide potential advantages.

CCS CONCEPTS

• **Mathematics of computing** → **Nonlinear equations**; • **Computing methodologies** → *Parallel algorithms*; • **Security and privacy** → **Mathematical foundations of cryptography**.

*Work begun on leave in the School of Mathematics & Statistics, UNSW, Sydney, Australia.

ISSAC '20, July 20–23, 2020, Kalamata, Greece

ACM ISBN 978-1-4503-7100-1/20/07…$15.00
https://doi.org/10.1145/3373207.3404044

KEYWORDS

Lyness map, elliptic curve method, scalar multiplication

ACM Reference Format:
Andrew Hone. 2020. Efficient ECM Factorization in Parallel with the Lyness Map. In *International Symposium on Symbolic and Algebraic Computation (ISSAC '20), July 20–23, 2020, Kalamata, Greece.* ACM, New York, NY, USA, 7 pages. https://doi.org/10.1145/3373207.3404044

1 INTRODUCTION

In 1942 it was observed by Lyness [24] that iterating the recurrence relation

$$u_{n+2}u_n = a\,u_{n+1} + a^2 \tag{1}$$

with an arbitrary pair of initial values u_0, u_1 produces the sequence

$$u_0, u_1, \frac{a(u_1+a)}{u_0}, \frac{a^2(u_0+u_1+a)}{u_0u_1}, \frac{a(u_0+a)}{u_1}, u_0, u_1, \ldots,$$

which is periodic with period five. The Lyness 5-cycle also arises in a frieze pattern [11], or as a simple example of Zamolodchikov periodicity in integrable quantum field theories [31], which can be explained in terms of the associahedron K_4 and the cluster algebra defined by the A_2 Dynkin quiver [16], leading to a connection with Abel's pentagon identity for the dilogarithm [26]. Moreover, the map corresponding to $a = 1$, that is

$$(x, y) \mapsto \left(y, \frac{y+1}{x}\right), \tag{2}$$

appears in the theory of the Cremona group: as proved by Blanc [8], the birational transformations of the plane that preserve the symplectic form

$$\omega = \frac{1}{xy}\,\mathrm{d}x \wedge \mathrm{d}y, \tag{3}$$

are generated by $SL(2,\mathbb{Z})$, the torus and transformation (2).

More generally, the name Lyness map is given to the birational map

$$\varphi : \quad (x, y) \mapsto \left(y, \frac{ay+b}{x}\right), \tag{4}$$

which contains two parameters a, b (and there are also higher order analogues [29]). The parameter $a \neq 0$ can be removed by rescaling $(x, y) \to (ax, ay)$, so that this is really a one-parameter family, referred to in [15] as "the simplest singular map of the plane." However, we will usually retain a below for bookkeeping purposes.

Unlike the special case $b = a^2$, corresponding to (1), in general the orbits of (4) do not all have the same period, and over an infinite field (e.g. $\mathbb{Q}, \mathbb{R}$ or $\mathbb{C}$) generic orbits are not periodic. However, the general map still satisfies $\varphi^*(\omega) = \omega$, i.e. the symplectic form (3) is preserved, and there is a conserved quantity $K = K(x, y)$ given by

$$K = \frac{xy(x+y) + a(x+y)^2 + (a^2+b)(x+y) + ab}{xy}. \tag{5}$$

Since $\varphi^*(K) = K$, each orbit lies on a fixed curve $K = \text{const}$. Thus the Lyness map is a simple discrete analogue of a Hamiltonian system with one degree of freedom, and (4) also commutes with the flows of the Hamiltonian vector field $\dot{x} = \{x, K\}, \dot{y} = \{y, K\}$, where $\{,\}$ is the Poisson bracket defined by (3). Moreover, generic level curves of K have genus one, so that (real or complex) iterates of the Lyness map can be expressed in terms of elliptic functions [7].

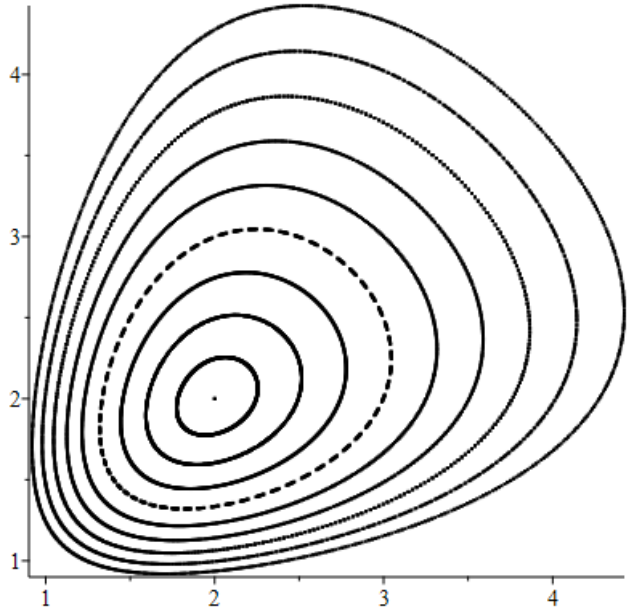


Figure 1: A family of rational orbits of (4) in the positive quadrant, iterated for $a = 1$, $b = 2$ with initial values $(x, y) = (2 + 0.2k, 2 + 0.2k)$ for $k = 0, \ldots, 9$.

The origin of the conserved quantity (5) may seem mysterious, but becomes less so when one observes that (4) is a particular example of a symmetric QRT map [27, 28], and as such it can be derived by starting from a pencil of biquadratic curves, in this case

$$xy(x + y) + a(x + y)^2 + (a^2 + b)(x + y) + ab + \lambda xy = 0, \quad (6)$$

which by symmetry admits the involution $\iota : (x, y) \mapsto (y, x)$. On each curve $\lambda = -K = \text{const}$ there are also the horizontal/vertical switches, obtained by swapping a point on the curve with the other intersection with a horizontal/vertical line. Using the Vieta formula for the product of roots of a quadratic, the horizontal switch can be written explicitly as the birational involution $\iota_h : (x, y) \mapsto (x^{-1}(ay + b), y)$, and then the Lyness map (4) is just the composition $\varphi = \iota \circ \iota_h$. Standard results about elliptic curves then imply that applying the map to a point $\mathcal{P}_0 = (x, y)$ corresponds to a translation $\mathcal{P}_0 \mapsto \mathcal{P}_0 + \mathcal{P}$ in the group law of the curve, where the shift $\mathcal{P}$ is independent of $\mathcal{P}_0$.

There is an associated elliptic fibration of the plane over $\mathbb{P}^1$, defined by $(x, y) \mapsto \lambda = -K(x, y)$, so that each point (x, y) lies in one of the fibres, apart from the base points where $xy(x + y) + a(x + y)^2 + (a^2 + b)(x + y) + ab$ and xy vanish simultaneously. (For more details on the geometry of QRT maps see [20, 21, 30], or the book [13], where the Lyness map is analysed in detail in chapter 11.)

Part of one such fibration can be seen in Figure 1, which for the case $a = 1$, $b = 2$ shows points on the fibres corresponding to the values

$$K = \frac{2(k^3 + 40k^2 + 575k + 2875)}{5(10 + k)^2} \quad (7)$$

for $k = 0, \ldots, 9$.

In the next section we describe the group law on the invariant curves of the Lyness map. Section 3 describes an algorithm, first outlined in [19], for carrying out the elliptic curve method (ECM) for integer factorization using the Lyness map in projective coordinates. There is a long history of finding speedups and improved curve choices for the ECM, e.g. using Montgomery curves [6, 10, 25], Hessian curves [17] and Edwards curves [14] or their twisted versions (see [1–5, 18] and references therein). In section 4 we explain how the ECM algorithm with Lyness curves can be implemented more efficiently in parallel, although this is still roughly twice as slow as the fastest parallel algorithm in [18]. The final section contains some conclusions.

2 LYNESS CURVES AS ELLIPTIC CURVES

The affine curve defined by fixing K in (5), that is

$$xy(x + y) + a(x + y)^2 + (a^2 + b)(x + y) + ab = Kxy. \quad (8)$$

is both cubic (total degree three) and biquadratic in x, y, and (subject to a discriminant condition, described below) it extends to a smooth projective cubic in $\mathbb{P}^2$, or a smooth curve of bidegree $(2, 2)$ in $\mathbb{P}^1 \times \mathbb{P}^1$. See Figure 2 for a plot of a smooth Lyness curve in $\mathbb{R}^2$. An example of a singular Lyness curve is given by

$$xy(x + y) + (x + y)^2 + 3(x + y) + 2 = \frac{23}{2}xy,$$

which is the case $k = 0$ of (7), and contains the fixed point at $(x, y) = (2, 2)$ in Figure 1.

In order to consider a Lyness curve (8) as an elliptic curve, we must define the group law, in terms of addition of pairs of points, with a distinguished point O as the identity element. For what follows, we will make use of the fact that a Lyness curve is birationally equivalent to a Weierstrass cubic, as described by the following (which paraphrases a result from [19]).

Theorem 1. *Given a fixed choice of rational point $(\nu, \xi) \in \mathbb{Q}^2$ on a Weierstrass cubic*

$$E(\mathbb{Q}) : \quad (y')^2 = (x')^3 + Ax' + B \quad (9)$$

over $\mathbb{Q}$, a point (x, y) on a Lyness curve (8) is given in terms of $(x', y') \in E(\mathbb{Q})$ by $x = -\beta(\alpha u + \beta)/(uv) - a$, $y = -\beta uv - a$, where $u = \nu - x'$, $v = (4\xi y' + Ju - \alpha)/(2u^2)$ and the parameters are related by

$$a = -\alpha^2 - \beta J, \quad b = 2a^2 + a\beta J - \beta^3, \quad K = -2a - \beta J, \quad (10)$$

with $\alpha = 4\xi^2$, $J = 6\nu^2 + 2A$, $\beta = \frac{1}{4}J^2 - 12\nu\xi^2$. Conversely, given $a, b, K \in \mathbb{Q}$, a point (x, y) on (8) corresponds to $(\bar{x}, \bar{y}) \in \bar{E}(\mathbb{Q})$, a twist of $E(\mathbb{Q})$ with coefficients $\bar{A} = \alpha^2\beta^4 A$, $\bar{B} = \alpha^3\beta^6 B$, and the point $\mathcal{P} = (\infty, -a)$ on (8) corresponds to $(\bar{\nu}, \bar{\xi}) = \left(\frac{1}{12}(\beta J)^2 - \frac{1}{3}\beta^3, \frac{1}{2}\alpha^2\beta^3\right)$ on $\bar{E}(\mathbb{Q})$.

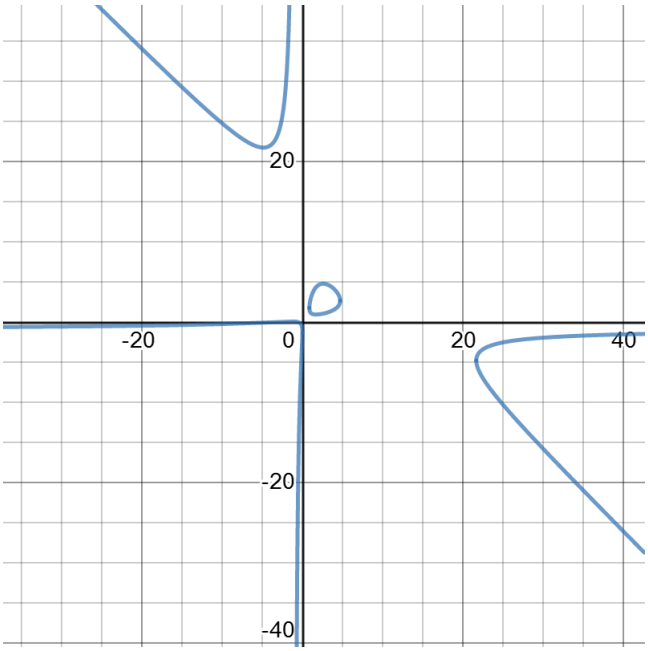


Figure 2: The Lyness curve $xy(x+y)+(x+y)^2+3(x+y)+2 = \frac{109}{8}xy$ **in** $\mathbb{R}^2$.

By rewriting $\bar{A}, \bar{B}$ in terms of a, b, K via the above relations, one can compute the discriminant $\Delta = -16(4\bar{A}^3 + 27\bar{B}^2)$, such that $\Delta \neq 0$ gives the condition for the curve (8) to be nonsingular. The j-invariant of the Lyness curve is

$$j = \frac{(K+a)^{-2}(Ka+b)^{-3}(\hat{g}_2)^3}{(Ka^3 - 8a^4 + K^2b - 10Kab + 13a^2b - 16b^2)},$$

where the numerator has the cube of

$$\hat{g}_2 = K^4 - 8K^3a + 16Ka^3 + 16a^4 - 16K^2b - 8Kab - 16a^2b + 16b^2.$$

With the above equivalence, the group law on the Lyness curve, with identity element given by the point $\mathcal{O} = (\infty, \infty)$, can be found by translating the standard Weierstrass addition formulae for (x', y') into the corresponding expressions for the coordinates (x, y). Alternatively, since the curve (8) is cubic, the usual chord and tangent method can be applied directly, yielding the formula for affine addition as

$$(x_1, y_1) + (x_2, y_2) = (x_3, y_3), \tag{11}$$

$$x_3 = \frac{(ay_1 - ay_2 - x_1y_2 + x_2y_1)(ax_1y_2 - ax_2y_1 - by_1 + by_2)}{y_1y_2(x_1 - x_2)(x_1 - x_2 + y_1 - y_2)},$$

$$y_3 = \frac{(ax_1 - ax_2 + x_1y_2 - x_2y_1)(ax_2y_1 - ax_1y_2 - bx_1 + bx_2)}{x_1x_2(y_1 - y_2)(x_1 - x_2 + y_1 - y_2)}.$$

The elliptic involution that sends any point $\mathcal{P}$ to its inverse $-\mathcal{P}$ is the symmetry $\iota : (x, y) \mapsto (y, x)$.

The above addition law is not unified, in the sense that it cannot be applied when the two points to be added are the same; nor does it make sense if one of the points is $\mathcal{O}$. However, for adding (x_1, y_1) to either of the other two points at infinity, which are $\mathcal{P} = (\infty, -a)$ and $-\mathcal{P} = (-a, \infty)$, this addition formula does make sense: taking the limit $x_2 \to \infty$ with $y_2 \to -a$, we see that

$$(x_1, y_1) + (\infty, -a) = \varphi\Big((x_1, y_1)\Big), \tag{12}$$

so on each level curve $K = \text{const}$ an iteration of the Lyness map (4) corresponds to addition of the point $\mathcal{P}$.

In the case $(x_1, y_1) = (x_2, y_2)$, either by transforming the doubling formula for the Weierstrass curve (9), or by computing the tangent to (8), the formula for doubling (x, y) to $(x, y) + (x, y) = 2(x, y)$ is found to be

$$\psi : \quad (x, y) \mapsto \Big(R(x, y), R(y, x)\Big), \tag{13}$$

where

$$R(x, y) = \frac{(xy - ay - b)(x^2y - a^2x - by - ab)}{x(x - y)(y^2 - ax - b)}, \tag{14}$$

and satisfies $\psi^*(\omega) = 2\omega$, so that the symplectic form is doubled by this transformation.

Apart from combinations involving exceptional points like $\mathcal{O}$, the formulae (11) and (13) define the abelian group law on the curve (8).

3 ECM USING LYNESS

In order to factor a composite integer N, for finding small factors one can use trial division, Pollard's rho method or the $p - 1$ method, while for the large prime factors of a modulus N used in RSA cryptography the number field sieve (NFS) is most effective [12]. However, for finding many medium-sized primes, the ECM is the method of choice, and is commonly used as a first stage in the NFS.

To implement the original version of the ECM, due to Lenstra [22], one should pick a random elliptic curve E, defined over $\mathbb{Q}$ by a Weierstrass cubic (9), and a random point $\mathcal{P} \in E$, then compute the scalar multiple $s\mathcal{P}$ in the group law of the curve, using arithmetic in the ring $\mathbb{Z}/N\mathbb{Z}$. The method succeeds if, at some stage in the computation of this scalar multiple $s\mathcal{P}$, the denominator D of the coordinate x' has a has a non-trivial common factor with N, that is $g = \gcd(D, N)$ with $1 < g < N$.

Typically s is chosen as a prime power less than some bound B_1, or the product of all such prime powers. For composite N, the curve is no longer a group, but rather is a group scheme (or pseudocurve [12]) over $\mathbb{Z}/N\mathbb{Z}$, meaning that the addition law $\mathcal{P}_1 + \mathcal{P}_2$ does not give a point in $(\mathbb{Z}/N\mathbb{Z})^2$ for every pair of points $\mathcal{P}_1, \mathcal{P}_2$. The success of the method is an indication that, for some prime factor $p|N$, $s\mathcal{P} = \mathcal{O}$ in the group law of the genuine elliptic curve $E(\mathbb{F}_p)$, which happens whenever s is a multiple of the order $\#E(\mathbb{F}_p)$.

The computation of the scalar multiple $s\mathcal{P}$ is usually regarded as the first stage of the ECM. If it is unsuccessful, then a second stage can be implemented, which consists of calculating multiples $\ell s\mathcal{P}$ for small primes ℓ less than some bound $B_2 > B_1$. If the second stage fails, then one can either increase the value of B_1, or start again with a new curve E and point $\mathcal{P}$. Here we are primarily concerned with calculating the scalar multiple $s\mathcal{P}$ in stage 1. Stage 2 requires an FFT extension [9], and the cost of the elliptic curve arithmetic involved is negligible in that context.[1]

[1]The author is grateful to one of the reviewers for pointing this out.

The x-coordinate on a Weierstrass curve can be replaced with any rational function on the curve with a pole at O. In particular, the x-coordinate on the Lyness curve (8) has a pole at O. Since, from (12), any sequence of iterates (u_n, u_{n+1}) of the Lyness map (4), satisfying the recurrence

$$u_{n+2}u_n = a\,u_{n+1} + b, \tag{15}$$

corresponds to a sequence of points $\mathcal{P}_n = \mathcal{P}_0 + n\mathcal{P}$ lying on a curve (8) with a value of K fixed by $\mathcal{P}_0 = (u_0, u_1)$ and $\mathcal{P} = (\infty, -a)$, we can implement the ECM by choosing an orbit that starts with $\mathcal{P}_0 = O = (\infty, \infty)$.

The point (∞, ∞) is not a suitable initial value for the affine map (4), but by using the isomorphism with a Weierstrass curve, as in Theorem 1, which identifies the point (v, ξ) on (9) with $\mathcal{P}$ on (8), or by using elliptic divisibility sequences, as mentioned in [19], we can compute the first few multiples of $\mathcal{P}$ as

$$\mathcal{P} = (\infty, -a) = (u_1, u_2), \quad 2\mathcal{P} = (-a, 0) = (u_2, u_3),$$

$$3\mathcal{P} = (0, -b/a) = (u_3, u_4),$$

and

$$4\mathcal{P} = \left(-\frac{b}{a}, -a - \frac{b(Ka+b)}{a(a^2-b)}\right) = (u_4, u_5) \tag{16}$$

The points $O, \pm\mathcal{P}, \pm 2\mathcal{P}, \pm 3\mathcal{P}$ are precisely the base points in the pencil (6), where the Lyness map is undefined, but the point $4\mathcal{P}$ (which depends on the value of K) is a suitable starting point for the iteration.

In terms of the choice of elliptic curve data, there are two ways to implement the ECM using the Lyness map: one can pick a Weierstrass curve (9) defined over $\mathbb{Q}$ (most conveniently, with $A, B \in \mathbb{Z}$) together with a choice of rational point $(x', y') = (v, \xi)$, and then use the birational equivalence in Theorem 1 to find the corresponding point $\mathcal{P}$ on a Lyness curve with parameters specified by (10); or instead, one can just pick the parameters a, b, K at random and proceed to calculate $s\mathcal{P}$ starting from the point $4\mathcal{P}$ given by (16). In fact, as already mentioned, it suffices to set $a \to 1$ before carrying out the iteration, since orbits with other values of a are equivalent to the case $a = 1$ by rescaling. In the first case, starting with a point on a Weierstrass cubic, one can calculate a, b, K from (10) and then replace these values by $1, b/a^2, K/a$, respectively; while in the second case it is sufficient to set $a = 1$ and just choose b, K at random, or (even more simply) just pick b, u_5 at random and then iterate from the point $4\mathcal{P} = (-b, u_5)$.

In order to have an efficient implementation of scalar multiplication, one should use an addition chain to calculate $s\mathcal{P}$ from $4\mathcal{P}$ by a sequence of addition steps $n\mathcal{P} \mapsto (n+1)\mathcal{P}$, corresponding to (4), and doubling steps $n\mathcal{P} \mapsto 2n\mathcal{P}$, corresponding to (13), so that $s\mathcal{P}$ can be obtained in a time $O(\log s)$. One can also subtract $\mathcal{P}$ using the inverse map

$$\varphi^{-1}: \quad (x, y) \mapsto \left(\frac{ax+b}{y}, x\right). \tag{17}$$

The affine maps φ and ψ are not computationally efficient because they both involve costly inversions (**I**), but inversions can be avoided by working with projective coordinates, as is commonly done with Montgomery curves using the Montgomery ladder [6, 10], or with twisted Edwards curves in EECM-MPFQ [3]. In the ECM this means that the only arithmetic needed is multiplication (**M**), squaring (**S**), multiplication by constants (**C**), and addition in $\mathbb{Z}/N\mathbb{Z}$. These operations are listed in order of decreasing cost: **S** is cheaper than **M**, multiplication by constants is even cheaper and may be neglected if they are suitably small, while the cost of addition is negligible compared with the rest.

Table 1: 2-Processor Lyness addition

Cost	Step	Processor 1	Processor 2
1**C**	1	$R_1 \leftarrow a \cdot Y$	$R_2 \leftarrow b \cdot Z$
	2	$R_1 \leftarrow R_1 + R_2$	*idle*
	3	$X^* \leftarrow Y$	$W^* \leftarrow Z$
1**M**	4	$Y^* \leftarrow W \cdot R_1$	$Z^* \leftarrow X \cdot Z$

For an addition chain starting from $4\mathcal{P}$, we may write

$$s = 2^{k_m}(2^{k_{m-1}}(\cdots(2^{k_1}(4+\delta_0)+\delta_1)\cdots)+\delta_{m-1})+\delta_m, \tag{18}$$

corresponding to δ_0 steps of adding $\mathcal{P}$, followed by k_1 doubling steps, then $|\delta_1|$ steps of adding or subtracting $\mathcal{P}$, etc. To avoid the base points we require $\delta_0 \geq 0$, and typically one might restrict to $\delta_j = \pm 1$ for $1 \leq j \leq m-1$, with $\delta_m = 0$ or ± 1, if subtraction of $\mathcal{P}$ is used, or only allow addition of $\mathcal{P}$ and take $0 \leq \delta_0 \leq 3$, $\delta_j = 1$ for $1 \leq j \leq m-1$ and $\delta_m = 0$ or 1 only. So for instance we could use $28 = 2^2 \times (2 \times 4 - 1)$ in the former case ($m = 2$, $\delta_0 = \delta_2 = 0$, $\delta_1 = -1$, $k_1 = 1$, $k_2 = 2$), or $2^2 \times (4 + 1 + 1 + 1)$ in the latter ($m = 1$, $\delta_0 = 3$, $\delta_1 = 0$, $k_1 = 2$). As we shall see, the cost of each projective addition or subtraction step is so low that using both addition and subtraction as much as possible may lead to savings in the total number of operations: finding an optimal addition/subtraction chain for Lyness scalar multiplication is an interesting open problem for future research.

To work with projective coordinates in $\mathbb{P}^1 \times \mathbb{P}^1$, we write the sequence of points generated by (15) as

$$n\mathcal{P} = (u_n, u_{n+1}) = \left(\frac{X_n}{W_n}, \frac{X_{n+1}}{W_{n+1}}\right),$$

and then each addition of $\mathcal{P}$ or doubling can be written as a polynomial map for the quadruple

$$(X, W, Y, Z) = (X_n, W_n, X_{n+1}, W_{n+1}),$$

where an addition step sends

$$(X_n, W_n, X_{n+1}, W_{n+1}) \mapsto (X_{n+1}, W_{n+1}, X_{n+2}, W_{n+2}),$$

and doubling sends

$$(X_n, W_n, X_{n+1}, W_{n+1}) \mapsto (X_{2n}, W_{2n}, X_{2n+1}, W_{2n+1}).$$

Taking projective coordinates in $\mathbb{P}^1 \times \mathbb{P}^1$, the Lyness map (4) becomes

$$\Big((X : W), (Y : Z)\Big) \mapsto \Big((X^* : W^*), (Y^* : Z^*)\Big), \tag{19}$$

where

$$X^* = Y, \;\; W^* = Z, \;\; (Y^* : Z^*) = ((aY + bZ)W : XZ)$$

with a included for completeness. If we set $a \to 1$ for convenience then each addition step, adding the point $\mathcal{P}$ using (19), requires $2\mathbf{M} + 1\mathbf{C}$, that is, two multiplications plus a multiplication by the constant parameter b. One can also try to choose b to be small enough, so that the effective cost reduces to $2\mathbf{M}$. If one wishes to include subtraction of $\mathcal{P}$, i.e. $n\mathcal{P} \mapsto (n-1)\mathcal{P}$, then this is achieved

using the projective version of the inverse (17), for which the cost is the same as for φ.

The doubling map ψ for the Lyness case, given by the affine map (13) with R defined by (14), lifts to the projective version

$$\Big((X : W), (Y : Z)\Big) \mapsto \Big((\hat{X} : \hat{W}), (\hat{Y} : \hat{Z})\Big), \tag{20}$$

where

$$\hat{X} = A_1B_1, \quad \hat{Y} = A_2B_2, \quad \hat{W} = C_1D_1, \quad \hat{Z} = C_2D_2,$$

with

$$\begin{array}{ll} A_1 = A_+ + A_-, & A_2 = A_+ - A_-, \\ B_1 = B_+ + B_-, & B_2 = B_+ - B_-, \\ C_1 = 2XT, & C_2 = -2YT, \\ D_1 = ZA_2 + C_2, & D_2 = WA_1 + C_1, \\ A_+ = 2G - aS - 2H', & A_- = aT, \\ B_+ = S(G - a^2H - H') - 2aHH', & S = E + F, \\ B_- = T(G - a^2H + H'), & T = E - F, \\ E = XZ,\ F = YW,\ G = XY, & H = WZ,\ H' = bH. \end{array}$$

Setting $a \to 1$ once again for convenience, and using the above formulae, we see that doubling can be achieved with $15\mathbf{M} + 1\mathbf{C}$, or $15\mathbf{M}$ if multiplication by b is ignored. (Note that multiplication by 2 is equivalent to addition: $2X = X + X$.)

We can illustrate the application of the ECM via the Lyness map with a simple example, taking

$$N = 3595474639,\ s = 28,\ a = 1,\ b = -u_4 = 2,\ u_5 = 17.$$

From (16) this means that

$$K = \left(1 - \frac{a^2}{b}\right)(u_5 + a) - \frac{b}{a} = 7,$$

but we shall not need this. Writing s as $28 = 2^2(2 \times 4 - 1)$, we compute $28\mathcal{P}$ via the chain $4\mathcal{P} \mapsto 8\mathcal{P} \mapsto 7\mathcal{P} \mapsto 14\mathcal{P} \mapsto 28\mathcal{P}$. As initial projective coordinates, we start with the quadruple

$$(X_4, W_4, X_5, W_5) = (-2, 1, 17, 1),$$

and then after one projective doubling step using (20), the quadruple (X_8, W_8, X_9, W_9) is found to be

$$(3595467431, 43928, 80648, 3595455259).$$

To obtain $7\mathcal{P}$ we use the projective version of the inverse map (17), which gives

$$X_{n-1} = (aX_n + bW_n)W_{n+1}, \qquad W_{n-1} = X_{n+1}W_n$$

for any n, so we get

$$(X_7, W_7) = (2032516399, 3542705344).$$

Then applying doubling to the quadruple (X_7, W_7, X_8, W_8) we find that $(X_{14}, W_{14}, X_{15}, W_{15})$ is

$$(160913035, 3261908647, 3049465821, 760206673),$$

and one final doubling step produces the projective coordinates of $28\mathcal{P}$, that is $(X_{28}, W_{28}, X_{29}, W_{29})$ given by

$$(558084862, 1754538456, 252369828, 1216214157).$$

Now we compute $\gcd(W_{28}, N) = 6645979$, and the method has succeeded in finding a prime factor of N. The projective coordinate W_{29} has the same common factor with N, but here we do not need the coordinates X_{29}, W_{29} at the final step; but if the method had failed then these would be needed for stage 2 of the ECM (computing multiples $\ell s\mathcal{P}$ for small primes ℓ).

It is worth comparing Lyness scalar multiplication with the most efficient state of the art method, which uses twisted Edwards curves, given by

$$ax^2 + y^2 = 1 + dx^2y^2, \tag{21}$$

with projective points in $\mathbb{P}^2$, or with extended coordinates in $\mathbb{P}^3$: with standard projective points, adding a generic pair of points uses $10\mathbf{M}+1\mathbf{S}+2\mathbf{C}$, while doubling uses only $3\mathbf{M}+4\mathbf{S}+1\mathbf{C}$ [3]; while with extended Edwards it is possible to achieve $8\mathbf{M}+1\mathbf{C}$ for addition of two points, or just $8\mathbf{M}$ in the case $a = -1$, and $4\mathbf{M} + 4\mathbf{S} + 1\mathbf{C}$ for doubling [18].

Clearly addition using the Lyness map is extremely efficient, compared with other methods. In contrast, Lyness doubling is approximately twice as costly as doubling with Edwards curves. Moreover, using (19) only allows addition of $\mathcal{P}$ to any other point, rather than adding an arbitrary pair of points, which would be much more costly using a projective version of (11). Since any addition chain is asymptotically dominated by doubling, with roughly as many doublings as the number of bits of s, this means that, without any further simplification of the projective formulae, scalar multiplication with Lyness curves should use on average roughly twice as many multiplications per bit as with twisted Edwards curves.

However, as we shall see, using ideas from [18], it is possible to make Lyness scalar multiplication much more efficient if parallel processors are used, as described in the next section.

4 DOUBLING IN PARALLEL

In [18] it was shown that if four processors are used in parallel in the case $a = -1$ of twisted Edwards curves (21), then with extended coordinates in $\mathbb{P}^3$ each addition step can be achieved with an algorithm that has an effective cost of only $2\mathbf{M} + 1\mathbf{C}$, reducing to just $2\mathbf{M}$ if the constant d is small - an improvement in speed by a full factor of 4 better than the sequential case, while doubling can be achieved with an effective cost of just $1\mathbf{M} + 1\mathbf{S}$. (Similarly, versions of these algorithms with two processors give an effective speed increase by a factor of 2.) Practical details of implementing the ECM in parallel with different types of hardware are discussed in [4].

Using two parallel processors, based on (19), each projective addition or subtraction step can be carried out in parallel with an effective cost of just $1\mathbf{M} + 1\mathbf{C}$. An algorithm with two processors is presented in Table 1 (where the parameter a has been included for reasons of symmetry, but can be set to 1). Spreading the addition step over four processors does not lead to any saving in cost.

For Lyness curves, the large amount of symmetry in the doubling formula (13) means that its projective version (20) can naturally be distributed over four processors in parallel, resulting in the algorithm presented in Table 2. This means that each Lyness doubling step is achieved with an effective cost of $4\mathbf{M} + 1\mathbf{C}$, or just $4\mathbf{M}$ if b is small.

In an addition chain (18) for Lyness, starting from $4\mathcal{P}$ with intermediate $\delta_j = \pm 1$, each step of adding or subtracting $\mathcal{P}$ is followed by a doubling. Thus a combined addition-doubling or subtraction-doubling step can be carried out in parallel with four processors,

Table 2: 4-Processor Lyness doubling

Cost	Step	Processor 1	Processor 2	Processor 3	Processor 4
1**M**	1	$R_1 \leftarrow X \cdot Z$	$R_2 \leftarrow Y \cdot W$	$R_3 \leftarrow X \cdot Y$	$R_4 \leftarrow W \cdot Z$
1**C**	2	$R_5 \leftarrow R_1 + R_2$	$R_6 \leftarrow R_1 - R_2$	$R_7 \leftarrow b \cdot R_4$	*idle*
1**M**	3	$R_1 \leftarrow X \cdot R_6$	$R_2 \leftarrow Y \cdot R_6$	$R_8 \leftarrow R_4 \cdot R_7$	$R_9 \leftarrow R_3 - R_7$
	4	$R_1 \leftarrow 2R_1$	$R_2 \leftarrow -2R_2$	$R_3 \leftarrow R_3 + R_7$	$R_{10} \leftarrow 2R_9$
	5	$R_3 \leftarrow R_3 - R_4$	$R_7 \leftarrow R_{10} - R_5$	$R_8 \leftarrow 2R_8$	$R_{11} \leftarrow R_9 - R_4$
	6	$R_9 \leftarrow R_7 + R_6$	$R_{10} \leftarrow R_7 - R_6$	*idle*	*idle*
1**M**	7	$R_3 \leftarrow R_3 \cdot R_6$	$R_4 \leftarrow W \cdot R_9$	$R_7 \leftarrow Z \cdot R_{10}$	$R_{11} \leftarrow R_{11} \cdot R_5$
	8	$R_5 \leftarrow R_2 + R_7$	$R_6 \leftarrow R_1 + R_4$	$R_{11} \leftarrow R_{11} - R_8$	*idle*
	9	$R_7 \leftarrow R_{11} + R_3$	$R_8 \leftarrow R_{11} - R_3$	*idle*	*idle*
1**M**	10	$\hat{X} \leftarrow R_7 \cdot R_9$	$\hat{W} \leftarrow R_1 \cdot R_5$	$\hat{Y} \leftarrow R_8 \cdot R_{10}$	$\hat{Z} \leftarrow R_2 \cdot R_6$

resulting in an effective cost of 5**M** + 2**C**, but no cost saving is achieved by combining them.

It is also clear that the algorithm in Table 2 can be adapted to the case of two processors in parallel. This leads to an effective cost of 8**M** + 1**C** per Lyness doubling.

Thus we have seen that implementing scalar multiplication in the ECM with Lyness curves can be made efficient if implemented in parallel with two or four processors. In the concluding section that follows we weigh up the pros and cons of using Lyness curves for scalar multiplication, and briefly mention other contexts where they may be useful.

5 CONCLUSIONS

We have presented an algorithm for scalar multiplication using Lyness curves, which can be applied to any rational point on a Weierstrass curve defined over $\mathbb{Q}$, and have shown how it can be implemented more efficiently in parallel with four processors.

Each step of addition (or subtraction) of a special point $\mathcal{P}$, based on the Lyness map, has a remarkably low cost: only 2**M** + 1**C** if carried out sequentially, or an effective cost of just 1**M** + 1**C** in parallel with two processors. The record for elliptic curve addition in [18] using twisted Edwards curves (21) with the special parameter choice $a = -1$ requires 8**M**, or an effective cost of 2**M** with four parallel processors; but this is for adding an arbitrary pair of points, whereas for Lyness we can only achieve such a low cost by adding/subtracting the special point $\mathcal{P}$. Nevertheless, for the purposes of scalar multiplication, addition/subtraction of $\mathcal{P}$ and doubling is all that is required.

At 15**M** + 1**C**, the cost of sequential Lyness doubling is much higher, and essentially twice the cost of sequential doubling with twisted Edwards curves [3]. Since asymptotically scalar multiplication is dominated by doubling steps, it appears that on average using Lyness curves for scalar multiplication should require about twice as many multiplications per bit compared with the twisted Edwards version.

However, if it is performed in parallel with four processors, as in Table 2, then the effective cost of Lyness doubling is reduced to 4**M** + 1**C**, and this becomes only 4**M** in the case that the parameter b is small. This is still higher than the speed record for doubling with four processors (1**M** + 1**S**), which is achieved in [18] with the $a = -1$ case of twisted Edwards curves. Nevertheless, performing Lyness addition and doubling in parallel is still quite efficient, and may have other possible advantages, which we now consider.

For the ECM it is desirable to have a curve with large torsion over $\mathbb{Q}$, since for an unknown prime $p|N$ this increases the probability of smoothness of the group order $\#E(\mathbb{F}_p)$ in the Hasse interval $[p+1-2\sqrt{p}, p+1+2\sqrt{p}]$, making success more likely. Twisted Edwards curves, which are birationally equivalent to Montgomery curves, do not cover all possible elliptic curves over $\mathbb{Q}$. In particular, it is known from [3] that for twisted Edwards curves with the special parameter choice $a = -1$ (which gives the fastest addition step) the torsion subgroups $\mathbb{Z}/10\mathbb{Z}$, $\mathbb{Z}/12\mathbb{Z}$, $\mathbb{Z}/2\mathbb{Z} \times \mathbb{Z}/8\mathbb{Z}$ are not possible, nor is $\mathbb{Z}/2\mathbb{Z} \times \mathbb{Z}/6\mathbb{Z}$ possible for any choice of a.

For Lyness curves (8), there is no such restriction on the choice of torsion subgroups interesting to look for families of Lyness curves having large torsion and rank at least one, employing a combination of empirical and theoretical approaches similar to [1, 2].

Another potentially useful feature of scalar multiplication with Lyness curves is that, since there is no loss of generality in setting $a \to 1$, it requires only the two parameters b, K (or, perhaps better, b, u_5) to be carried out, and these at the same time fix an elliptic curve E and a point $\mathcal{P} \in E$. Moreover, both parameters can be chosen small. This parsimony is aesthetically pleasing because the moduli space of elliptic curves with a marked point is two-dimensional.

On the other hand, if one wishes to start from a given Weierstrass curve (9) with a point on it, then in general the formula in (10) produces a Lyness curve with a value of $a \neq 1$, so if the other parameters are subsequently rescaled to fix $a \to 1$ then in general the requirement of smallness will need to be sacrificed for the new parameter b so obtained.

We have concentrated on scalar multiplication in stage 1 of the ECM, but for stage 2 one usually computes $\ell_1 s\mathcal{P}, \ell_2 \ell_1 s\mathcal{P}$, etc. for a sequence of primes $\ell_1, \ell_2, \ldots$ all smaller than some bound B_2. This can be carried out effectively using a baby-step-giant-step method [3], requiring addition of essentially arbitrary multiples of $\mathcal{P}$. For the latter approach, using addition with the Lyness map has the disadvantage that one can only add $\mathcal{P}$ at each step, so to add some other multiple of $\mathcal{P}$ one would need to redefine the parameters a, b, K (and then rescale $a \to 1$ if desired), leading to extra intermediate computations.

Scalar multiplication is an essential feature of elliptic curve cryptography: in particular, it is required for Alice and Bob to perform

the elliptic curve version of Diffie-Hellman key exchange [23]. In that context, one requires a curve $E(\mathbb{F}_q)$ with non-smooth order, to make the discrete logarithm problem as hard as possible. Bitcoin uses the arithmetic of the curve $y^2 = x^3 + 7$, known as secp256k1, which is not isomorpic to a twisted Edwards curve. Also, the sequence of scalar multiples of a point on an elliptic curve over a finite field or a residue ring can be used for pseudorandom number generation; the fact that the cost of addition of a point is so low for Lyness curves may make them particularly well suited to this. It would be interesting to see if Lyness curves can offer any advantages in these and other cryptographic settings.

ACKNOWLEDGMENTS

The research of the author is funded by fellowship EP/M004333/1 from the EPSRC and grant IEC\R3\193024 from the Royal Society. Thanks to the School of Mathematics and Statistics, UNSW for hosting him as a Visiting Professorial Fellow twice during 2017-2019 with funding from the Distinguished Researcher Visitor Scheme, to John Roberts and Wolfgang Schief for providing additional financial support, and to Reinout Quispel, Igor Shparlinski and the anonymous reviewers for their helpful comments.

REFERENCES

[1] Razvan Barbulescu, Joppe W. Bos, Cyril Bouvier, Thorsten Kleinjung, and Peter L. Montgomery. 2013. Finding ECM-friendly curves through a study of Galois properties. *The Open Book Series* 1, 1 (Nov. 2013), 63–86. https://doi.org/10.2140/obs.2013.1.63

[2] Daniel J. Bernstein, Peter Birkner, , and Tanja Lange. 2010. Starfish on Strike. In *Progress in Cryptology – LATINCRYPT 2010*, Michel Abdalla and Paulo S. L. M. Barreto (Eds.). Springer, Berlin, Heidelberg, 61–80. https://doi.org/10.1007/978-3-642-14712-8_4

[3] Daniel J. Bernstein, Peter Birkner, Tanja Lange, and Christiane Peters. 2013. ECM using Edwards curves. *Math. Comput.* 82, 282 (Apr. 2013), 1139–1179. https://doi.org/10.1090/S0025-5718-2012-02633-0

[4] Daniel J. Bernstein, Tien-Ren Chen, Chen-Mou Cheng, Tanja Lange, and Bo-Yin Yang. 2009. ECM on Graphics Cards. In *Advances in Cryptology - EUROCRYPT 2009*, Antoine Joux (Ed.). Springer, Berlin, Heidelberg, 483–501. https://doi.org/10.1007/978-3-642-01001-9_28

[5] Daniel J. Bernstein and Tanja Lange. 2007. Faster Addition and Doubling on Elliptic Curves. In *Advances in Cryptology – ASIACRYPT 2007*, Kaoru Kurosawa (Ed.). Springer, Berlin, Heidelberg, 29–50. https://doi.org/10.1007/978-3-540-76900-2_3

[6] Daniel J. Bernstein and Tanja Lange. 2017. *Montgomery Curves and the Montgomery Ladder.* Cambridge University Press, 82–115. https://doi.org/10.1017/9781316271575.005

[7] Frits Beukers and Richard Cushman. 1998. Zeeman's monotonicity conjecture. *J. Differ. Equ.* 143, 1 (Feb. 1998), 191–200. https://doi.org/10.1006/jdeq.1997.3359

[8] Jérémy Blanc. 2013. Symplectic birational transformations of the plane. *Osaka J. Math.* 50, 2 (June 2013), 573–590. https://doi.org/10.18910/25084

[9] Richard P. Brent, Alexander Kruppa, and Paul Zimmermann. 2017. *FFT Extension for Algebraic-Group Factorization Algorithms.* Cambridge University Press, 189–205. https://doi.org/10.1017/9781316271575.009

[10] Craig Costello and Benjamin Smith. 2018. Montgomery curves and their arithmetic. *J. Cryptogr. Eng.* 8 (Sept. 2018), 227–240. https://doi.org/10.1007/s13389-017-0157-6

[11] Harold Coxeter. 1971. Frieze patterns. *Acta Arithmetica* 18, 1 (1971), 297–310. http://eudml.org/doc/204992

[12] Richard Crandall and Carl Pomerance. 2010. *Prime Numbers - A Computational Perspective* (2nd ed.). Springer, New York.

[13] Johannes J. Duistermaat. 2005. *Discrete Integrable Systems: QRT Maps and Elliptic Surfaces.* Springer-Verlag, New York.

[14] Harold M. Edwards. 2007. A normal form for elliptic curves. *Bull. Amer. Math. Soc.* 44, 3 (July 2007), 393–422. https://doi.org/10.1090/S0273-0979-07-01153-6

[15] J. Esch and Thomas D. Rogers. 2001. The screensaver map: dynamics on elliptic curves arises from polygonal folding. *Discrete Comput. Geom.* 25 (Apr. 2001), 477–502. https://doi.org/10.1007/s004540010075

[16] Sergey Fomin and Andrei Zelevinsky. 2003. *Y*-systems and generalized associahedra. *Ann. Math. (2)* 158, 3 (Nov. 2003), 977–1018. https://doi.org/10.4007/annals.2003.158.977

[17] Henriette Heer, Gary McGuire, and Oisín Robinson. 2016. JKL-ECM: an implementation of ECM using Hessian curves. *LMS Journal of Computation and Mathematics* 19, A (2016), 83–99. https://doi.org/10.1112/S1461157016000231

[18] Huseyin Hisil, Kenneth Koon-Ho Wong, Gary Carter, and Ed Dawson. 2008. Twisted Edwards Curves Revisited. In *Advances in Cryptology - ASIACRYPT 2008*, Josef Pieprzyk (Ed.). Springer Berlin Heidelberg, Berlin, Heidelberg, 326–343.

[19] Andrew N. W. Hone. 2020. ECM factorization with QRT maps. (2020). arXiv:2001.09076

[20] Apostolos Iatrou and John A. G. Roberts. 2001. Integrable mappings of the plane preserving biquadratic invariant curves. *Journal of Physics A: Mathematical and General* 34, 34 (Aug. 2001), 6617–6636. https://doi.org/10.1088/0305-4470/34/34/308

[21] Apostolos Iatrou and John A. G. Roberts. 2002. Integrable mappings of the plane preserving biquadratic invariant curves II. *Nonlinearity* 15, 2 (Feb. 2002), 459–489. https://doi.org/10.1088/0951-7715/15/2/313

[22] H. W. jun. Lenstra. 1987. Factoring integers with elliptic curves. *Ann. Math. (2)* 126 (1987), 649–673. https://doi.org/10.2307/1971363

[23] Neal Koblitz. 1998. *Algebraic Aspects of Cryptography.* Springer, Berlin, Heidelberg.

[24] Robert C. Lyness. 1961. 2952. Cycles. *Math. Gaz.* 45, 353 (Oct. 1961), 207–209. https://doi.org/10.2307/3612778

[25] Peter L. Montgomery. 1987. Speeding the Pollard and elliptic curve methods of factorization. *Math. Comput.* 48, 177 (Jan. 1987), 243–264. https://doi.org/10.1090/S0025-5718-1987-0866113-7

[26] Tomoki Nakanishi. 2011. Periodicities in cluster algebras and dilogarithm identities. In *Representations of Algebras and Related Topics (EMS Series of Congress Reports)*, Vol. 5. European Mathematical Society, Zurich, 407–444.

[27] G. Reinout W. Quispel, John A.G. Roberts, and Colin John Thompson. 1989. Integrable mappings and soliton equations II. *Physica D* 34, 1 (Jan. 1989), 183–192. https://doi.org/10.1016/0167-2789(89)90233-9

[28] G. Reinout W. Quispel, John A. G. Roberts, and Colin John Thompson. 1988. Integrable mappings and soliton equations. *Phys. Lett. A* 126, 7 (Jan. 1988), 419–421. https://doi.org/10.1016/0375-9601(88)90803-1

[29] Dinh T. Tran, Peter H. van der Kamp, and G. Reinout W. Quispel. 2010. Sufficient number of integrals for the pth-order Lyness equation. *J. Phys. A: Math. Theor.* 43, 30 (June 2010), 302001. https://doi.org/10.1088/1751-8113/43/30/302001

[30] Teruhisa Tsuda. 2004. Integrable mappings via rational elliptic surfaces. *J. Phys. A: Math. Gen.* 37, 7 (Feb. 2004), 2721–2730. https://doi.org/10.1088/0305-4470/37/7/014

[31] Alexei B. Zamolodchikov. 1991. On the thermodynamic Bethe ansatz equations for reflectionless ADE scattering theories. *Physics Letters B* 253, 3 (Jan. 1991), 391 – 394. https://doi.org/10.1016/0370-2693(91)91737-G

Algorithmic Averaging for Studying Periodic Orbits of Planar Differential Systems

Bo Huang
LMIB-School of Mathematical Sciences, Beihang University
Beijing, China
Courant Institute of Mathematical Sciences, New York University
New York, USA
bohuang0407@buaa.edu.cn

ABSTRACT

One of the main open problems in the qualitative theory of real planar differential systems is the study of limit cycles. In this article, we present an algorithmic approach for detecting how many limit cycles can bifurcate from the periodic orbits of a given polynomial differential center when it is perturbed inside a class of polynomial differential systems via the averaging method. We propose four symbolic algorithms to implement the averaging method. The first algorithm is based on the change of polar coordinates that allows one to transform a considered differential system to the normal form of averaging. The second algorithm is used to derive the solutions of certain differential systems associated to the unperturbed term of the normal of averaging. The third algorithm exploits the partial Bell polynomials and allows one to compute the integral formula of the averaged functions at any order. The last algorithm is based on the aforementioned algorithms and determines the exact expressions of the averaged functions for the considered differential systems. The implementation of our algorithms is discussed and evaluated using several examples. The experimental results have extended the existing relevant results for certain classes of differential systems.

CCS CONCEPTS

• **Computing methodologies** → Symbolic and algebraic manipulation; • **Symbolic and algebraic algorithms** → Symbolic calculus algorithms.

KEYWORDS

Algorithmic approach; averaging method; limit cycles; planar differential systems; periodic orbits

ISSAC '20, July 20–23, 2020, Kalamata, Greece

ACM ISBN 978-1-4503-7100-1/20/07…$15.00
https://doi.org/10.1145/3373207.3404064

ACM Reference Format:
Bo Huang. 2020. Algorithmic Averaging for Studying Periodic Orbits of Planar Differential Systems. In *International Symposium on Symbolic and Algebraic Computation (ISSAC '20), July 20–23, 2020, Kalamata, Greece.* ACM, New York, NY, USA, 8 pages. https://doi.org/10.1145/3373207.3404064

1 INTRODUCTION

We deal with polynomial differential systems in $\mathbb{R}^2$ of the form

$$\frac{dx}{dt} = \dot{x} = f_n(x, y), \quad \frac{dy}{dt} = \dot{y} = g_n(x, y), \tag{1.1}$$

where n is the maximum degree of the polynomials f and g. As we knew, the second part of the 16th Hilbert's problem [19, 25] asks for "the maximal number $H(n)$ and relative configurations of limit cycles" for the differential system (1.1). Here $H(n)$ is called the Hilbert number. The problem is still open even for $n = 2$. However, there have been many interesting results on the lower bound of $H(n)$ for $n \geq 2$: it is shown in [5, 49] that $H(2) \geq 4$ and $H(3) \geq 13$ in [29]. In [8], it is proved that $H(n)$ grows at least as rapidly as $n^2 \log n$. For the latest development about $H(n)$, we refer the reader to [6, 16, 30].

We recall that a limit cycle of the differential system (1.1) is an isolated periodic orbit of the system. One of the best ways of producing limit cycles is by perturbing a differential system which has a center. In this case the perturbed system displays limit cycles that bifurcate, either from the center (having the so-called Hopf bifurcation), or from some of the periodic orbits surrounding the center, see the book of Christopher-Li [6] and the references cited therein.

Usually, a limit cycle which bifurcates from a center equilibrium point is called a *small amplitude limit cycle*, and a *medium amplitude limit cycle* is one which bifurcates from a periodic orbit surrounding a center (see [33, 36]). Note that the notation of "large" limit cycle may occur in several situations in the literature, see [18, 54]. In the past seven decades, many researchers have considered the small amplitude limit cycles and obtained many results (e.g., [1, 28, 41, 50, 53]). Over the years, a number of algebraic methods and algorithms have been developed (e.g., [13, 17, 46, 51, 52]) based on the tools of Liapunov constants or Melnikov function.

In our recent work [24], we provide an algorithmic approach to small amplitude limit cycles of nonlinear differential systems by the averaging method, and give an upper bound of

the number of zeros of the averaged functions for the general class of perturbed differential systems ([24], Thm. 3.1). The goal of this paper is to extend our algorithmic approach to study the maximal number of medium amplitude limit cycles that bifurcate from some periodic orbits surrounding the centers of the unperturbed systems. The main technique is based on the general form of the averaging method for planar differential systems.

The method of averaging is an important tool to study the existence of isolated periodic solutions of nonlinear differential systems in the presence of a small parameter. It can be used to find a lower bound of the Hilbert number $H(n)$ for certain differential systems. The method has a long history that started with the classical works of Lagrange and Laplace, who provided an intuitive justification of the method. The first formalization of this theory was done in 1928 by Fatou. Important practical and theoretical contributions to the averaging method were made in the 1930's by Bogoliubov-Krylov, and in 1945 by Bogoliubov. The ideas of averaging method have extended in several directions for finite and infinite dimensional differentiable systems. For a modern exposition of this subject, see the books of Sanders-Verhulst-Murdock [48] and Llibre-Moeckel-Simó [37].

The averaging method provides a straightforward calculation approach to determine the number of limit cycles that bifurcate from some periodic orbits of the regarded particular class of differential systems. However, in practice, the evaluation of the averaged functions is a computational problem that requires powerful computerized resources. Moreover, the computational complexity grows very fast with the averaging order. Our objective in this paper is to present an algorithmic approach to develop the averaging method at any order and to further study the number of medium amplitude limit cycles for nonlinear differential systems.

In general, to obtain analytically periodic solutions of a differential system is a very difficult problem, many times a problem impossible to solve. As we shall see when we can apply the averaging method, this difficult problem is reduced to finding the zeros of a nonlinear function in an open interval of $\mathbb{R}$, i.e., now the problem has the same difficulty as the problem of finding the singular or equilibrium points of a differential system.

The structure of our paper is as follows. In Section 2, we introduce the basic results on the averaging method for planar differential systems. We give our algorithms and briefly describe their implementation in Maple in Section 3. Its application is illustrated in Section 4 using several examples including a class of generalized Kukles polynomial differential systems and certain differential systems with uniform isochronous centers of degrees 3 and 4. Finally, a conclusion is provided in Section 5. The Maple code of the algorithms can be download from https://github.com/Bo-Math/limit-cycle.

In view of space limitation, we put the full text of the paper in the arXiv https://arxiv.org/pdf/2005.03487.pdf.

2 MAIN RESULTS OF THE AVERAGING METHOD

In this section we introduce the basic theory of the averaging method. We consider the following polynomial differential system of degree n_1

$$\dot{x} = P(x, y), \quad \dot{y} = Q(x, y) \tag{2.1}$$

having a center at the point $\bar{x} \in \mathbb{R}^2$. Without loss of generality we can assume that the center $\bar{x}$ of system (2.1) is the origin of coordinates. The following definition is due to Poincaré ([4], Sect. 2).

Definition 2.1. We say that an isolated singular point $\bar{x}$ of (2.1) is a *center* if there exists a neighbourhood of $\bar{x}$, such that every orbit in this neighbourhood is a cycle surrounding $\bar{x}$.

Remark 2.2. Determining the conditions on the parameters under which the origin for system (2.1) is a center is the well-known *center problem*, see [42, 47]. There are many partial results for the centers of system (2.1) of degree $n_1 \geq 2$. Unfortunately, at present, we are very far from obtaining the classification of all the centers of cubic polynomial differential systems. In general, the huge number of computations necessary for obtaining complete classification becomes the central problem which is computationally intractable, see for instances, [15] and the references cited therein.

Now consider the perturbations of (2.1) of the form

$$\begin{aligned} \dot{x} &= P(x, y) + p(x, y, \varepsilon), \\ \dot{y} &= Q(x, y) + q(x, y, \varepsilon), \end{aligned} \tag{2.2}$$

where the polynomials p, q are of degree at most n_2 (usually $n_2 \geq n_1$) in x and y, and ε is a small parameter. We are interested in the maximum number of medium amplitude limit cycles of (2.2) for $|\varepsilon| > 0$ sufficiently small, which bifurcate from some periodic orbits surrounding the centers of system (2.1).

Usually, the averaging method deals with planar differential systems in the following normal form

$$\frac{dr}{d\theta} = \sum_{i=0}^{k} \varepsilon^i F_i(\theta, r) + \varepsilon^{k+1} R(\theta, r, \varepsilon), \tag{2.3}$$

where $F_i : \mathbb{R} \times D \to \mathbb{R}$ for $i = 0, 1, \ldots, k$, and $R : \mathbb{R} \times D \times (-\varepsilon_0, \varepsilon_0) \to \mathbb{R}$ are $\mathcal{C}^k$ functions, 2π-periodic in the first variable, being D an open and bounded interval of $(0, \infty)$, and ε_0 is a small parameter. As one of the main hypotheses, it is assumed that $r(\theta, z)$ is a 2π-periodic solution of the unperturbed differential system $dr/d\theta = F_0(\theta, r)$, for every initial condition $r(0, z) = z \in D$.

The averaging method consists in defining a collection of functions $f_i : D \to \mathbb{R}$, called the i-th order averaged functions, for $i = 1, 2, \ldots, k$, which control (their simple zeros control), for ε sufficiently small, the isolated periodic solutions of the differential system (2.3). In Llibre-Novaes-Teixeira [39] it has been established that

$$f_i(z) = \frac{y_i(2\pi, z)}{i!}, \tag{2.4}$$

where $y_i : \mathbb{R}\times D \to \mathbb{R}$, for $i = 1, 2, \ldots, k$, is defined recursively by the following integral equations

$$y_1(\theta, z) = \int_0^\theta \Big(F_1(s, r(s, z)) + \partial F_0(s, r(s, z))y_1(s, z)\Big)ds,$$

$$y_i(\theta, z) = i! \int_0^\theta \Bigg(F_i(s, r(s, z)) + \sum_{\ell=1}^{i} \sum_{S_\ell} \frac{1}{b_1!b_2!2!^{b_2}\cdots b_\ell!\ell!^{b_\ell}} \cdot \partial^L F_{i-\ell}(s, r(s, z)) \prod_{j=1}^{\ell} y_j(s, z)^{b_j}\Bigg)ds, \tag{2.5}$$

where S_ℓ is the set of all ℓ-tuples of nonnegative integers $[b_1, b_2, \ldots, b_\ell]$ satisfying $b_1 + 2b_2 + \cdots + \ell b_\ell = \ell$ and $L = b_1 + b_2 + \cdots + b_\ell$. Here, $\partial^L F(\theta, r)$ denotes the Fréchet's derivative of order L with respect to the variable r.

In [14, 39] the averaging method at any order was developed to study isolated periodic solutions of nonsmooth but continuous differential systems. Recently, the averaging method has also been extended to study isolated periodic solutions of discontinuous differential systems; see [35, 38]. The following k-th order averaging theorem is proved in Llibre-Novaes-Teixeira [39].

THEOREM 2.3. *Assume that the following conditions hold:*

(a) for each $i = 0, 1, \ldots, k$ and $\theta \in \mathbb{R}$, the function $F_i(\theta, \cdot)$ is of class $\mathcal{C}^{k-i}$, $\partial^{k-i}F_i$ is locally Lipschitz in the second variable, and $R(\theta, \cdot, \varepsilon)$ is a continuous function locally Lipschitz in the second variable;

(b) $f_i \equiv 0$ for $i = 1, 2, \ldots, j-1$ and $f_j \neq 0$ with $j \in \{1, 2, \ldots, k\}$;

(c) for some $z^* \in D$ with $f_j(z^*) = 0$, there exists a neighborhood $V \subset D$ of z^* such that $f_j(z) \neq 0$ for all $z \in \bar{V}\backslash\{z^*\}$, and that $d_B(f_j(z), V, 0) \neq 0$.

Then, for $|\varepsilon| > 0$ sufficiently small, there exists a 2π-periodic solution $r_\varepsilon(\theta)$ of (2.3) such that $r_\varepsilon(0) \to z^*$ when $\varepsilon \to 0$.

Remark 2.4. The above symbol d_B denotes the Browder degree; see Browder [2] for a general definition. When f_j is a $\mathcal{C}^1$ function and the derivative of f_j at $z \in V$ is distinct from zero (i.e., $f_j'(z) \neq 0$), then in this case, $f_j'(z^*) \neq 0$ implies $d_B(f_j(z), V, 0) \neq 0$.

Recently in [45] the partial Bell polynomials were used to provide a relatively simple alternative formula for the recurrence (2.5). Since the Bell polynomials are implemented in algebraic manipulators as Maple and Mathematica, this new formula can make easier the computational implementation of the averaged functions. In this paper, we will exploit this new formula in our algorithmic approach for solving the problem of evaluating the recurrence (2.5) (see Section 3.2). In the sequel, for ℓ and m positive integers, we recall the Bell polynomials:

$$B_{\ell,m}(x_1, \ldots, x_{\ell-m+1}) = \sum_{\tilde{S}_{\ell,m}} \frac{\ell!}{b_1!b_2!\cdots b_{\ell-m+1}!} \prod_{j=1}^{\ell-m+1} \left(\frac{x_j}{j!}\right)^{b_j},$$

where $\tilde{S}_{\ell,m}$ is the set of all $(\ell - m + 1)$-tuples of nonnegative integers $[b_1, b_2, \ldots, b_{\ell-m+1}]$ satisfying $b_1 + 2b_2 + \cdots + (\ell - m + 1)b_{\ell-m+1} = \ell$, and $b_1 + b_2 + \cdots + b_{\ell-m+1} = m$.

The following result is an equivalent formulation of the integral equation (2.5) via above Bell polynomials, its proof can be found in [45].

THEOREM 2.5. *For $i = 1, 2, \ldots, k$ the recursive equation (2.5) reads*

$$y_1(\theta, z) = Y(\theta, z) \int_0^\theta Y(s, z)^{-1} F_1(s, r(s, z))ds,$$

$$y_i(\theta, z) = Y(\theta, z) \int_0^\theta Y(s, z)^{-1} \Bigg(i!F_i(s, r(s, z)) + \sum_{m=2}^{i} \partial^m F_0(s, r(s, z)) B_{i,m}\big(y_1(s, z), \ldots, y_{i-m+1}(s, z)\big) + \sum_{\ell=1}^{i-1} \sum_{m=1}^{\ell} \frac{i!}{\ell!} \partial^m F_{i-\ell}(s, r(s, z)) B_{\ell,m}\big(y_1(s, z), \ldots, y_{\ell-m+1}(s, z)\big)\Bigg)ds, \tag{2.6}$$

where $Y(\theta, z)$ is the fundamental solution of the variational equation $Y' = \partial F_0(\theta, r(\theta, z))Y$ satisfying the initial condition $Y(0, z) = 1$.

The general study of the exact number of simple zeros of the averaged functions (2.4) up to every order is also very difficult to be done, since the averaged functions may be too complicated, such as including square root functions, logarithmic functions, and the elliptic integrals. In the literature there is an abundance of papers dealing with zeros of the averaged functions (see, for instance, [20–23, 32, 40] and references therein). Note that one can estimate the size of bifurcated limit cycles by using the expressions of the averaged functions. In fact we know that if the averaged functions $f_j = 0$ for $j = 1, \ldots, k-1$ and $f_k \neq 0$, and $\bar{z} \in D$ is a simple zero of f_k, then by Theorem 2.3 there is a limit cycle $r_\varepsilon(\theta)$ of differential system (2.3) such that $r_\varepsilon(\theta) = r(\theta, \bar{z}) + \mathcal{O}(\varepsilon)$. Then, going back through the changes of variables we have for the differential system (2.2) the limit cycle $(x(t, \varepsilon), y(t, \varepsilon)) = (r(\theta, \bar{z})\cos\theta, r(\theta, \bar{z})\sin\theta) + \mathcal{O}(\varepsilon)$.

3 ALGORITHMIC AVERAGING FOR THE STUDY OF LIMIT CYCLES

The process of using the averaging method for studying limit cycles of differential systems can be divided into three steps ([24], Sect. 4).

STEP 1. Write the perturbed system (2.2) in the normal form of averaging (2.3) up to k-th order in ε.

STEP 2. (i) Compute the exact formula for the k-th order integral function $y_k(\theta, z)$ in (2.6). (ii) Derive the symbolic expression of the k-th order averaged function $f_k(z)$ by (2.4).

STEP 3. Determine the exact upper bound of the number of simple zeros of $f_k(z)$ for $z \in D$.

In the following subsections we will present algorithms to implement the first two steps. We use "Maple-like" pseudo-code, based on our Maple implementation. Using these algorithms we reduce the problem of studying the number of limit cycles of system (2.2) to the problem of detecting **STEP 3**.

3.1 Algorithm for transformation into normal form

In this subsection we will devise an efficient algorithm which can be used to transform system (2.2) into the form (2.3).

We first describe the underlying equations before presenting the algorithm. Doing the change of polar coordinates $x = rC$, $y = rS$ with $C = \cos\theta$ and $S = \sin\theta$, then we can transform system (2.2) into the following form

$$\begin{aligned}\frac{dr}{d\theta} &= \frac{dr/dt}{d\theta/dt} = \frac{r(x\dot{x}+y\dot{y})}{x\dot{y}-y\dot{x}}\Big|_{x=rC,y=rS}\\ &= r\frac{C\left(P(x,y)+p(x,y,\varepsilon)\right)+S\left(Q(x,y)+q(x,y,\varepsilon)\right)}{C\left(Q(x,y)+q(x,y,\varepsilon)\right)-S\left(P(x,y)+p(x,y,\varepsilon)\right)}\Big|_{x=rC,y=rS}\\ &= r\frac{\frac{CP(x,y)+SQ(x,y)}{CQ(x,y)-SP(x,y)}+\frac{Cp(x,y,\varepsilon)+Sq(x,y,\varepsilon)}{CQ(x,y)-SP(x,y)}}{1+\frac{Cq(x,y,\varepsilon)-Sp(x,y,\varepsilon)}{CQ(x,y)-SP(x,y)}}\Big|_{x=rC,y=rS}\\ &= F_0(\theta,r)+\varepsilon F_1(\theta,r)+\ldots+\varepsilon^k F_k(\theta,r)+\mathcal{O}(\varepsilon^{k+1}).\end{aligned} \tag{3.1}$$

The last equality is obtained by carrying the order $k+1$ Taylor series expansion of the penultimate equality, with respect to the variable ε, around the point $\varepsilon = 0$. The first algorithm **NormalForm**, presented below, is a direct implementation of the formula derivation in (3.1).

Algorithm 1 NormalForm(P, Q, p, q, k)

Input: a perturbed system (2.2) with an order $k \geq 0$ in (2.3)
Output: an expression of $dr/d\theta$ up to k-th order in ε

1: $d1 := \text{normal}(\text{subs}(x = r\cdot C, y = r\cdot S, x\cdot(P+p)+y\cdot(Q+q))/r)$;
2: $d2 := \text{normal}(\text{subs}(x = r\cdot C, y = r\cdot S, x\cdot(Q+q)-y\cdot(P+p))/r^2)$;
3: $T := \text{taylor}(d1/d2, \varepsilon = 0, k+1)$;
4: $H := \text{convert}\,(T, \text{polynom})$;
5: $F_0 := \text{coeff}(\varepsilon \cdot H, \varepsilon)$;
6: **for** i **from** 1 **to** k **do**
7: $\quad c_i := \text{coeff}(H, \varepsilon^i)$;
8: $\quad F_{i,1} := \text{prem}\left(\text{numer}(c_i), C^2+S^2-1, S\right)$;
9: $\quad F_{i,2} := \text{prem}\left(\text{denom}(c_i), C^2+S^2-1, S\right)$;
10: $\quad F_i := F_{i,1}/F_{i,2}$;
11: $dr/d\theta := \text{subs}(C = \cos\theta, S = \sin\theta, F_0 + \sum_{j=1}^{k} F_j\varepsilon^j)$;
12: **return** $dr/d\theta$;

In line 8 the function $\text{prem}(a, b, x)$ is the pseudo-remainder of a with respect to b in the variable x. By the property of the pseudo-remainder we know that the degree in S is at most 1 of the polynomials $F_{i,1}$ and $F_{i,2}$.

3.2 Algorithms for computing formulae and functions of averaging

This subsection is devoted to provide effective algorithms to compute the formula and exact expression of the k-th order averaged function. According to Theorem 2.5, we should take the following substeps to compute the k-th order averaged function of system (2.3).

Substep 1. Determine the open and bounded interval D, the 2π-periodic solution $r(\theta, z)$ of the unperturbed system $dr/d\theta = F_0(\theta, r)$ with initial condition $r(0, z) = z \in D$, and the fundamental solution $Y(\theta, z)$ of the variational equation $Y' = \partial F_0(\theta, r(\theta, z))Y$ with initial condition $Y(0, z) = 1$.

Substep 2. Compute the exact formula for the k-th order integral function $y_k(\theta, z)$.

Substep 3. Output the symbolic expression for the k-th order averaged function $f_k(z)$ (simplified by using the conditions for $f_1 \equiv f_2 \equiv \cdots \equiv f_{k-1} \equiv 0$) for a given differential system (2.2).

We provide each of the substep an algorithm. For the Substep 1 we first derive the 2π-periodic solution $r(\theta, z)$, and then use it to further obtain the interval D and the fundamental solution $Y(\theta, z)$.

Algorithm 2 DSolutions(F_0)

Input: the unperturbed term F_0 in (2.3)
Output: $r(\theta, z)$, a set of inequalities (SIs) with respect to z, and $Y(\theta, z)$

1: $\text{de1} := \text{diff}(r(\theta), \theta) = \text{subs}(r = r(\theta), F_0)$;
2: $\text{ans1} := \textbf{dsolve}(\{\text{de1}, r(0) = z\}, r(\theta))$;
3: $r(\theta, z) := \text{op}(2, \text{ans1})$;
4: $\text{minvalue} := \textbf{minimize}(r(\theta, z), \theta = 0..2\pi)$;
5: $m := \text{nops}([\text{op}(\text{minvalue})])$;
6: $SIs := \{\}$;
7: **for** i **from** 1 **to** m **do**
8: $\quad SIs := SIs$ **union** $\{\text{op}(i, \text{minvalue}) > 0\}$;
9: $\text{de2} := \text{diff}(Y(\theta), \theta) = \text{subs}(r = r(\theta, z), \text{diff}(F_0, r)) \cdot Y(\theta)$;
10: $\text{ans2} := \textbf{dsolve}(\{\text{de2}, Y(0) = 1\}, Y(\theta))$;
11: $Y(\theta, z) := \text{op}(2, ans2)$;
12: **return** $[r(\theta, z),\ SIs,\ Y(\theta, z)]$;

Remark 3.1. The output results of $r(\theta, z)$ and $Y(\theta, z)$ can be reduced by using the identity $\sin^2\theta + \cos^2\theta = 1$ so that the degree of what are left in $\sin\theta$ is at most 1. We use the routine **dsolve** built-in Maple for solving an ordinary differential equation. We remark that the unperturbed term $F_0(\theta, r)$ is usually a rational trigonometric function in r, $\sin\theta$ and $\cos\theta$. As far as we know, we do not have a systematic approach to the solution of the differential equation $dr/d\theta = F_0(\theta, r)$ in the general case. In Section 4 we will consider certain classes of differential systems with uniform isochronous centers to illustrate the effectiveness of our algorithm. It is important to emphasize that the interval D can be determined by using the output set SIs. Since the original system may contain some parameters, the resulting set SIs could be parametric. In order to derive the interval D in this case, we will construct an equivalent solution set $\overline{SIs}$ of SIs that contains only the rational polynomial inequalities, and then use the **SemiAlgebraic** command in Maple to compute the solutions. Below we provide a concrete example to show the feasibility this algorithm, one may check the results in [32]. More experiments can be found in Section 4.

Example 1. Consider the following quintic polynomial differential system

$$\dot{x} = -y + x^2 y(x^2 + y^2), \quad \dot{y} = x + xy^2(x^2 + y^2). \tag{3.2}$$

Applying our algorithm **NormalForm** for $p = q = k = 0$, we have $dr/d\theta = r^5 \cos\theta \sin\theta$. Then applying the algorithm **DSolutions** we obtain a list $[r(\theta, z),\ SIs,\ Y(\theta, z)]$, where

$$r(\theta, z) = \frac{z}{(2z^4(\cos^2\theta - 1) + 1)^{1/4}},$$
$$SIs = \left\{0 < z, 0 < \frac{z}{(-2z^4 + 1)^{1/4}}\right\},$$
$$Y(\theta, z) = \frac{1}{(2z^4(\cos^2\theta - 1) + 1)^{5/4}}.$$

To obtain the interval D in this case, we construct an equivalent solution set $\overline{SIs}$ of SIs that contains only the rational polynomials: $\overline{SIs} := \{0 < z, 0 < \frac{z}{-2z^4+1}\}$. Then using the Maple command **SolveTools[SemiAlgebraic]**, we compute the solution of the set $\overline{SIs}$, and obtain that $D = \{0 < z < 2^{-1/4}\}$.

We want to say that the expressions of the returned results on $r(\theta, z)$ and $Y(\theta, z)$ may be complicated, such as including square root functions, and exponential functions. Below we give a simple example to show this, one may find the related results in [31].

Example 2. Consider the following polynomial differential system

$$\dot{x} = -y(3x^2 + y^2), \quad \dot{y} = x(x^2 - y^2). \tag{3.3}$$

The normal form $dr/d\theta = -2r\cos\theta\sin\theta$ can be obtained by the algorithm **NormalForm**, and the algorithm **DSolutions** returns a list $\left[ze^{\cos^2\theta - 1}, \left\{0 < z, 0 < ze^{-1}\right\}, e^{\cos^2\theta - 1}\right]$.

For the Substep 2, we present our algorithm **AveragingFormula**. This algorithm can be used to compute the exact formula of the k-th order integral function $y_k(\theta, z)$. Correctness of it follows from Theorem 2.5.

Algorithm 3 AveragingFormula(k)

Input: an order $k \geq 1$ of the normal form (2.3)
Output: the integral function $y_k(\theta, z)$

1: $T_1 := 0$; $T_2 := 0$;
2: **for** m **from** 2 **to** k **do**
3: $T_1 := T_1 + \mathrm{Diff}(F_0(s, r(s, z)), r\$m) \cdot$ **IncompleteBellB**$(k, m, y_1(s, z), \ldots, y_{k-m+1}(s, z))$;
4: **for** ℓ **from** 1 **to** $k - 1$ **do**
5: **for** m **from** 1 **to** ℓ **do**
6: $T_2 := T_2 + \frac{k!}{\ell!} \cdot \mathrm{Diff}(F_{k-\ell}(s, r(s, z)), r\$m) \cdot$ **IncompleteBellB**$(\ell, m, y_1(s, z), \ldots, y_{\ell-m+1}(s, z))$;
7: $y_k(\theta, z) := Y(\theta, z) \cdot \mathrm{Int}\left(Y^{-1}(s, z) \cdot (k! \cdot F_k(s, r(s, z)) + T_1 + T_2), s = 0..\theta\right)$;
8: **return** $y_k(\theta, z)$;

We deduce explicitly the formulae of y_k's up to $k = 5$ in Appendix A. In fact our algorithm can compute arbitrarily high order formulae of y_k's. In Section 4, we will study several differential systems to show the feasibility of our algorithm.

In the last subsection, we provide an algorithm **NormalForm** to transform system (2.2) into the form $dr/d\theta$. The algorithm **DSolutions** admits one to obtain the fundamental solutions $r(\theta, z)$, $Y(\theta, z)$ and the interval D (Substep 1). The algorithm **AveragedFunction**, presented below, is based on the algorithms **NormalForm**, **DSolutions** and Theorem 2.5, which provides a straightforward calculation method to derive the exact expression of the k-th order averaged function for a given differential system in the form (2.2) (Substep 3).

Algorithm 4 AveragedFunction$(dr/d\theta, r(\theta, z), Y(\theta, z), k)$

Input: a normal formal of averaging (3.1) with an order $k \geq 1$ and the fundamental solutions $r(\theta, z)$, $Y(\theta, z)$
Output: a list of expressions of the averaged functions

1: $F_0 := \mathrm{coeff}(\varepsilon \cdot (dr/d\theta), \varepsilon)$;
2: **for** j **from** 1 **to** k **do**
3: $F_j := \mathrm{coeff}(dr/d\theta, \varepsilon^j)$;
4: $A_j :=$ **AFormula**(j);
5: $H_j := \mathrm{normal}\left(\frac{1}{Y(\theta,z)} \cdot \mathrm{expand}(\mathrm{subs}(r = r(\theta, z), \mathrm{value}(A_j)))\right)$;
6: $H_{j,1} := \mathrm{collect}(\mathrm{expand}(\mathrm{numer}(H_j)), \{\cos\theta, \sin\theta\}, \mathrm{distributed})$;
7: $H_{j,2} := \mathrm{denom}(H_j)$;
8: **for** h **from** 1 **to** $\mathrm{nops}(H_{j,1})$ **do**
9: $g_{j,h} := \mathrm{int}\left(\frac{\mathrm{op}(h, H_{j,1})}{H_{j,2}}, \theta = 0..\theta, \mathrm{AllSolutions}\right)$;
10: $s_{j,h} := \mathrm{int}\left(\frac{\mathrm{op}(h, H_{j,1})}{H_{j,2}}, \theta = 0..2\pi\right)$;
11: $y_j := Y(\theta, z) \cdot \mathrm{sum}(g_{j,t}, t = 1..\mathrm{nops}(H_{j,1}))$;
12: $f_j := \frac{1}{j!} \cdot \mathrm{sum}(s_{j,t}, t = 1..\mathrm{nops}(H_{j,1}))$;
13: **return** $[y_k,\ f_k]$;

In line 4, the routine **AFormula** is a subalgorithm we use for the generation of the expression in the parenthesis of equation (2.6) without dependence on (s, z). The detailed information of this subalgorithm is as follows.

Subalgorithm: AFormula
INPUT: An averaging order $k \geq 1$;
OUTPUT: The expression in the parenthesis of equation (2.6) without dependence on (s, z).
STEP 0. $U = 0$; $V = 0$;
STEP 1. **For** m **from** 2 **to** k do
$U := U + \mathrm{Diff}(F_0, r\$m) \cdot$ **IncompleteBellB**$(k, m, \mathrm{seq}(y_i, i = 1..k - m + 1))$; **end do**;
STEP 2. **For** ℓ **from** 1 **to** $k - 1$ do
for m **from** 1 **to** ℓ do
$V := V + \frac{k!}{\ell!} \cdot \mathrm{Diff}(F_{k-\ell}, r\$m) \cdot$ **IncompleteBellB**$(\ell, m, \mathrm{seq}(y_i, i = 1..\ell - m + 1))$; **end do**; **end do**;
STEP 3. Output $k!F_k + U + V$.

Remark 3.2. In order to obtain an exact and simplified expression of the averaged function, one should make some assumptions (e.g., the interval D on z, and possible conditions on the parameters that may appear in the original differential systems) before preforming the algorithm **AveragedFunction**. For more details see our experiments in Section 4. We also remark that, throughout the computation,

an assumption on θ (i.e., $\theta \in (2\pi-\epsilon, 2\pi+\epsilon)$ with ϵ a small number) was made to identify a valid branch of the possible returned piecewise functions (in line 9), since the integral functions $y_i(\theta, z)$ for $i = 1, \ldots, k$ evaluate at the point $\theta = 2\pi$ in (2.4).

We implemented all the algorithms presented in this section in Maple. In the next section, we will apply our general algorithmic approach to analyze the bifurcation of limit cycles for several concrete differential systems.

4 IMPLEMENTATION AND EXPERIMENTS

In this section we demonstrate our algorithmic tests using several examples. We present the bifurcation of limit cycles for a class of generalized Kukles polynomial differential systems as an illustration of our approach explained in previous sections. In addition, we study the number of limit cycles that bifurcate from some periodic solutions surrounding the isochronous centers for certain differential systems by the first and second order averaging method. The obtained results of our experiments extend the existing relevant results and show the feasibility of our approach.

4.1 A class of generalized Kukles differential systems

In this subsection we consider a very particular case of the 16th Hilbert problem; we study the number of limit cycles of the generalized Kukles polynomial differential system

$$\dot{x} = -y, \quad \dot{y} = x + Q(x, y), \tag{4.1}$$

where $Q(x, y)$ is a polynomial with real coefficients of degree n. This system was introduced by Kukles in [28], examining the conditions under which the origin of the system

$$\dot{x} = -y,$$
$$\dot{y} = x + a_1x^2 + a_2xy + a_3y^2 + a_4x^3 + a_5x^2y + a_6xy^2 + a_7y^3$$

is a center. For long time, it had been thought that the conditions given by Kukles were necessary and sufficient conditions, but some new cases have been found, see [7, 27].

Here we are interested in studying the maximum number of limit cycles that bifurcate from the periodic orbits of the linear center $\dot{x} = -y$, $\dot{y} = x$, perturbed inside the following class of generalized Kukles polynomial differential systems

$$\begin{aligned} \dot{x} &= -y + \sum_{k\geq 1} \varepsilon^k l_m^k(x), \\ \dot{y} &= x - \sum_{k\geq 1} \varepsilon^k \left(f_{n_1}^k(x) + g_{n_2}^k(x)y + h_{n_3}^k(x)y^2 + d_0^k y^3 \right), \end{aligned} \tag{4.2}$$

where for every k the polynomials $l_m^k(x)$, $f_{n_1}^k(x)$, $g_{n_2}^k(x)$, and $h_{n_3}^k(x)$ have degree m, n_1, n_2, and n_3 respectively, $d_0^k \neq 0$ is a real number and ε is a small parameter. This question has been studied in [43] for $k = 1, 2$, and the authors obtained the following result.

THEOREM 4.1. *Assume that for $k = 1, 2$ the polynomials $l_m^k(x)$, $f_{n_1}^k(x)$, $g_{n_2}^k(x)$, and $h_{n_3}^k(x)$ have degree m, n_1, n_2, and n_3 respectively, with m, n_1, n_2, $n_3 \geq 1$, and $d_0^k \neq 0$ is a real number. Then for ε sufficiently small the maximum number of limit cycles of the Kukles polynomial system* (4.2) *bifurcating from the periodic orbits of the linear center $\dot{x} = -y$, $\dot{y} = x$,*

1. *is $\max\left\{\left[\frac{m-1}{2}\right], \left[\frac{n_2}{2}\right], 1\right\}$ by using the first order averaging method;*
2. *is $\max\left\{\left[\frac{n_1}{2}\right]+\left[\frac{n_2-1}{2}\right], \left[\frac{n_1}{2}\right]+\left[\frac{m}{2}\right]-1, \left[\frac{n_1+1}{2}\right], \left[\frac{n_3+3}{2}\right], \left[\frac{n_3}{2}\right]+\left[\frac{m}{2}\right], \left[\frac{n_2+1}{2}\right]+\left[\frac{n_3}{2}\right], \left[\frac{n_2}{2}\right], \left[\frac{m-1}{2}\right], \left[\frac{n_1-1}{2}\right]+\mu, \left[\frac{n_3+1}{2}\right]+\mu, 1\right\}$ by using the second order averaging method, where $\mu = \min\left\{\left[\frac{m-1}{2}\right], \left[\frac{n_2}{2}\right]\right\}$.*

Here, $[\cdot]$ denotes the integer part function. Remark that, many researchers have discussed the bifurcation of limit cycles for generalized Kukles polynomial differential system in the form (4.1). We refer the readers to [34, 44] for some interesting results on this subject. The next result extends Theorem 4.1 to arbitrary order of averaging.

LEMMA 4.2. *Let $\max\{m, n_1, n_2+1, n_3+2\} = N \geq 3$, then the Kukles polynomial system* (4.2) *for ε sufficiently small has no more than $[k(N-1)/2]$ limit cycles bifurcating from the periodic orbits of the linear center $\dot{x} = -y$, $\dot{y} = x$, using the averaging method up to order k.*

PROOF. This result follows directly from Theorem 6 in [14]. □

In what follows, using our algorithms we will do some experimental results by fixing some values of the degrees in system (4.2). Note that the maximum numbers of limit cycles in Theorem 4.1 and Lemma 4.2 may not be reached. The following corollary shows that these maximum numbers can be reached for some orders of averaging.

COROLLARY 4.3. *(i) When $m = 3$, $n_1 = 3$, $n_2 = 2$, and $n_3 = 1$, the maximum number of limit cycles of the Kukles polynomial system* (4.2) *bifurcating from the periodic orbits of the linear center $\dot{x} = -y$, $\dot{y} = x$, using the fifth order averaging method is five and it is reached.*

(ii) When $m = 5$, $n_1 = 1$, $n_2 = 2$, and $n_3 = 1$, the maximum number of limit cycles of the Kukles polynomial system (4.2) *bifurcating from the periodic orbits of the linear center $\dot{x} = -y$, $\dot{y} = x$, using the fourth order averaging method is five and it is reached.*

The detailed proof of the first statement of Corollary 4.3 can be found in Appendix B. Since the calculations and arguments of the second part are quite similar to those used in the first one, we omit the proof of statement (ii) in Corollary 4.3. More concretely, we provide in Table 1 the maximum number of limit cycles for system (4.2) in each case of Corollary 4.3 up to the k-th order averaging method for $k = 1, \ldots, 5$.

The number of limit cycles in statement (i) can be reached for each order of averaging. That is to say, the bound given in Lemma 4.2 is sharp for the case in statement (i). However, for the statement (ii), the bound given in Lemma 4.2 is only sharp for the first order of averaging. We note also that

Table 1: Number of limit cycles of system (4.2) in Corollary 4.3

Averaging order	Statement (i)	Statement (ii)
1	1	2
2	2	2
3	3	4
4	4	5
5	5	-

for each statement in Corollary 4.3, the bound provided in Theorem 4.1 can be reached up to the second order.

Remark 4.4. The calculation of the high order averaged function f_k involves heavy computations with complicated expressions. It may not work effectively when one of the degrees (m and n_i, $i = 1, 2, 3$) is large. It turns out that we can greatly improve the speed by updating the obtained $dr/d\theta$ using the conditions on the parameters of $f_1 \equiv f_2 \equiv \cdots \equiv f_{k-1} = 0$.

4.2 Limit cycles for certain differential systems with uniform isochronous centers

Recall that a center $\bar{x}$ of system (2.1) is an *isochronous center* if it has a neighborhood such that in this neighborhood all the periodic orbits have the same period. An isochronous center is *uniform* if in polar coordinates $x = r\cos\theta$, $y = r\sin\theta$, it can be written as $\dot{r} = G(\theta, r)$, $\dot{\theta} = \eta$, $\eta \in \mathbb{R}\backslash\{0\}$, see Conti [10] for more details. The next result on the uniform isochronous center (UIC) is well-known, a proof of it can be found in [26].

Proposition 4.5. *Assume that system* (2.1) *has a center at the origin $\bar{x}$. Then $\bar{x}$ is a UIC if and only if by doing a linear change of variables and a rescaling of time the system can be written as*

$$\dot{x} = -y + xf(x, y), \quad \dot{y} = x + yf(x, y), \tag{4.3}$$

where f is a polynomial in x and y of degree $n-1$, and $f(0,0) = 0$.

In what follows, we recall some important results on the UICs of planar cubic and quartic differential systems. The following result due to Collins [9] in 1997, also obtained by Devlin, Lloyd and Pearson [11] in 1998, and by Gasull, Prohens and Torregrosa [12] in 2005 characterizes the UICs of cubic polynomial systems.

Theorem 4.6. *A planar cubic differential system has a UIC at the origin if and only if it can be written as system* (4.3) *with $f(x, y) = a_1x + a_2y + a_3x^2 + a_4xy - a_3y^2$ satisfying that $a_1^2a_3 - a_2^2a_3 + a_1a_2a_4 = 0$. Moreover, this planar cubic differential system can be reduced to either one of the following two forms:*

$$\dot{x} = -y + x^2y, \quad \dot{y} = x + xy^2, \tag{4.4}$$

$$\dot{x} = -y + x^2 + Ax^2y, \quad \dot{y} = x + xy + Axy^2, \tag{4.5}$$

where $A \in \mathbb{R}$.

Systems (4.4) and (4.5) are known as *Collins First Form* and *Collins Second Form*, respectively. See ([33], Thm. 9) for more details of the global phase portraits of the Collins forms.

The following characterization of planar quartic polynomial differential systems with an isolated UIC at the origin is provided by Chavarriga, García and Giné [3], in 2001.

Theorem 4.7. *A planar quartic differential system has a UIC at the origin if and only if it can be written as*

$$\begin{aligned} \dot{x} &= -y + x\left(ax + bxy + cx^3 + dxy^2\right), \\ \dot{y} &= x + y\left(ax + bxy + cx^3 + dxy^2\right), \end{aligned} \tag{4.6}$$

where $a, b, c, d \in \mathbb{R}$.

A classification of the global phase portraits of the quartic differential systems of the form (4.6) is provided in [26].

In order to save space, we put the remaining results in Appendix C.

5 CONCLUSION

We have presented a systematical approach to analyze how many limit cycles of differential system (2.2) can bifurcate from the periodic orbits of an unperturbed one via the averaging method. We designed four algorithms to analyze the averaging method and shown that the general study of the number of limit cycles of system (2.2) can be reduced to the problem of estimating the number of simple zeros of the obtained averaged functions with the aid of these algorithms.

Our algorithms admit a generalization to the case of studying the bifurcation of limit cycles for discontinuous differential systems. It would be interesting to employ our approach to analyze the bifurcation of limit cycles for differential systems in many different fields, which are of high interest in nature sciences and engineering. It will be beneficial to generalize our current approach to the case of higher dimension differential systems by using the general form of the averaging method. We leave this as the future research problems.

In addition, we noticed the phenomenon of tremendous growth of expressions in intermediate calculations while we done experiments for the linear center $\dot{x} = -y$, $\dot{y} = x$ by using the high order of averaging. For the nonlinear polynomial differential centers, the evaluation of the high order averaged functions is highly nontrivial; the main difficulty exists in the technical and cumbersome computations of some complicated integral equations. How to simplify and optimize the steps of the computations of the averaged functions is also a question that remains for further investigation.

ACKNOWLEDGMENTS

Huang's work is partially supported by China Scholarship Council under Grant No.: 201806020128. The author is grateful to Professor Chee Yap and Professor Dongming Wang for their profound concern and encouragement. The author thanks the referees for their valuable comments and suggestions to improve the presentation of this paper.

REFERENCES

[1] Nikolai N. Bautin. 1954. On the number of limit cycles which appear with the variation of the coefficients from an equilibrium position of focus or center type. *Amer. Math. Soc. Transl.* 100 (1954), 397–413.

[2] Felix E. Browder. 1983. Fixed point theory and nonlinear problems. *Bull. Amer. Math. Soc.* 9 (1983), 1–39.

[3] Javier Chavarriga, Isaac García, and Jaume Giné. 2001. On the integrability of differential equations defined by the sum of homogeneous vector fields with degenerate infinity. *Int. J. Bifur. Chaos* 11 (2001), 711–722.

[4] Javier Chavarriga and Marco Sabatini. 1999. A survey of isochronous centers. *Qual. Theory Dyn. Syst.* 1 (1999), 1–70.

[5] Lan S. Chen and Ming S. Wang. 1979. The relative position and the number of limit cycles of a quadratic diffenertial system. *Acta Math. Sinica* 22 (1979), 751–758.

[6] Colin J. Christopher and Cheng Z. Li. 2007. *Limit Cycles of Differential Equations*. Birkhäuser, Boston.

[7] Colin J. Christopher and Noel G. Lloyd. 1990. On the paper of Jin and Wang concerning the conditions for a centre in certain cubic systems. *Bull. London Math. Soc.* 22 (1990), 5–12.

[8] Colin J. Christopher and Noel G. Lloyd. 1995. Polynomial systems: A lower bound for the Hilbert numbers. *Proc. R. Soc. Lond. Ser. A* 450 (1995), 218–224.

[9] Christopher B. Collins. 1997. Conditions for a centre in a simple class of cubic systems. *Differential Integral Equations* 10 (1997), 333–356.

[10] Roberto Conti. 1994. Uniformly isochronous centers of polynomial systems in $\mathbb{R}^2$. In Differential equations, dynamical systems, and control science. Lecture Notes in Pure and Appl. Math. 152, New York, 21–31.

[11] James Devlin, Noel G. Lloyd, and Jane M. Pearson. 1998. Cubic systems and Abel equations. *J. Differ. Equ.* 147 (1998), 435–454.

[12] Armengol Gasull, Rafel Prohens, and Joan Torregrosa. 2005. Limit cycles for rigid cubic systems. *J. Math. Anal. Appl.* 303 (2005), 391–404.

[13] Armengol Gasull and Joan Torregrosa. 2001. A new algorithm for the computation of the Lyapunov constants for some degenerated critical points. *Nonlin. Anal.* 47 (2001), 4479–4490.

[14] Jaume Giné, Maite Grau, and Jaume Llibre. 2013. Averaging theory at any order for computing periodic orbits. *Phys. D* 250 (2013), 58–65.

[15] Jaume Giné and Xavier Santallusia. 2004. Implementation of a new algorithm of computation of the Poincaré-Liapunov constants. *J. Comput. Appl. Math.* 166 (2004), 465–476.

[16] Mao A. Han and Ji B. Li. 2012. Lower bounds for the Hilbert number of polynomial systems. *J. Differ. Equ.* 252 (2012), 3278–3304.

[17] Mao A. Han, Jun M. Yang, and Pei Yu. 2009. Hopf bifurcation for near-Hamiltonian. *Int. J. Bifur. Chaos* 19 (2009), 4117–4130.

[18] Mao A. Han, Tong H. Zhang, and Hong Zang. 2004. On the number and distribution of limit cycles in a cubic system. *Int. J. Bifur. Chaos* 14 (2004), 4285–4292.

[19] David Hilbert. 1902. Mathematical problems. *Bull. Am. Math. Soc.* 8 (1902), 437–479.

[20] Bo Huang. 2017. Bifurcation of limit cycles from the center of a quintic system via the averaging method. *Int. J. Bifur. Chaos* 27 (2017), 1750072–1–16.

[21] Bo Huang. 2019. Limit cycles for a discontinuous quintic polynomial differential system. *Qual. Theory Dyn. Syst.* 18 (2019), 769–792.

[22] Bo Huang. 2020. On the limit cycles for a class of discontinuous piecewise cubic polynomial differential systems. *Electron. J. Qual. Theory Differ. Equ.* 25 (2020), 1–24.

[23] Bo Huang and Wei Niu. 2019. Limit cycles for two classes of planar polynomial differential systems with uniform isochronous centers. *J. Appl. Anal. Comput.* 9 (2019), 943–961.

[24] Bo Huang and Chee Yap. 2019. An algorithmic approach to limit cycles of nonlinear differential systems: the averaging method revisited. In Proc. ISSAC'19. ACM Press, New York, 211–218.

[25] Yulij S. Ilyashenko. 2002. Centennial history of Hilbert's 16th problem. *Bull. Am. Math. Soc.* 39 (2002), 301–354.

[26] Jackson Itikawa and Jaume Llibre. 2015. Phase portraits of uniform isochronous quartic centers. *J. Comput. Appl. Math.* 287 (2015), 98–114.

[27] Xiao F. Jin and Dong M. Wang. 1990. On the conditions of Kukles for the existence of a centre. *Bull. London Math. Soc.* 22 (1990), 1–4.

[28] Isaak S. Kukles. 1944. Sur quelques cas de distinction entre un foyer et un centre. *Dokl. Akad. Nauk. SSSR.* 42 (1944), 208–211.

[29] Cheng Z. Li, Chang J. Liu, and Jia Z. Yang. 2009. A cubic system with thirteen limit cycles. *J. Differ. Equ.* 246 (2009), 3609–3619.

[30] Ji B. Li. 2003. Hilbert's 16th problem and bifurcations of planar polynomial vector fields. *Int. J. Bifur. Chaos* 13 (2003), 47–106.

[31] Shi M. Li, Yu L. Zhao, and Zhao H. Sun. 2015. On the limit cycles of planar polynomial system with non-rational first integral via averaging method at any order. *Appl. Math. Comput.* 256 (2015), 876–880.

[32] Hai H. Liang, Jaume Llibre, and Joan Torregrosa. 2016. Limit cycles coming from some uniform isochronous centers. *Adv. Nonlinear Stud.* 16 (2016), 197–220.

[33] Jaume Llibre and Jackson Itikawa. 2015. Limit cycles for continuous and discontinuous perturbations of uniform isochronous cubic centers. *J. Comput. Appl. Math.* 277 (2015), 171–191.

[34] Jaume Llibre and Ana C. Mereu. 2011. Limit cycles for generalized Kukles polynomial differential systems. *Nonlin. Anal.* 74 (2011), 1261–1271.

[35] Jaume Llibre, Ana C. Mereu, and Douglas D. Novaes. 2015. Averaging theory for discontinuous piecewise differential systems. *J. Differ. Equ.* 258 (2015), 4007–4032.

[36] Jaume Llibre, Ana C. Mereu, and Marco A. Teixeira. 2010. Limit cycles of the generalized polynomial Liénard differential equations. *Math. Proc. Camb. Phil. Soc.* 148 (2010), 363–383.

[37] Jaume Llibre, Richard Moeckel, and Carles Simó. 2015. *Central Configurations, Periodic Orbits, and Hamiltonian Systems*. Birkhäuser, Basel.

[38] Jaume Llibre, Douglas D. Novaes, and Camila A.B. Rodrigues. 2017. Averaging theory at any order for computing limit cycles of discontinuous piecewise differential systems with many zones. *Phys. D* 353-354 (2017), 1–10.

[39] Jaume Llibre, Douglas D. Novaes, and Marco A. Teixeira. 2014. Higher order averaging theory for finding periodic solutions via Brouwer degree. *Nonlinearity* 27 (2014), 563–583.

[40] Jaume Llibre and Grzegorz Świrszcz. 2011. On the Limit cycles of polynomial vector fields. *Dyn. Contin. Discrete Impuls. Syst. Ser A: Math Anal.* 18 (2011), 203–214.

[41] Noel G. Lloyd. 1988. Limit cycles of polynomial systems-some recent developments. *London Math. Soc. Lecture Note Ser.* 127 (1988), 192–234.

[42] Adam Mahdi, Claudio Pessoa, and Jonathan D. Hauenstein. 2017. A hybrid symbolic-numerical approach to the center-focus problem. *J. Symb. Comput.* 82 (2017), 57–73.

[43] Nawal Mellahi, Amel Boulfoul, and Amar Makhlouf. 2019. Maximum number of limit cycles for generalized Kukles polynomial differential systems. *Differ. Equ. Dyn. Syst.* 27 (2019), 493–514.

[44] Ana C. Mereu, Regilene Oliveira, and Camila A.B. Rodrigues. 2018. Limit cycles for a class of discontinuous piecewise generalized Kukles differential systems. *Nonlin. Dyn.* 93 (2018), 2201–2212.

[45] Douglas D. Novaes. 2017. *An Equivalent Formulation of the Averaged Functions via Bell Polynomials*. Springer, New York, 141–145.

[46] Valery G. Romanovski. 1993. Calculation of Lyapunov numbers in the case of two pure imaginary roots. *Differ. Equ.* 29 (1993), 782–784.

[47] Valery G. Romanovski and Douglas S. Shafer. 2009. *The Center and Cyclicity Problems: A Computational Algebra Approach*. Birkhäuser, Boston.

[48] Jan A. Sanders, Ferdinand Verhulst, and James Murdock. 2007. *Averaging Methods in Nonlinear Dynamical Systems*. Springer, New York.

[49] Song L. Shi. 1980. A concrete example of the existence of four limit cycles for quadratic system. *Sci. Sinica* 23 (1980), 153–158.

[50] Dong M. Wang. 1990. A class of cubic differential systems with 6-tuple focus. *J. Differ. Equ.* 87 (1990), 305–315.

[51] Dong M. Wang. 1991. Mechanical manipulation for a class of differential systems. *J. Symb. Comput.* 12 (1991), 233–254.

[52] Pei Yu and Guan R. Chen. 2008. Computation of focus values with applications. *Nonlin. Dyn.* 51 (2008), 409–427.

[53] Pei Yu and Mao A. Han. 2005. Twelve limit cycles in a cubic case of the 16th Hilbert problem. *Int. J. Bifur. Chaos* 15 (2005), 2191–2205.

[54] Pei Yu and Mao A. Han. 2012. Four limit cycles from perturbing quadratic integrable systems by quadratic polynomials. *Int. J. Bifur. Chaos* 22 (2012), 1250254–1–28.

New Progress in Univariate Polynomial Root Finding

Rémi Imbach*
remi.imbach@nyu.edu
New York University

Victor Y. Pan†
victor.pan@lehman.cuny.edu
City University of New York

ABSTRACT

The recent advanced sub-division algorithm is nearly optimal for the approximation of the roots of a dense polynomial given in monomial basis; moreover, it works locally and slightly outperforms the user's choice MPSolve when the initial region of interest contains a small number of roots. Its basic and bottleneck block is counting the roots in a given disc on the complex plain based on Pellet's theorem, which requires the coefficients of the polynomial and expensive shift of the variable. We implement a novel method for both root-counting and exclusion test, which is faster, avoids the above requirements, and remains efficient for sparse input polynomials. It relies on approximation of the power sums of the roots lying in the disc rather than on Pellet's theorem. Such approximation was used by Schönhage in 1982 for the different task of deflation of a factor of a polynomial provided that the boundary circle of the disc is sufficiently well isolated from the roots. We implement a faster version of root-counting and exclusion test where we do not verify isolation and significantly improve performance of subdivision algorithms, particularly strongly in the case of sparse inputs. We present our implementation as heuristic and cite some relevant results on its formal support presented elsewhere.

KEYWORDS

Polynomial root finding, Subdivision, Root counting

ACM Reference Format:
Rémi Imbach and Victor Y. Pan. 2020. New Progress in Univariate Polynomial Root Finding. In *International Symposium on Symbolic and Algebraic Computation (ISSAC '20), July 20–23, 2020, Kalamata, Greece.* ACM, New York, NY, USA, 8 pages. https://doi.org/10.1145/3373207.3404063

1 INTRODUCTION

We seek complex roots of a degree d univariate polynomial p with real or complex coefficients. For a while the user choice for this problem has been the package `MPsolve` based on Erhlich-Aberth (simultaneous Newton-like) iterations. Their empirical global convergence (right from the start) is very fast, but its formal support is a long-known challenge, and the iterations approximate the roots in a fixed region of interest (ROI) about as slow as all complex roots.

In contrast, for the known algorithms subdividing a ROI, e.g., box, the cost of root-finding in a ROI decreases at least proportionally to the number of roots in it. Some recent subdivision algorithms have a proved nearly optimal complexity, are robust in the case of root clusters and multiple roots, and their implementation in [8] a little outperforms `MPsolve` for ROI containing only a small number of roots, which is an important benefit in many computational areas.

The Root Clustering Problem. $Z(\mathcal{S}, p)$ or $Z(\mathcal{S})$ is the root subset of p in a complex set $\mathcal{S}$; $\#(\mathcal{S}, p)$ or $\#(\mathcal{S})$ denotes the number of roots of p in $\mathcal{S}$. We always count roots with multiplicity.

We consider boxes (that is, squares with horizontal and vertical edges, parallel to coordinate axis) and discs $D(c, r) = \{z \text{ s.t. } |z - c| \leq r\}$ on the complex plane. For such a box (resp. disc) $\mathcal{S}$ and a positive δ we denote by $\delta\mathcal{S}$ the concentric δ-dilation. A disc Δ is an *isolator* if $\#(\Delta) > 0$; it is a *natural* isolator if in addition $\#(\Delta) = \#(3\Delta)$. A set $\mathcal{R}$ of roots of p is a *natural cluster* or just *cluster* for short if there exists a natural isolator Δ with $Z(\mathcal{R}) = Z(\Delta)$. Δ is an ε-isolator and the set $\mathcal{R}$ is an ε-cluster if ε exceeds the diameter of Δ.

The *Local Clustering Problem* (LCP) is the problem of computing natural ε-isolators for natural ε-clusters together with the sum of multiplicities of roots in the clusters in a fixed ROI:

Local Clustering Problem (LCP):
Given: a polynomial $p \in \mathbb{C}[z]$, a ROI $B_0 \subset \mathbb{C}$, $\varepsilon > 0$
Output: a set of pairs $\{(\Delta^1, m^1), \ldots, (\Delta^\ell, m^\ell)\}$ where:
- the Δ^j's are pairwise disjoint discs of radius $\leq \varepsilon$,
- $m^j = \#(\Delta^j, p) = \#(3\Delta^j, p)$ and $m^j > 0$ for $j = 1, \ldots, \ell$
- $Z(B_0, p) \subseteq \bigcup_{j=1}^{\ell} Z(\Delta^j, p) \subseteq Z(2B_0, p)$.

Root Clustering Problem (RCP) is a global version of LCP:

Root Clustering Problem:
Given: a polynomial $p \in \mathbb{C}[z]$ of degree d
Output: a set of pairs $\{(\Delta^1, m^1), \ldots, (\Delta^\ell, m^\ell)\}$ where:
- the Δ^j's are pairwise disjoint discs,
- $m^j = \#(\Delta^j, p) = \#(3\Delta^j, p)$ and $d > m^j > 0$ for $j = 1, \ldots, \ell$
- $\bigcup_{j=1}^{\ell} Z(\Delta^j, p) = Z(\mathbb{C}, p)$.

We can readily transform an algorithm for LCP into that RCP by using a bound on the norm of the roots of p, e.g., the Fujiwara bound (see [5]) for the ROI. Conversely, an algorithm RCP can initialize an algorithm for the LCP, followed by refining natural isolators to a fixed size, e.g., by means of solving the RCP itself. We can achieve quadratic convergence to the clusters by using Newton's iterations.

A nearly optimal subdivision algorithm of [1] solves the LCP by means of subdivision. It combines exclusion and counting tests based on Pellet's theorem and Newton iterations. [8] describes high-level improvements of [1] and a `C` implementation of its algorithm

*Rémi's work is supported by NSF Grants # CCF-1563942 and # CCF-1564132.
†Victor's work is supported by NSF Grants # CCF-1116736 and # CCF-1563942 and by PSC CUNY Award 698130048.

ISSAC '20, July 20–23, 2020, Kalamata, Greece

ACM ISBN 978-1-4503-7100-1/20/07…$15.00
https://doi.org/10.1145/3373207.3404063

called `Ccluster`[1]. Computational cost of application of Pellet's theorem to a disc $D(c,r)$ is dominated by the cost of shifting and scaling the variable $z \longrightarrow c + zr$ and of Dandelin-Gräffe's root-squaring iterations.

Our Contributions. The core tool for solving both LCP and RCP is a test for counting the number s_0 of roots in a disc $\Delta = D(c,r)$. If the boundary $\partial\Delta$ contains no roots, then by virtue of Cauchy's theorem

$$s_0 = \frac{1}{2\pi \mathbf{i}} \int_{\partial\Delta} \frac{p'(z)}{p(z)} dz, \text{ for } \mathbf{i} = \sqrt{-1}, \tag{1}$$

By following [7, 13, 20], we compute approximation s_0^* to s_0 by means of the evaluation of p'/p on q points of $\partial\Delta$. We give an *effective*[2] (*i.e.* implementable) description of our test, said to be P^*-test, for counting the number of roots in any disc Δ. This test involves no coefficients of p and can be applied to a *black box polynomial*, that is, a polynomial p given by a black box for its evaluation (and for implied evaluation of p' [10]). Unlike the counting tests based on Pellet's theorem, we do not require shifting and scaling the variable z and, moreover, replace Dandelin-Gräffe's costly root-squaring iterations by recursively doubling the number q of evaluation points.

By restricting our root-counting to decision whether the number of roots is 0 or not, we arrive at our exclusion test, said to be P^0-test; it decides if a disc contains no roots.

We show how to use our exclusion test in a subdivision algorithm for solving the RCP. Our algorithm can fail but always terminates. We provide some heuristic support for its correctness, and in Sec. 6 we point out to the most recent results on the formal support of the correctness of our approach, which seems to preserve the nearly optimal Boolean cost bound of the algorithms of [1] and [2]. Our goal, however, is not to compete with but to cooperate with algorithms of [1] and [2] and possibly to amend them.

We have implemented our algorithm in a procedure called `CclusterF`[3] and showed empirically that it allows significant practical improvements of root clustering compared to `Ccluster`. For sparse polynomials and polynomials defined by recursive process such as Mandelbrot's polynomials (see [3, Eq. (16)]), the resulting acceleration of the clustering algorithm of [1] is particularly strong. In experiments we carried out, `CclusterF` never failed.

Organization of the Paper. In Sec. 2 we approximate s_0 and estimate approximation error. Secs. 3 and 4 present our P^* and P^0-tests, respectively. In Sec. 5 we present our subdivision algorithm for solving the RCP using the P^0-test. In the rest of the present section, we recall the related work and the clustering algorithm of [1].

1.1 Previous Works

Univariate polynomial root-finding is a long-standing and still actual problem; it is intrinsically linked to numerical factorization of a polynomial into the product of its linear factors. The algorithms of [12] solved both problems of factorization and root-finding in record Boolean time, which is nearly optimal, that is, optimal up to a polylog factor in the input size and output precision. The algorithms are involved and have never been implemented. User's choice has been for a while the package of subroutines `MPsolve` (see [3] and [4]), based on simultaneous Newton-like (*i.e.* Ehrlich-Aberth) iterations. They converge to all roots simultaneously. As we said already, empirically they do this very fast right from the start, albeit with no formal support, and they approximate a small number of roots in a ROI not much faster than all roots. In contrast the nearly optimal cost of the algorithms of [12] and [1], already cited, is roughly proportional to the number of roots in a ROI. [1] extends the method of [2] to root clustering, *i.e.* it solves the LCP and is robust in the case of multiple roots; its implementation [8] is a little more efficient than `MPsolve` for ROIs containing a small number of roots; when all the roots are sought, `MPsolve` remains the user's choice. The algorithms of [1] and [2] follow subdivision algorithms of [17] and [11], presented there under the name of Quad-tree algorithms (inherited from [6]). [15, 16, 19] achieve a nearly optimal complexity in the real case; [9] implements the algorithm of [19]. Much more rudimentary variants of our algorithms and of their implementation appeared in [14] and [7], respectively. In Remark 8 we comment on a technical link to [20].

1.2 Solving the RCP

The root clustering algorithm in [1] combines two tests, called exclusion and counting test, with recursive subdivision of an initial box.

C^0 and C^* tests. The two tests C^0 and C^* exclude boxes with no roots of p and count the number of roots in a box, respectively. Both tests have a failure mode, *i.e.* return -1 when they cannot make decision. For a given complex disc Δ, $C^*(\Delta, p)$ (resp. $C^0(\Delta, p)$) returns an integer $k \geq 0$ (resp. 0) that indicate that there are precisely k (resp. no) roots in Δ. In the following, we frequently write $C^0(\Delta)$ for $C^0(\Delta, p)$ and $C^*(\Delta)$ for $C^*(\Delta, p)$.

In [1, 2, 8], both C^0 and C^* are based on the so called "soft Pellet test" denoted $T^*(\Delta, p)$ or $T^*(\Delta)$ which returns an integer $k \geq -1$ such that $k \geq 0$ only if p has k roots in Δ:

$$\begin{aligned} C^0(\Delta) &:= \begin{cases} 0 & \text{if } T^*(\Delta) = 0 \\ -1 & \text{otherwise} \end{cases} \\ C^*(\Delta) &:= T^*(\Delta). \end{aligned} \tag{2}$$

Boxes, Quadri-section and Connected Components. The box B centered in $c = a + \mathbf{i}b$ with width w is defined as $[a - w/2, a + w/2] + \mathbf{i}[b - w/2, b + w/2]$. $w(B)$ denotes the width of B. The *containing disc* of B is the disc $\Delta(B) := D(c, \frac{3}{4}w(B))$ The four children of B are the four boxes centered in $(a \pm \frac{w}{4}) + \mathbf{i}(b \pm \frac{w}{4})$ and having width $\frac{w}{2}$.

Recursive subdivisions of a ROI B_0 amounts to the construction of a tree rooted in B_0. Below we refer to boxes that are nodes (and possibly leafs) of this tree as the boxes of the subdivision tree of B_0.

A *component* C is a set of connected boxes. The component box B_C of a component C is a smallest square box subject to $C \subseteq B_C \subseteq B_0$, where B_0 is the initial ROI. We write $\Delta(C)$ for $\Delta(B_C)$ and $w(C)$ for $w(B_C)$. Below we consider components made up of boxes of the same width; such a component is *compact* if $w(C)$ is at most 3 times the width of its boxes. Finally, a component C is *separated* from a set S of components if $\forall C' \in S, 4\Delta(C) \cap C' = \emptyset$ and $4\Delta(C) \subseteq 2B_0$.

[1] https://github.com/rimbach/Ccluster

[2] by effective, we refer to the pathway proposed in [21] to describe algorithms in three levels: abstract, interval, effective

[3] We have done this before deterministic support for correctness of our exclusion test appeared in [13] and verified correctness by using a test from [8].

Algorithm 1 Root Clustering Algorithm

Input: A polynomial $p \in \mathbb{C}[z]$ of degree d.
Output: Set R of components solving the LCP.

1: $w \leftarrow$ upper bound for the norm of the roots of p
2: $B_0 \leftarrow$ box centered in 0 with width w
3: $R \leftarrow \emptyset$, $Q \leftarrow \{B_0\}$ // *Initialization*
4: **while** Q is not empty **do** // *Main loop*
5: $C \leftarrow Q.pop()$ //C *has the widest component box in* Q
// *Validation*
6: **if** C is compact **and** C is separated from Q **then**
7: $k \leftarrow C^*(2\Delta(C), p)$
8: **if** $d > k > 0$ **then**
9: $R.push((C, k))$
10: **break**
// *Bisection*
11: $S \leftarrow$ empty set of boxes
12: **for** each box B of C **do**
13: **for** each child B' of B **do**
14: **if** $C^0(\Delta(B'), p)$ returns -1 **then**
15: $S.push(B')$
16: $Q.push($ connected components in S $)$
17: **return** R

A Root Clustering Algorithm. We give in Algo. 1 a simple root clustering algorithm based on subdivision. A ROI containing all the roots of p is constructed using the so-called Fujiwara bound for the norm of the roots of p (see [5]). In step 16, an implicit processing of S groups the boxes in components. The paper [1] proves that Algo. 1 terminates and outputs a correct solution provided that the C^0 and C^*-tests are as in Eq. (2).

In the **while** loop of Algo. 1, components with widest component box are processed first; together with the definition of a separated component, this implies the following remarks:

Remark 1. *Let C be a component in Algo. 1 that passes the test in step 6. Then C satisfies $\#(\Delta(C)) = \#(4\Delta(C))$.*

Remark 2. *If $\#(\frac{2\sqrt{2}}{3}\Delta) = \#(\frac{4}{3}\Delta)$, then $T^*(\Delta) \geq 0$ (see [1, Lem. 3]) and if C^* is defined as in (2), then k in step 7 of Algo. 1 is non-negative.*

2 THE POWER SUMS OF THE ROOTS IN THE UNIT DISC

Let roots $\{\alpha_1, \ldots, \alpha_{d_\Delta}\}$ of p lie in Δ, roots $\{\alpha_{d_\Delta+1}, \ldots, \alpha_d\}$ lie outside Δ, and no roots lie on the boundary $\partial\Delta$.

Definition 3 (The power sums of the roots in a disc). *The h-th power sum of the roots of p in Δ is the complex number*

$$s_h = \sum_{j=1}^{d_\Delta} \alpha_j^h \tag{3}$$

Hereafter q is an integer exceeding 1 and ζ denotes a primitive q-th root of unity. The h-th power sum of the roots in the unit disc $\Delta = D(0,1)$ can be approximated by

$$s_h^* = \frac{1}{q} \sum_{g=0}^{q-1} \zeta^{g(h+1)} \frac{p'(\zeta^g)}{p(\zeta^g)} \tag{4}$$

provided that Δ has no root on its boundary. The following theorem of [13] explicitly expresses the sums s_h^* through the roots.

Theorem 4. *Let $\alpha_1, \ldots, \alpha_d$ be all d roots of $p(z)$. Then*

$$s_h^* = \sum_{j=1}^{d} \frac{\alpha_j^h}{1-\alpha_j^q} \text{ unless } \alpha_j^q = 1 \text{ for some } j. \tag{5}$$

2.1 Proof of Theorem 4

We begin with recalling some auxiliary properties.

(i) Differentiate the equation $p(z) = \mathrm{lcf}(p) \prod_{j=1}^{d}(z-\alpha_j)$ and obtain

$$\frac{p'(z)}{p(z)} = \sum_{j=1}^{d} \frac{1}{z-\alpha_j}. \tag{6}$$

(ii) For a primitive q-th root of unity ζ, it holds that

$$\zeta^g \neq 1 \text{ for } 0 < g < q, \ \sum_{g=0}^{q-1} \zeta^g = 0, \text{ and } \zeta^q = 1. \tag{7}$$

(iii) Newman's expansion: if $|y| < 1$ then $\frac{1}{1-y} = \sum_{s=0}^{\infty} y^s$.

Lemma 5. *For a complex z with $|z| \neq 1$, integers $h \geq 0$ and $q > 1$ and a primitive q-th root of unity ζ it holds that*

$$\frac{1}{q} \sum_{g=0}^{q-1} \frac{\zeta^{(h+1)g}}{\zeta^g - z} = \frac{z^h}{1-z^q}. \tag{8}$$

Proof of Lem. 5: First let $|z| < 1$ and obtain

$$\frac{\zeta^{(h+1)g}}{\zeta^g - z} = \frac{\zeta^{hg}}{1-\frac{z}{\zeta^g}} = \zeta^{hg} \sum_{s=0}^{\infty} \left(\frac{z}{\zeta^g}\right)^s = \sum_{s=0}^{\infty} \frac{z^s}{\zeta^{(s-h)g}}$$

where the equation in the middle follows from Newman's expansion for $y = \frac{z}{\zeta^g}$. We can apply it because $|y| = |z|$ while $|z| < 1$.

Sum the fractions $\frac{z^s}{\zeta^{(s-h)g}}$
in g and deduce from (7) that

$$\frac{1}{q} \sum_{g=0}^{q-1} \frac{z^s}{\zeta^{(s-h)g}} = \begin{cases} z^s \text{ when } s = h + ql \text{ for an integer } l, \\ 0 \text{ otherwise.} \end{cases}$$

Therefore

$$\frac{1}{q} \sum_{g=0}^{q-1} \frac{\zeta^{(h+1)g}}{\zeta^g - z} = z^h \sum_{l=0}^{\infty} z^{ql}.$$

Apply Newman's expansion for $y = z^q$ and deduce (8) provided that $|z| < 1$. Now let $|z| > 1$. Then

$$\frac{\zeta^{(h+1)g}}{\zeta^g - z} = -\frac{\zeta^{(h+1)g}}{z} \frac{1}{1-\frac{\zeta^g}{z}} = -\frac{\zeta^{(h+1)g}}{z} \sum_{s=0}^{\infty} \left(\frac{\zeta^g}{z}\right)^s = -\sum_{s=0}^{\infty} \frac{\zeta^{(s+h+1)g}}{z^{s+1}}.$$

Sum these expressions in g, write $s := ql - h - 1$, and apply (7):

$$\frac{1}{q} \sum_{g=0}^{q-1} \frac{\zeta^{(h+1)g}}{\zeta^g - z} = -\sum_{l=1}^{\infty} \frac{1}{z^{ql-h}} = -z^{h-q} \sum_{l=0}^{\infty} \frac{1}{z^{ql}}.$$

Apply Newman's expansion for $y = 1/z^q$ and obtain that

$$\frac{1}{q} \sum_{g=0}^{q-1} \frac{\zeta^{(h+1)g}}{\zeta^g - z} = -\frac{z^{h-q}}{1-\frac{1}{z^q}} = \frac{z^h}{1-z^q}.$$

Hence (8) holds in the case where $|z| > 1$ as well. □

2.2 Error bounds for the approximation of the power sums

Definition 6 (Isolation ratio). *A complex disc Δ has an* isolation ratio $\rho \geq 1$ *for a polynomial p or equivalently is* ρ-isolated *if* $Z(\frac{1}{\rho}\Delta) = Z(\rho\Delta)$.

Corollary 7. *For $\rho > 1$, $\theta := 1/\rho$, and an integer h such that $0 \leq h < q$ let the disc $\Delta = D(0,1)$ be ρ-isolated and contain d_Δ roots of p. Then*

$$|s_h^* - s_h| \leq \frac{d_\Delta \theta^{q+h} + (d - d_\Delta)\theta^{q-h}}{1 - \theta^q}. \tag{9}$$

Remark 8. *The corollary does not imply Theorem 4. In [20] Schönhage proved the corollary and applied it to deflation of p, ignoring the case of $h = 0$ and root-counting problem and bypassing the theorem.*

Proof of Corollary 7: Let $\{\alpha_1, \ldots, \alpha_{d_\Delta}\}$ be the roots of p in Δ and $\{\alpha_{d_\Delta+1}, \ldots, \alpha_d\}$ be the roots of p outside Δ. First combine (5) with (3) to obtain

$$s_h^* - s_h = \sum_{j=1}^{d_\Delta} \frac{\alpha_j^{q+h}}{1 - \alpha_j^q} + \sum_{j=d_\Delta+1}^{d} \frac{\alpha_j^h}{1 - \alpha_j^q} \tag{10}$$

Recall that $\theta = 1/\rho < 1$. For $1 \leq j \leq d_\Delta$, one has

$$\left| \frac{\alpha_j^{q+h}}{1 - \alpha_j^q} \right| \leq \frac{\theta^{q+h}}{1 - \theta^q} \tag{11}$$

For $d_\Delta + 1 \leq j \leq d$, it holds that $1/\alpha_j \leq \theta$ and

$$\left| \frac{\alpha_j^h}{1 - \alpha_j^q} \right| = \left| \frac{\alpha_j^{h-q}}{1/\alpha_j^q - 1} \right| \leq \frac{\theta^{q-h}}{1 - \theta^q} \tag{12}$$

Combining (10), (11) and (12) implies inequality (9). □

3 COUNTING THE NUMBER OF ROOTS IN A WELL-ISOLATED DISC

Given $\rho > 1$, a black box polynomial p, and ρ-isolated disc $D(0,1)$, Corollary 7 suggests the following recipe for counting the roots of p in $D(0,1)$: first choose q such that $|s_0^* - s_0|$ is less than $1/4$ and then compute s_0^* of (4), at the overall cost of the evaluation of p and p' at $q = O(\log(d))$ points and $O(q)$ additional arithmetic operations. Clearly a unique integer in the disc $D(s_0^*, 1/4)$ is the number of roots in $D(0,1)$. In this section we extend this recipe to P^*-test for counting the roots of a black box polynomial p in any ρ-isolated disc $\Delta = D(c,r)$ for $\rho > 1$.

When Δ has isolation ratio 2 and p has degree 500, our test amounts to evaluating p and p' on $q = 11$ points.

If p and p' can be evaluated at a low computational cost, e.g. if p is sparse or defined by a recurrence as the Mandelbrot polynomial (see [3, Eq. (16)]), our P^*-test can be dramatically simplified.

3.1 Approximation of the 0-th Power Sum of the Roots in any Disc

Let $\Delta = D(c,r)$ and s_0 be the 0-th power sum of the roots of p in Δ, as defined in Def. 3. For a positive integer q, define

$$s_0^* = \frac{r}{q} \sum_{g=0}^{q-1} \zeta^g \frac{p'(c + r\zeta^g)}{p(c + r\zeta^g)} \tag{13}$$

where ζ is a primitive q-th root of unity.

Corollary 9. *Let Δ have isolation ratio ρ, and $\theta = 1/\rho$. Then*

$$|s_0^* - s_0| \leq \frac{d\theta^q}{1 - \theta^q} \tag{14}$$

$$\textit{Fix } e > 0. \textit{ If } q = \lceil \log_\theta(\frac{e}{d+e}) \rceil \textit{ then } |s_0^* - s_0| \leq e \tag{15}$$

Proof of Corollary 9: Let $p_\Delta(z)$ be the polynomial $p(c + rz)$. Then $p'_\Delta(z) = rp'(c + rz)$ and Eq. (13) rewrites $s_0^* = \frac{1}{q} \sum_{g=0}^{q-1} \zeta^g \frac{p'_\Delta(\zeta^g)}{p_\Delta(\zeta^g)}$. In addition, the unit disc $D(0,1)$ has isolation ratio ρ for p_Δ and contains s_0 roots of p_Δ. Then apply Thm. 7 to $p_\Delta(z)$ to obtain (14). (15) is a direct consequence of (14). □

Remark that in (15), the required number q of evaluation points increases as the logarithm of ρ: if Δ has isolation ratio $\sqrt{\rho}$ (resp. ρ^2) instead of ρ, then $\frac{1}{2}q$ (resp. $2q$) evaluation points are required. Thus doubling the number of evaluation points has the same effect as root squaring operations.

Our test uses the following bound.

Lemma 10. *Suppose that $\Delta = D(c,r)$ has isolation ratio $\rho > 1$, $z \in \mathbb{C}$, $|z| = 1$, and g is an integer. Then*

$$|p(c + rz^g)| \geq \mathrm{lcf}(p) \frac{r^d(\rho - 1)^d}{\rho^d} \tag{16}$$

Proof of Lem. 10. Suppose that p has d_Δ non-necessarily distinct roots $\alpha_1, \ldots, \alpha_{d_\Delta}$ in Δ and $d - d_\Delta$ roots $\alpha_{d_\Delta+1}, \ldots, \alpha_d$ outside Δ. Since Δ has isolation ratio ρ, it follows that

$$|c + rz^g - \alpha_i| \geq r - \frac{r}{\rho} = \frac{r(\rho - 1)}{\rho} \text{ when } i \leq d_\Delta, \text{ and} \tag{17}$$

$$\geq \rho r - r = r(\rho - 1) \text{ when } i \geq d_\Delta + 1 \tag{18}$$

Write

$$p(c + rz^g) = \mathrm{lcf}(p) \prod_{i=1}^{d_\Delta} (c + rz^g - \alpha_i) \prod_{i=d_\Delta+1}^{d} (c + rz^g - \alpha_i)$$

and deduce from inequalities (17) and (18) that

$$|p(c + rz^g)| \geq \mathrm{lcf}(p) (\frac{r(\rho - 1)}{\rho})^{d_\Delta} (r(\rho - 1))^{d - d_\Delta} = \mathrm{lcf}(p) \frac{r^d(\rho - 1)^d}{\rho^{d_\Delta}}.$$

Bound (16) follows since $\rho > 1$. □

3.2 Black Box for Evaluating a Polynomial on an Oracle Number

Our P^*-test deals with *oracle numbers*, the black boxes for arbitrary precision approximation of complex numbers. Such oracle numbers can be implemented through arbitrary precision interval arithmetic or ball arithmetic. Let $\square\mathbb{C}$ be the set of complex intervals. If $\square a \in \square\mathbb{C}$, then $w(\square a)$ is the maximum width of real and imaginary parts of $\square a$.

For a number $a \in \mathbb{C}$, we call *oracle* for a a function $O_a : \mathbb{N} \to \square\mathbb{C}$ such that $a \in O_a(L)$ and $w(O_a(L)) \leq 2^{-L}$ for any L. Let $\mathcal{O}_\mathbb{C}$ be the set of oracle numbers which can be computed with a Turing machine.

For a polynomial $p \in \mathbb{C}[z]$, we call *evaluation oracle* for p a function $\mathcal{I}_p : (\mathcal{O}_\mathbb{C}, \mathbb{N}) \to \square\mathbb{C}$, such that if O_a is an oracle for a and $L \in \mathbb{N}$, then $p(a) \in \mathcal{I}_p(O_a, L)$ and $w(\mathcal{I}_p(O_a, L)) \leq 2^{-L}$.

Algorithm 2 $P^*(\mathcal{I}_p, \mathcal{I}_{p'}, \Delta, \rho)$

Input: $\mathcal{I}_p$, $\mathcal{I}_{p'}$ evaluation oracles for p and p', $\Delta = D(c, r)$, $\rho > 1$. p is monic and has degree d.
Output: an integer in $\{-1, 0, \ldots, d\}$

1: $L \leftarrow 53$, $w \leftarrow 1$, $e \leftarrow 1/4$
2: $\theta \leftarrow 1/\rho$, $q \leftarrow \lceil \log_\theta(\frac{e}{d+e}) \rceil$
3: $\ell \leftarrow r^d(\rho-1)^d/\rho^d$
4: **while** $w \geq 1/2$ **do**
5: **for** $g = 0, \ldots, q-1$ **do**
6: Compute intervals $\mathcal{I}_p(O_{c+r\zeta^g}, L)$ and $\mathcal{I}_{p'}(O_{c+r\zeta^g}, L)$
7: **if** $|\mathcal{I}_p(O_{c+r\zeta^g}, L)| < \ell$ **then**
8: **return** -1
9: Compute interval $\square s_0^*$ as $\frac{r}{q} \sum_{g=0}^{q-1} O_{\zeta^g}(L) \frac{\mathcal{I}_{p'}(O_{c+r\zeta^g}, L)}{\mathcal{I}_p(O_{c+r\zeta^g}, L)}$
10: $w \leftarrow w(\square s_0^*)$
11: $L \leftarrow 2L$
12: $\square s_0 \leftarrow \square s_0^* + [-e, e] + \mathbf{i}[-e, e]$
13: **if** $\square s_0$ contains a unique integer k **then**
14: **return** k
15: **return** -1

Consider evaluation oracles $\mathcal{I}_p$ and $\mathcal{I}_{p'}$ for p and its derivative p'. If p is given by $d+1$ oracles for its coefficients, one can easily construct $\mathcal{I}_p$ and $\mathcal{I}_{p'}$ by using, for instance, Horner's rule. However for some polynomials defined by a procedure, one can construct fast evaluation oracles $\mathcal{I}_p$ and $\mathcal{I}_{p'}$ from procedural definition.

3.3 The P^*-test

In Algo. 2 we describe our counter of the roots of a monic polynomial p in a disc $\Delta = D(c, r)$. Its input is made up of evaluation oracles for p and p', Δ, and a fixed isolation ratio $\rho > 1$ for Δ. It may fail and return -1 only if Δ is not ρ-isolated for $\rho > 1$. In the latter case its correctness cannot be guaranteed. The termination of Algo. 2 amounts to the termination of the **while** loop in step 4.

First suppose that $0 \leq g < q$ for an integer g, a disc Δ is ρ-isolated for $\rho > 1$, and $|p(c + r\zeta^g)| \geq \ell > 0$ (cf. (16)). Thus for $2^{-L} < \ell$, none of the $\mathcal{I}_p(O_{c+r\zeta^g}, L)$ can contain 0, and the width of the interval $\square s_0^*$ computed in step 9 strictly decreases with L (see, e.g., [18, Sec. 5] and in particular Eq. (5.10) and (5.11) that directly extend to $\mathbb{C}$).

Now suppose that the disc Δ is not ρ-isolated for a fixed $\rho > 1$. If one of the evaluation points $c + r\zeta^g$ is a root of p or if $p(c + r\zeta^g) < l$ then condition in step 7 is satisfied for $2^{-L} < \ell$, and the test returns -1. Otherwise $p(c + r\zeta^g) \geq \ell$ for all $g = 0, \ldots, q-1$, and then the interval $\square s_0^*$ computed in step 9 has width that strictly decreases with L.

This proves the termination of Algo. 2 when p is monic. One can easily write a terminating algorithm for non-monic polynomials assuming a lower bound on the leading coefficient of p.

The correctness of Algo. 2 is stated in the following proposition:

Proposition 11. *Let Δ be ρ-isolated. Then p has k roots in Δ if and only if $P^*(\mathcal{I}_p, \mathcal{I}_{p'}, \Delta, \rho)$ returns k.*

Proof of Prop. 11. Since Δ is ρ-isolated Lemma 10 implies that the condition in step 7 is never reached. By virtue of Corollary 9, the interval $\square s_0$ computed in step 12 of Algo. 2 has width less than 1 and contains a unique integer s_0, the number of roots of p in Δ counted with multiplicity. □

4 AN ALMOST SURE EXCLUSION TEST

The P^*-test of sec. 3 counts the roots in Δ using no Taylor's shift but just evaluates p at $O(\log(d))$ points on the contour of Δ, which is a major benefit versus [1]. However, we cannot ensure its success unless we know that Δ is ρ-isolated for ρ noticeably exceeding 1, and this disqualifies its use as an exclusion test within a subdivision framework of [1]. Our alternative version of the P^*-test in [7] works in the case where ρ is not known, and we used it as an exclusion test while confirming its output with the T^*-test.

Here we define an exclusion test based on the computation of approximations of the first k power sums (for a small $k > 0$). For a disk with a fixed isolation ratio one can compute an interval $\square s_0$ containing a unique integer s as in Algo. 2. However, if the isolation ratio is smaller, then s may differ from s_0. In the test described below, we verify further necessary conditions that $s = s_0$. Namely, we compute the k first power sums of the roots of p_Δ in $D(0, 1)$ for a small k: for h less that k, p contains no root in Δ only if the h-th powers of the roots of p in Δ sum to 0, which in turn happens only if the h-th powers of the roots of p_Δ in $D(0, 1)$ sum to 0.

Subsec. 4.1 extends Corollary 9 to the approximation of the h-th power sum of the roots of p_Δ in $D(0, 1)$. In Subsec. 4.2, we define our exclusion test and give a sufficient condition in terms of the distances of the roots to a box B for our test to exclude B. In Subsec. 4.3 we provide experimental results confirming our heuristic.

4.1 Approximation of h-th power sum

For a positive integer q and $0 \leq h < q$, define

$$s_h^* = \frac{r}{q} \sum_{g=0}^{q-1} \zeta^{g(h+1)} \frac{p'(c + r\zeta^g)}{p(c + r\zeta^g)}. \tag{19}$$

By replacing h by 0, (19) directly extends (13), and likewise bound (20) below directly extends (9).

Corollary 12 (of Theorem 7). *Let $\Delta = D(c, r)$ have isolation ratio ρ, and $\theta = 1/\rho$. Let p have degree d and d_Δ roots in Δ. Then*

$$|s_h^* - s_h| \leq \frac{d_\Delta \theta^{q+h} + (d - d_\Delta)\theta^{q-h}}{1 - \theta^q} \tag{20}$$

$$|s_h^* - s_h| \leq \frac{d\theta^{q-h}}{1 - \theta^q} \tag{21}$$

$$\text{Fix } e > 0. \text{ If } q = \lceil \log_\theta(\frac{e}{d+e}) \rceil + h, \text{ then } |s_h^* - s_h| \leq e \tag{22}$$

Proof of (21) and (22) in Corollary 12: We deduce (21) from (20) by noticing that $\theta < 1$ and $d_\Delta \theta^{q+h} \leq d_\Delta \theta^{q-h}$. (22) is a direct consequence of (21). □

4.2 The P^0-test

We describe our P^0-test in Algo. 3 in the case where p is monic. At the first stage, we rely on eq. (19) and for $0 \leq h \leq k$, compute the interval $\square s_h^*$, containing s_h^* and having width less than 1/2. At the second stage, for $0 \leq h \leq k$, we obtain the interval $\square s_h$ from $\square s_h^*$ by adding the errors bounded in (21). $\square s_h$ contains s_h for all h if Δ

Algorithm 3 $P^0(\mathcal{I}_p, \mathcal{I}_{p'}, \Delta, \rho, k)$

Input: $\mathcal{I}_p, \mathcal{I}_{p'}$ evaluation oracles for p and p', $\Delta = D(c, r)$, $\rho > 1$. p is monic and has degree d. $k \geq 0$ is an integer.
Output: an integer in $\{-1, 0\}$

1: $L \leftarrow 53$, $w \leftarrow 1$, $\theta \leftarrow 1/\rho$
2: $e \leftarrow 1/4$, $q \leftarrow \lceil \log_\theta(\frac{e}{d+e}) \rceil + k$
3: $\ell \leftarrow r^d(\rho-1)^d/\rho^d$
4: **while** $w \geq 1/2$ **do**
5: **for** $g = 0, \ldots, q-1$ **do**
6: Compute intervals $\mathcal{I}_p(O_{c+r\zeta^g}, L)$ and $\mathcal{I}_{p'}(O_{c+r\zeta^g}, L)$
7: **if** $|\mathcal{I}_p(O_{c+r\zeta^g}, L)| < \ell$ **then**
8: **return** -1
9: **for** $h = 0, \ldots, k$ **do**
10: Compute interval $\square s_h^*$ as $\frac{r}{q} \sum_{g=0}^{q-1} O_{\zeta^{g(h+1)}}(L) \frac{\mathcal{I}_{p'}(O_{c+r\zeta^g}, L)}{\mathcal{I}_p(O_{c+r\zeta^g}, L)}$
11: $w \leftarrow \max_{h=0,\ldots,k} w(\square s_k^*)$
12: $L \leftarrow 2 * L$
13: **for** $h = 0, \ldots, k$ **do**
14: $e_h \leftarrow (d\theta^{q-h})/(1-\theta^q)$
15: $\square s_h \leftarrow \square s_h^* + [-e_h, e_h] + \mathbf{i}[-e_h, e_h]$
16: **if** $0 \notin \square s_h$ **then**
17: **return** -1
18: **return** 0

has isolation ratio $\rho > 1$. We have chosen q such that $\square s_h$ contains at most one integer for all h and we arrive at

Proposition 13. *If Δ has isolation ratio ρ then for $k \geq 0$, $P^0(\mathcal{I}_p, \mathcal{I}_{p'}, \Delta, \rho, k)$ returns 0 if and only if Δ contains no root of p.*

One proves the termination of Algo. 3 with the same arguments as for Algo. 2, and the same remark about the assumption that p is monic holds. For a box B that contains no root of p, we give a sufficient condition for our test to return 0:

Proposition 14. *Let a disc $\Delta(B)$ contain a box B. If $2B$ contains no root of p then $P^0(\mathcal{I}_p, \mathcal{I}_{p'}, \Delta(B), 4/3, k)$ returns 0 for any $k \geq 0$.*

Proof of Prop. 14. Let B have center c and width w. Recall that $\Delta(B) = D(c, \frac{3}{4}w)$, thus $\frac{4}{3}\Delta(B) = D(c, w)$. Now, $D(c, w) \subseteq 2B$ and if $2B$ contains no roots of p, then so does $D(c, w)$ as well. Thus $\Delta(B)$ has isolation ratio $\geq \frac{4}{3}$. In this case, by virtue of Lemma 10, $p(c + r\zeta^g) \geq \ell$ for all $g = 0, \ldots, q-1$. As a consequence, after the **while** loop of Algo. 3, each interval $\square s_h^*$ has width strictly less than $1/2$, and contains s_h^*. Now, one has the following bounds:

$$|s_h^* - s_h| \leq \frac{d\theta^{q-h}}{1-\theta q} \leq \frac{d\theta^{q-k}}{1-\theta q} \leq 1/4 \qquad (23)$$

The first inequality comes from (21). The inequality in the middle holds since $\theta < 1$ and $h \leq k < q$. The right-hand side inequality is a consequence of (22) and the choice of e. Thus $\square s_h$ computed in step 15 of Algo. 3 contains s_h, which is 0, and contains a unique integer since it has width strictly less than 1. □

4.3 On the success of the P^0-test

Here we give experimental evidences that for a given disc Δ, if $P^0(\mathcal{I}_p, \mathcal{I}_{p'}, \Delta, \rho, k)$ with $\rho = 4/3$ and $k = 2$ returns 0, then Δ is very likely to contain no roots of p.

		T^*-tests	P^0-tests, $k=0$		P^0-tests, $k=1$		P^0-tests, $k=2$		
d	n	t_0/t (%)	#TN	#FP	#TN	#FP	#TN	#FP	t_0^2/t_0
100 monic random dense polynomials per degree									
64	116302	87.2	3741	4	5611	0	7260	0	1.14
128	227842	90.5	6417	21	9935	0	12972	0	.599
191	340348	92.0	8850	26	13770	1	18004	0	.455
Bernoulli polynomials									
64	1566	87.5	32	0	42	0	60	0	.836
128	2954	88.4	49	0	65	0	87	0	.578
191	4026	88.7	100	0	163	0	212	0	.462
100 monic random sparse (10 monomials) polynomials per degree									
64	115850	86.2	3628	10	5430	0	6986	0	.981
128	226266	91.3	6471	11	9660	0	12556	0	.403
191	331966	92.1	8690	11	13425	2	17452	0	.280
Mignotte polynomials									
64	1196	85.7	30	0	48	0	63	0	1.00
128	2296	92.9	63	0	93	0	129	0	.298
191	3218	92.4	70	2	109	0	154	0	.264

Table 1: True negatives and false positives when using the P^*-test with $\rho = 4/3$ and $k = 0, 1, 2$.

We run Algo. 1 implemented in `Ccluster` for dense and sparse polynomials, random and taken from literature; each time `Ccluster` applies exclusion test based on T^*-test for a box B, we also apply $P^0(\mathcal{I}_p, \mathcal{I}_{p'}, \Delta(B), \frac{4}{3}, k)$ with values 0, 1, 2 for k.

The *false positives* are the cases where for a disc $\Delta(B)$, the T^*-test returns a positive number of roots or cannot decide whether $P^0(\mathcal{I}_p, \mathcal{I}_{p'}, \Delta, 2, k)$ returns 0. For the polynomials we tested, when $k = 2$, there was no such false positives.

The *true negatives, i.e.* cases where a disc contains no root according to the T^*-test but $P^0(\mathcal{I}_p, \mathcal{I}_{p'}, \Delta, 2, k)$ returns -1, shows how less efficacious than the T^*-test is our test.

4.3.1 Testing suite. For each degree $d \in \{64, 128, 191\}$, we generated 100 random monic dense polynomials whose coefficients are rational numbers $\frac{c}{256}$ where c is an integer chosen uniformly in $[-256, 256]$. We also generated for each degree above 100 random monic sparse polynomials as follows: choose 8 distinct random integers $d_1, \ldots, d_8$ in the range $[1, d-1]$, and let the coefficients of monomials of degrees $0, d_1, \ldots, d_8$ be rational numbers $\frac{c}{256}$ for a random integer chosen uniformly in $[-256, 256]$.

We also consider Bernoulli and Mignotte polynomials. The Bernoulli polynomial of degree d is $B_d(z) = \sum_{k=0}^{d} \binom{d}{k} b_{d-k} z^k$ where the b_i's are the Bernoulli numbers. It has about $d/2$ non-zero coefficients. The Mignotte polynomial of degree d and parameter $a = 8$ is $M_d(z) = z^d - 2(2^a z - 1)^2$.

4.3.2 Results. For a polynomial in our testing suite, let n be the number of exclusion tests performed by `Ccluster`, t be the running time of `Ccluster` and t_0 be the time spent in exclusion T^*-test. For each exclusion test, we also applied three times our P^0-test with isolation ratio $\rho = 4/3$ and $k = 0, 1, 2$. We denote by #TN the number of true negatives, #FP the number of false positives and t_0^2 the total time spent in the P^0-test with $k = 2$. We report in Table 1 the values n, t_0/t, #TN, #FP and t_0^2/t for each degree and each family of polynomials. For random dense and sparse polynomials, these values represented overall count over the 100 polynomials.

As expected, the number of false positives decreased when k increased, and we had no such false positives when we used $k = 2$. As a counterpart, the number of true negatives increased with k.

5 A FAST AND ALMOST SURE ROOT CLUSTERING ALGORITHM

In this section we present a fast root clustering algorithm based on Algo. 1 and on exclusion and counting tests defined as:

$$\begin{array}{ll} C^0(\Delta) := & P^0(\mathcal{I}_p, \mathcal{I}_{p'}, \Delta, 4/3, 2) \\ C^*(\Delta) := & T^*(\Delta) \end{array} \tag{24}$$

Notice that the exclusion test is performed while assuming an isolation ratio 4/3 for Δ, condition that cannot be ensured when Δ is the containing disc of the boxes of a subdivision tree constructed by Algo. 1. As a consequence, exclusion tests defined in (24) may return wrong results. Although very unlikely, as we show in Subsec. 4.3, this would compromise correctness of the process (termination is ensured by Prop. 14).

In Subsec. 5.1, we describe how we modified Algo. 1 to obtain a root clustering algorithm using C^0 and C^*-tests of (24) that always terminates and has a failure mode. When it succeeds, its result is correct. This procedure has been implemented in `C` within `Ccluster`, and we call it `CclusterF` below.

In Subsec. 5.2 we show experimental results on using `Ccluster` and `CclusterF` for clustering the roots of a bunch of polynomials. `CclusterF` never failed in the experiments we carried out. Moreover by comparing running times of both procedures, we show that using C^0-tests of (24) can lead to important improvement, which grow with degree and sparsity of considered polynomials.

5.1 Description of our algorithm

We give an informal description of how we modified Algo. 1 to deal with uncertainty of the result of the exclusion test P^0.

First, in addition to a list of clusters, our algorithm returns a flag in {`fail`, `success`} indicating whether its result is reliable.

Second, we replace steps 6 to 10 in Algo. 1 with steps 6 to 12 below:

6: **if** C is compact **and** C is separated from Q **then**
7: $k \leftarrow C^*(2\Delta(C), p)$
8: **if** $d > k > 0$ **then**
9: $R.push((C, k))$
10: **break**
11: **if** $k == -1$ **then**
12: **return** `fail`, R

Third, we replace the **return** statement in step 17 in Algo. 1 with the following simple routine:

17: sum the number of roots in the components in R
18: **if** it is equal to d **then**
19: **return** `success`, R
20: **else**
21: **return** `fail`, R

Notice that step 14 of Algo. 1 also involves the C^0-test which has to be understood here as defined in (24). Recall that for a box B, $C^0(\Delta(B))$ returns -1 when $2B$ contains a root (see Prop. 14); however when it returns 0, B may contain a root.

To see that our algorithm terminates, consider Prop. 14: it implies that after a finite number of subdivision steps, boxes in the subdivision tree form separated and compact connected components, at most one per root. Then for each of these connected components C our algorithm enters step 6 above and terminates.

When our algorithm returns the flag `success`, its output is correct, *i.e.* the components in R solve the root clustering problem. This is a direct consequence of the fact that $T^*(\Delta, p)$ returns $k \geq 0$ only if Δ contains k roots of p.

Our algorithm returns the flag `fail` only if an exclusion test returns a wrong result, *i.e.* excludes a box of the subdivision tree that contains a root. Assume the opposite: no exclusion test returns a wrong result. Then Rem. 1 holds; in particular $2\Delta(C)$ has isolation ratio 2 and from Rem. 2, $C^*(2\Delta(C), p)$ in step 7 above returns k positive. Moreover, each root lies in a box in a component in R before the step 17 above, and our algorithm returns `success`.

5.2 Experimental results

5.2.1 Test polynomials. In addition to the test polynomials of Subsec. 4.3, we consider the following ones.

(i) $T_d(z)$, the Chebyshev polynomial (of the first kind) of degree d: $T_0(z) = 1$, $T_1(z) = z$ and $T_{d+1}(z) = 2zT_d(z) - T_{d-1}(z)$, $d = 2, 3, \ldots$

(ii) $L_d(z)$, the Legendre polynomial of degree d: $L_0(z) = 1$, $L_1(z) = z$ and $L_{d+1}(z) = \frac{2d+1}{d+1} zL_d(z) - \frac{d}{d+1} L_{d-1}(z)$, $d = 2, 3, \ldots$

(iii) For an integer $n > 0$, we define polynomials with $(2n+1) \times (2n+1)$ roots on the nodes of a regular grid centered in 0 as

$$P_{(2n+1)\times(2n+1)}(z) = \prod_{-n \leq a, b \leq n} (z - a + \mathbf{i}b)$$

(iv) Letting $M_1(z) = z$ and $M_k(z) = zM_{k-1}(z)^2 + 1$, we define the Mandelbrot's polynomial $M_k(z)$ of degree $2^k - 1$.

Bernoulli, Chebyshev and Legendre polynomials of degree d have about $d/2$ nonzero coefficients. Polynomials with roots on a grid of degree d have about $d/4$ nonzero coefficients. Mignotte polynomials have 4 nonzero coefficients. Mandelbrot polynomials have no zero coefficients, but can be evaluated very fast by a straight line program.

5.2.2 Results. We computed clusters of roots of each polynomial of our testing set by using both `Ccluster` and `CclusterF`. In Table 2 we report for both solvers the size of the subdivision tree (columns TS) and the sequential running time in seconds on Intel(R) Core(TM) i7-8700 CPU @ 3.20GHz machine with Linux (columns t and t'). In Table 2 we also report the number of failures of `CclusterF` (column #Fails) and the ratio t'/t in percents. Column t'_1/t' shows percents of time spent on evaluating oracle polynomials in the P^0-test. Column t'_2/t' shows percents of time spent on applying T^*-tests in `CclusterF`. As in Subsec. 4.3, the Table 2 displays the average data for random dense and sparse polynomials over the 100 polynomials of the family.

Remarks: (i) There was no occurrence of a failure of `CclusterF` for all the polynomials we tested.

(ii) The running time of `CclusterF` decreased as the degree and the sparsity of the polynomial increased. For random sparse polynomials and Mignotte polynomials, of degree 191, this was a 3-fold speed-up. The speed-up was more dramatic for polynomials evaluated very fast such as Mandelbrot polynomials. Except for the latter cases, `CclusterF` spent most of its computational time on evaluating oracle polynomials and checking correctness of the results.

	Ccluster		CclusterF					
d	TS	t	#Fails	TS	t'	t'/t (%)	t'_1/t' (%)	t'_2/t' (%)
100 monic random dense polynomials per degree								
64	127100	31.5	0	155992	41.2	130	76.8	1.72
128	250928	222	0	300696	149	67.3	83.2	4.52
191	361340	665	0	447628	340	51.1	85.1	5.96
Bernoulli polynomials								
64	1884	0.46	0	2148	0.49	106	73.4	2.04
128	3596	3.24	0	3932	1.86	57.4	85.4	2.15
191	4684	9.17	0	5476	4.84	52.7	84.2	5.99
Chebyshev polynomials								
64	2532	0.74	0	2980	0.79	106	82.2	1.26
128	4708	5.62	0	5188	3.33	59.2	84.9	.900
191	7268	17.0	0	8108	8.86	51.9	86.9	1.01
Legendre polynomials								
64	2676	0.75	0	2940	0.81	108	77.7	0.0
128	4836	5.76	0	5244	3.73	64.7	86.8	1.60
191	6996	16.4	0	7732	9.61	58.3	88.7	1.45
Polynomials with roots on a regular grid								
225	3412	8.74	0	3580	2.62	29.9	76.3	15.2
289	4548	17.0	0	5304	5.40	31.6	74.8	15.5
361	6276	30.9	0	7588	8.52	27.5	76.5	17.8
100 monic random sparse (10 monomials) polynomials per degree								
64	127220	27.9	0	159972	31.7	113	70.8	1.57
128	251196	216	0	303260	100	46.3	75.5	5.65
191	374872	638	0	457084	209	32.7	76.4	8.80
Mignotte polynomials								
64	1572	0.30	0	1856	0.31	103	74.1	0.0
128	2572	2.24	0	3564	0.95	42.4	74.7	4.21
191	3640	5.99	0	4228	1.79	29.8	72.6	10.0
Mandelbrot polynomials								
127	2852	3.46	0	3424	0.56	16.1	42.8	10.7
255	4968	18.4	0	5952	1.79	9.70	33.5	41.3
511	9632	118	0	11556	7.61	6.42	19.5	66.3

Table 2: Runs of `Ccluster` and `CclusterF` on polynomials of our testing suite.

In all the examples we tested, the depth of the subdivision tree constructed by `CclusterF` was at most one plus the depth of the tree constructed by `Ccluster`. Columns TS in Table 2 suggest that `CclusterF` tends to construct a slightly wider subdivision tree than `Ccluster`, which shows that the P^0-test is slightly less efficacious than the T^*-test for box exclusion.

6 CONCLUSION

We presented our exclusion test with doubling the number q of evaluation points as heuristic, but actually it has already probabilistic and even deterministic support; moreover even our root-counting has probabilistic support. Namely, by virtue of [13, Thm. 29 and Remark 30], based on our Theorem 4, if s_0^* is within a specified distance from an integer k, then $k = s_0$ with a high probability (whp) under the random root model and under no assumption about isolation of the disc. Furthermore by virtue of [13, Corollary 4.7] a disc contains no roots whp under a random coefficient model and again under no assumption about isolation of the disc as long as $2\tau^2 d^2 < 1$ for $\tau^2 = \sum_{h=0}^{q-1} |s_h^* - s_h|^2$ and $q \geq 2$. For $q > d$ under this bound the disc definitely contains no roots by virtue of [13, Corollary 4.6]. Notice that we can compute s_h^* for $h = 0, 1, \ldots, q-1$ at the cost of computing just s_0^* and in addition performing discrete Fourier transform at q points.

Our initial but extensive experiments showed significant acceleration of the known subdivision root finders, which is particularly strong for sparse inputs. Moreover they suggest that the latter result of [13] is overly pessimistic because exclusion test was always correct in these experiments already for q much smaller than d.

REFERENCES

[1] Ruben Becker, Michael Sagraloff, Vikram Sharma, Juan Xu, and Chee Yap. 2016. Complexity Analysis of Root Clustering for a Complex Polynomial. In *Proceedings of the ACM on International Symposium on Symbolic and Algebraic Computation (ISSAC '16)*. ACM, New York, NY, USA, 71–78. https://doi.org/10.1145/2930889.2930939

[2] Ruben Becker, Michael Sagraloff, Vikram Sharma, and Chee Yap. 2018. A Near-Optimal Subdivision Algorithm for Complex Root Isolation based on Pellet Test and Newton Iteration. *Journal of Symbolic Computation* 86 (May-June 2018), 51–96.

[3] Dario A Bini and Giuseppe Fiorentino. 2000. Design, analysis, and implementation of a multiprecision polynomial rootfinder. *Num. Alg.* 23, 2 (2000), 127–173.

[4] Dario A Bini and Leonardo Robol. 2014. Solving secular and polynomial equations: A multiprecision algorithm. *J. Comput. Appl. Math.* 272 (2014), 276–292.

[5] Matsusaburô Fujiwara. 1916. Über die obere Schranke des absoluten Betrages der Wurzeln einer algebraischen Gleichung. *Tohoku Mathematical Journal, First Series* 10 (1916), 167–171.

[6] Peter Henrici and Irene Gargantini. 1969. Uniformly convergent algorithms for the simultaneous approximation of all zeros of a polynomial. In *Constructive Aspects of the Fundamental Theorem of Algebra*. Wiley-Interscience New York, 77–113.

[7] Rémi Imbach and Victor Y Pan. 2019. New practical advances in polynomial root clustering. *arXiv preprint arXiv:1911.06706* (2019).

[8] Rémi Imbach, Victor Y. Pan, and Chee Yap. 2018. Implementation of a Near-Optimal Complex Root Clustering Algorithm. In *Mathematical Software – ICMS 2018*. 235–244.

[9] Alexander Kobel, Fabrice Rouillier, and Michael Sagraloff. 2016. Computing Real Roots of Real Polynomials ... And Now For Real!. In *Proceedings of the ACM on International Symposium on Symbolic and Algebraic Computation (ISSAC '16)*. ACM, New York, NY, USA, 303–310. https://doi.org/10.1145/2930889.2930937

[10] Seppo Linnainmaa. 1976. Taylor expansion of the accumulated rounding error. *BIT Numerical Mathematics* 16, 2 (1976), 146–160.

[11] Victor Y Pan. 2000. Approximating complex polynomial zeros: modified Weyl's quadtree construction and improved Newton's iteration. *J. of Complexity* 16, 1 (2000), 213–264.

[12] Victor Y Pan. 2002. Univariate polynomials: nearly optimal algorithms for numerical factorization and root-finding. *J. of Symb. Comp.* 33, 5 (2002), 701–733.

[13] Victor Y. Pan. 2018. New Acceleration of Nearly Optimal Univariate Polynomial Root-finders. (2018). arXiv:cs.NA/1805.12042, last revised May 2020

[14] Victor Y. Pan. 2019. Old and New Nearly Optimal Polynomial Root-Finders. In *Computer Algebra in Scientific Computing*, Matthew England, Wolfram Koepf, Timur M. Sadykov, Werner M. Seiler, and Evgenii V. Vorozhtsov (Eds.). Springer International Publishing, Cham, 393–411.

[15] Victor Y. Pan and Elias P. Tsigaridas. 2013. On the Boolean Complexity of Real Root Refinement. In *Proceedings of the 38th International Symposium on Symbolic and Algebraic Computation (ISSAC '13)*. ACM, New York, NY, USA, 299–306. https://doi.org/10.1145/2465506.2465938

[16] Victor Y Pan and Elias P Tsigaridas. 2016. Nearly optimal refinement of real roots of a univariate polynomial. *J. of Symb. Comp.* 74 (2016), 181–204.

[17] James Renegar. 1987. On the worst-case arithmetic complexity of approximating zeros of polynomials. *J. of Complexity* 3, 2 (1987), 90–113.

[18] Siegfried M Rump. 2010. Verification methods: Rigorous results using floating-point arithmetic. *Acta Numerica* 19 (2010), 287–449.

[19] Michael Sagraloff and Kurt Mehlhorn. 2016. Computing real roots of real polynomials. *J. of Symb. Comp.* 73 (2016), 46–86.

[20] Arnold Schönhage. 1982. The fundamental theorem of algebra in terms of computational complexity. *Manuscript. Univ. of Tübingen, Germany* (1982).

[21] Juan Xu and Chee Yap. 2019. Effective subdivision algorithm for isolating zeros of real systems of equations, with complexity analysis. In *Proceedings of the 2019 on International Symposium on Symbolic and Algebraic Computation*. 355–362.

On FGLM Algorithms with Tropical Gröbner bases

Yuki Ishihara
Graduate School of Science, Rikkyo University
Tokyo, Japan
yishihara@rikkyo.ac.jp

Tristan Vaccon
Université de Limoges; CNRS, XLIM UMR 7252
Limoges, France
tristan.vaccon@unilim.fr

Kazuhiro Yokoyama
Departement of Mathematics, Rikkyo University
Tokyo, Japan
kazuhiro@rikkyo.ac.jp

ABSTRACT

Let K be a field equipped with a valuation. Tropical varieties over K can be defined with a theory of Gröbner bases taking into account the valuation of K. Because of the use of the valuation, the theory of tropical Gröbner bases has proved to provide settings for computations over polynomial rings over a p-adic field that are more stable than that of classical Gröbner bases. In this article, we investigate how the FGLM change of ordering algorithm can be adapted to the tropical setting.

As the valuations of the polynomial coefficients are taken into account, the classical FGLM algorithm's incremental way, monomomial by monomial, to compute the multiplication matrices and the change of basis matrix can not be transposed at all to the tropical setting. We mitigate this issue by developing new linear algebra algorithms and apply them to our new tropical FGLM algorithms.

Motivations are twofold. Firstly, to compute tropical varieties, one usually goes through the computation of many tropical Gröbner bases defined for varying weights (and then varying term orders). For an ideal of dimension 0, the tropical FGLM algorithm provides an efficient way to go from a tropical Gröbner basis from one weight to one for another weight. Secondly, the FGLM strategy can be applied to go from a tropical Gröbner basis to a classical Gröbner basis. We provide tools to chain the stable computation of a tropical Gröbner basis (for weight $[0, \ldots, 0]$) with the p-adic stabilized variants of FGLM of [RV16] to compute a lexicographical or shape position basis.

All our algorithms have been implemented into SageMath. We provide numerical examples to illustrate time-complexity. We then illustrate the superiority of our strategy regarding to the stability of p-adic numerical computations.

CCS CONCEPTS

• **Computing methodologies → Algebraic algorithms**.

KEYWORDS

Algorithms, Tropical Geometry, Gröbner bases, FGLM algorithm, p-adic precision

ISSAC '20, July 20–23, 2020, Kalamata, Greece

ACM ISBN 978-1-4503-7100-1/20/07…$15.00
https://doi.org/10.1145/3373207.3404037

ACM Reference Format:
Yuki Ishihara, Tristan Vaccon, and Kazuhiro Yokoyama. 2020. On FGLM Algorithms with Tropical Gröbner bases. In *International Symposium on Symbolic and Algebraic Computation (ISSAC '20), July 20–23, 2020, Kalamata, Greece.* ACM, New York, NY, USA, 8 pages. https://doi.org/10.1145/3373207.3404037

1 INTRODUCTION

The development of tropical geometry is now more than three decades old. It has generated significant applications to very various domains, from algebraic geometry to combinatorics, computer science, economics, optimisation, non-archimedean geometry and many more. We refer to [MS15] for a complete introduction.

Effective computation of tropical varieties are now available using Gfan and Singular (see [JRS19] , [GRZ19]). Those computations often rely on the computation of so-called tropical Gröbner bases (we use *GB* for Gröbner bases in the following). Since Chan and Maclagan's definition of tropical Gröbner bases taking into account the valuation in [CM19], computations of tropical GB are available over fields with trivial or non-trivial valuation, using various methods: Matrix F5 in [Va15], F5 in [VY17, VVY18] or lifting in [MR19].

An important motivation for studying the computation of tropical GB is their numerical stability. It has been proved in [Va15] that for polynomial ideals over a p-adic field, computing tropical GB (which by definition take into account the valuation), can be significantly more stable than classical GB.

Unfortunately, no tropical term ordering can be an elimination order, hence tropical GB can not be used directly for solving polynomial systems. Our work is then motivated by the following question: can we take advantage of the numerical stability of the computation of tropical GB to compute a shape position basis in dimension zero through a change of ordering algorithm?

In this article, we tackle this problem by studying the main change of ordering algorithm, FGLM [FGLM93]. On the way, we investigate some adaptations and optimizations of this algorithm designed to take advantage of some special properties of the ideal (*e.g.* Borel-fixedness of its initial ideal).

We also provide a way to go from a tropical term order to another. This produces another motivation: difficulty of computation can vary significantly depending on the term order (see §8.1 of [VVY18]), hence, using a tropical FGLM algorithm, one could go from an easy term order to a harder one in an efficient way.

Finally, we conclude with numerical data to estimate the loss in precision for the computation of a lex Gröbner basis using a tropical F5 algorithm followed by an FGLM algorithm, in an affine setting, and also numerical data to illustrate the behavior of the various variants of FGLM handled along the way.

1.1 Related works

Chan and Maclagan have developed in [CM19] a Buchberger algorithm to compute tropical GB for homogeneous input polynomials (using a special division algorithm). Following their work, adaptations of the F5 strategies have been developped in [Va15, VY17, VVY18] culminating with complete F5 algorithms for affine input polynomials.

A completely different approach has been developped by Markwig and Ren in [MR19], relating the computation of tropical GB in $K[X_1, \dots, X_n]$ to the computation of standard basis in $R[\![t]\!][X_1, \dots, X_n]$ (for R a subring of the ring of integers of K). It can be connected to the Gfanlib interface in Singular to compute tropical varieties (see: [JRS19]).

Finally, Görlach, Ren and Zhang have developped in [GRZ19] a way to compute zero-dimensional tropical varieties using shape position bases and projections. Their algorithms take as input a lex Gröbner basis in shape position. Our strategies can be used to provide such a basis stably (precision-wise) when working with p-adic numbers, and be chained with their algorithms.

1.2 Notations

Let K be a field with a discrete valuation val such that K is complete with respect to the norm defined by val. We denote by $R = O_K$ its ring of integers, m_K its maximal ideal (with π a uniformizer), and $k = O_K/m_K$ its fraction field. We refer to Serre's Local Fields [Ser79] for an introduction to such fields. Classical examples of such fields are $K = \mathbb{Q}_p$, with p-adic valuation, and $\mathbb{Q}((X))$ or $\mathbb{F}_q((X))$ with X-adic valuation.

The polynomial ring $K[X_1, \dots, X_n]$ (for some $n \in \mathbb{Z}_{>0}$) will be denoted by A, and for $u = (u_1, \dots, u_n) \in \mathbb{Z}^n_{\geq 0}$, we write x^u for $X_1^{u_1} \dots X_n^{u_n}$. For $g \in A$, $|g|$ denotes the total degree of g and $A_{\leq d}$ the set of all polynomials in A of total degree less than d. The matrix of a finite list of polynomials (of total degree $\leq d$ for some d) written in a basis of monomials (of total degree $\leq d$) is called a *Macaulay matrix*.

For $w \in Im(\mathrm{val})^n \subset \mathbb{R}^n$ and $\leq_m$ a monomial order on A, we define $\leq$ a tropical term order as in the following definition:

Definition 1.1. Given $a, b \in K^* = K \setminus \{0\}$ and x^α and x^β two monomials in A, we write $ax^\alpha < bx^\beta$ if:

- $|x^\alpha| < |x^\beta|$, or
- $|x^\alpha| = |x^\beta|$, and $\mathrm{val}(a) + w \cdot \alpha > \mathrm{val}(b) + w \cdot \beta$, or
- $|x^\alpha| = |x^\beta|$, $\mathrm{val}(a) + w \cdot \alpha = \mathrm{val}(b) + w \cdot \beta$ and $x^\alpha <_m x^\beta$.

For u of valuation 0, we write $ax^\alpha =_\leq uax^\alpha$. Accordingly, $ax^\alpha \leq bx^\beta$ if $ax^\alpha < bx^\beta$ or $ax^\alpha =_\leq bx^\beta$.

Leading terms (LT) and leading monomials (LM) are defined according to this term order. See Subsec. 2.3 of [VVY18] for more information on this definition and its comparison with Def. 2.3 of [CM19].

Let $I \subset A$ be a 0-*dimensional*. Let $B_\leq$ the canonical linear K-basis of A/I made of the $x^\alpha \notin LM_\leq(I)$. Let δ be the cardinality of $B_\leq$. We denote by $\mathscr{B}_\leq$ the border of $B_\leq$ (*i.e.* the $x_k x^\alpha$ for $k \in [\![1, n]\!]$ such that $x^\alpha \in B_\leq$ and $x_k x^\alpha$ not in $B_\leq$). $NF_\leq$ is the normal form mapping defined by I and $\leq$. We define D such that $D = 1 + \max_{x^\alpha \in B_\leq} |x^\alpha|$.

2 MULTIPLICATION MATRICES

The first task in the FGLM strategy is to develop the tools for computations in A/I. The main ingredients are the multiplication matrices, $M_1, \dots, M_n$, corresponding to the matrices of the linear maps given by the multiplication by x_i written in the basis $B_\leq$.

Once they are known, it is clear that one can perform any K-algebra operation on elements of A/I written in the basis $B_\leq$.

To compute those matrices, a natural strategy is to go through the computation of the normal forms $NF(x_i x^\alpha)$ for $x^\alpha \in B_\leq$.

We investigate in this section how to proceed with this task, and how it compares to the classical case.

2.1 Linear algebra

We recall here the tropical row-echelon form algorithm of [Va15] that we use for computing normal forms using linear algebra.

Algorithm 1: The tropical row-echelon form algorithm

input : M, a Macaulay matrix of degree d in A, with n_{row} rows and n_{col} columns, and *mon* a list of monomials indexing the columns of M.

output : $\widetilde{M}$, the U of the tropical LUP-form of M

$\widetilde{M} \leftarrow M$;

for $i = 1$ *to* n_{row} **do**

  Find j such that $\widetilde{M}[i, j]$ has the greatest term $\widetilde{M}[i, j]x^{mon_j}$ for $\leq$ of the row i ;

  Swap the columns i and j of $\widetilde{M}$, and the i and j entries of *mon* ;

  By **pivoting** with the i-th row, eliminates the coefficients of the other rows on the first column; ;

Return $\widetilde{M}$;

We refer the interested reader to [Va15, VVY18]. We illustrate this algorithm with the following example.

Example 2.1. We present the following Macaulay matrices, over $\mathbb{Q}_3[x, y]$ with $w = (0, 0)$, and $\leq_m$ be the graded lexicographical ordering. The second one is the output of the tropical LUP algorithm applied on the first one. The monomials indexing the columns are written on top of the matrix.

$$\begin{array}{|cccccc|}
x^4 & x^3y & y^4 & x^2 & xy & y^2 \\
\hline
1 & & & & 3 & \\
 & & & 1 & 9 & 3 \\
 & 9 & 9 & & & \\
 & 9 & 9 & 3 & 1 & 9
\end{array}
\qquad
\begin{array}{|cccccc|}
x^4 & x^2 & x^3y & xy & y^4 & y^2 \\
\hline
1 & & & & 3 & \\
 & 1 & & 0 & & -\frac{57}{35} \\
 & & 9 & 0 & 9 & -\frac{162}{35} \\
 & & & -35 & 0 & -18
\end{array}.$$

If all four polynomials represented by the matrix belong to some ideal I (and assuming that $y^4, y^2 \in B_\leq(I)$) then we can conclude that $NF_\leq(xy) = -\frac{18}{35}y^2$ and $NF_\leq(x^3y) = -y^4 + \frac{18}{35}y^2$.

2.2 Comparison with classical case

The classical strategy to compute the $NF_{\leq_m}(x_i x^\alpha)$ ($x^\alpha \in B_{\leq_m}$) when working with a monomial ordering $\leq_m$, starting with a reduced GB G, is to set apart the following only three cases possible:

(Type 1) $x_i x^\alpha \in B_{\leq_m}$; **(Type 2)** $x_i x^\alpha \in LT(G)$;
(Type 3) $x_i x^\alpha \in LT_{\leq_m}(I)$ but neither in $B_{\leq_m}$ nor in $LT(G)$.

Type 1 is the easiest, as in this case $NF_{\leq_m}(x_i x^\alpha) = x_i x^\alpha$. Type 2 is not very difficult either. If for some $g \in G$, $LM(g) = x_i x^\alpha$, $g = x_i x^\alpha + \sum_{x^\beta \in B_{\leq_m}} c_\beta x^\beta$, then as G is reduced, we get directly that $NF_{\leq_m}(x_i x^\alpha) = -\sum_{x^\beta \in B_{\leq_m}} c_\beta x^\beta$.

Type 3 is the trickiest. We assume that we have already computed all the $NF(x_j x^\beta)$ for $x_j x^\beta <_m x_i x^\alpha$. Let x_k be the smallest (for $\leq_m$) variable dividing $x_i x^\alpha$. Then the normal form

$$NF\left(\frac{x_i x^\alpha}{x_k}\right) = \sum_{x^\beta \in B_{\leq_m},\ x^\beta <_m \frac{x_i x^\alpha}{x_k}} c_\beta x^\beta$$

is already known. As in the previous sum, $x^\beta <_m \frac{x_i x^\alpha}{x_k}$, then $x_k x^\beta <_m x_i x^\alpha$, and all the $NF(x_k x^\beta)$'s are also already known. Therefore, we can write

$$NF(x_i x^\alpha) = \sum_{x^\beta \in B_{\leq_m},\ x^\beta <_m \frac{x_i x^\alpha}{x_k}} c_\beta NF(x_k x^\beta),$$

and $NF(x_i x^\alpha)$ can be obtained from the previous normal forms.

It is easy to see that the cost of computation of a normal form in the third case is in $O(\delta^2)$ field operations. The other two cases are negligible. As there are $O(n\delta)$ multiples to consider, the total cost for the computation of the multiplication matrices is in $O(n\delta^3)$ field operations.

Unfortunately, this strategy can not be completely generalized to the tropical context. There is no issue with the first two computations. However, there is no straightforward way to adapt the third one. We illustrate this failure with the following example.

Example 2.2. Over $\mathbb{Q}_3[x, y]$ with $\leq$ defined by $w = (0, 0)$, and $\leq_m$, the graded lexicographical ordering, let us take $I = \langle f_1, f_2, f_3, f_4\rangle$ with $f_1 = x^7$, $f_2 = x^4y^2 + 3x^5y + 12x^3y^3 + 9xy^5$, $f_3 = x^2y^4 + 9x^5y + 18x^3y^3 + 9xy^5$, $f_4 = y^6 + 12x^5y + 3x^3y^3 + 6xy^5$. The first monomials of the third type arrive in degree 7, namely $xy^6, x^2y^5, x^4y^3, x^5y^2$. Due to the fact that we use a tropical term order, f_2, f_3, and f_4 all involve the monomials x^5y, x^3y^3, xy^5. In consequence if one wants to use multiples of the $NF(x^4y^2)$, $NF(x^2y^4)$, $NF(y^6)$, one gets quantity involving each three monomials among xy^6, x^2y^5, x^4y^3, and x^5y^2. They are all intertwined, and the trick we saw previously for monomials of the third type can not be used.

2.3 Tropical GB: General case

To untangle the reduction of monomials of the third type, we can use linear algebra. We have to proceed degree by degree. While monomials of the first type do not need any special proceeding, we need to interreduce the reductions of the monomials of the second and third types. The general strategy is described in Algorithm 2.

Proposition 2.3. *Algorithm 2 is correct, and is in $O(n^3\delta^3)$ field operations over K.*

Proof. The essentially different part compared to the classical case starts on Line 13. Lines 16 and 18 are crucial. By definition, monomials of the third type are in $LT(I)$. If $x^\alpha \in \overline{L}$ can not be written as $x_k x^\beta$ with x^β of type 2 or 3, it means that all its divisors are in $B_{\leq}$. Consequently, it is a minimal generator ot $LT(I)$ and is of type 2, which is a contradiction. Therefore, any monomial of the third type is a simple multiple of a monomial of type 2 or 3.

As in the **for loop** on Line 14, we proceed by increasing degree, it is an easy induction to prove that such desired x_i and g exist.

For the complete set of reducers on Line 18, we use the fact that the monomials appearing in $\mathscr{M}$ all are in $B_{\leq} \cup L$, again by an easy induction (using the fact that the rows of $\mathscr{M}$ in previous degree

Algorithm 2: Multiplication matrices computation algorithm

input : A reduced GB G of the ideal I for $\leq$, a tropical term ordering.
output : $M_1, \ldots, M_n$ the multiplication matrices of A/I (over the basis $B_{\leq}$).

1 Using $LT(G)$, compute $B_{\leq}$ (and $\delta = \sharp(B_{\leq})$);
2 Define $M_1, \ldots, M_n$ as zero matrices in $K^{\delta\times\delta}$, their rows and columns are indexed by the $x^\alpha \in B_{\leq}$;
3 Compute $L = \{x_i x^\alpha,\ \text{for } i \in [\![1, n]\!] \text{ and } x^\alpha \in B_{\leq}\}.$;
4 Compute $\overline{L} = L \cap (B_{\leq} \cup LT(G))^c$;
5 **for** $x^\alpha \in L \cap B_{\leq}$ **do**
6 **for** *i such that x_i divides x^α* **do**
7 Set $M_i[x^\alpha, \frac{x^\alpha}{x_i}] = 1$;
  /* The column indexed by $\frac{x^\alpha}{x_i}$ is zero, except on its coefficient indexed by x^α/x_i */
8 **for** $x^\alpha \in L \cap LT(G)$ **do**
9 Take $g \in G$ such that g can be written $g = x^\alpha + \sum_{x^\beta \in B_{\leq}} g_{x^\beta} x^\beta$;
10 **for** *i such that x_i divides x^α* **do**
11 **for** $x^\beta \in B_{\leq}$ **do**
12 Set $M_i[x^\beta, \frac{x^\alpha}{x_i}] = -g_{x^\beta}$;
13 Set $\mathscr{M}$ to be a matrix over K with 0 rows and with columns indexed by $\overline{L} \cup LT(G) \cup B_{\leq}$. ;
14 **for** *d a degree of a monomial in $\overline{L}$ (in ascending order)* **do**
15 **for** *$x^\alpha \in \overline{L}$ of degree d* **do**
16 Find x_i, and g either in G or as a row of $\mathscr{M}$ such that $LT(x_i g) = x^\alpha$;
17 Stack $x_i g$ at the bottom of $\mathscr{M}$;
18 Using multiples of the form $x_i g$ or g, for g either in G or as a row of $\mathscr{M}$, find a complete set of reducers for all the monomials in $\overline{L} \cup LT(G)$ appearing with a non-zero coefficient in their column, and stack them at the bottom of $\mathscr{M}$;
19 Compute the Tropical Row-echelon form of $\mathscr{M}$ by Algorithm 1 and replace $\mathscr{M}$ with it ;
20 **for** $x^\alpha \in \overline{L}$ **do**
21 Take the row s of $\mathscr{M}$ with leading coefficient x^α. ;
22 **for** *i such that x_i divides x^α* **do**
23 **for** $x^\beta \in B_{\leq}$ **do**
24 Set $M_i[x^\beta, \frac{x^\alpha}{x_i}] = -\frac{\mathscr{M}[s,x^\beta]}{\mathscr{M}[s,x^\alpha]}$;
25 **Return** $M_1, \ldots, M_n$

are already reduced), and therefore, the complete set of reducers can be built.

The Tropical Row-echelon form computation then produces the desired normal forms. The correctness is then clear.

Regarding to the arithmetic complexity, we should note that both rows and columns of $\mathscr{M}$ are indexed by monomials in $L \cup B_{\leq}$ and there are $O(n\delta)$ of them. With the row-reduction, the total cost is then in $O(n^3\delta^3)$ arithmetic operations. □

Remark 2.4. The matrix $\mathscr{M}$ is sparse: any row added to the matrix on Line 17 has at most $\delta + 1$ non-zero coefficients: it is obtained as the multiple of a reduced row. Can we take advantage of this $\frac{1}{n}$ sparsity ratio for a better complexity?

Example 2.5. Let $G = (y + 2x, x^2 + 4)$ be a GB for $w = [0, 0]$ and grevlex of the ideal it spans in $\mathbb{Q}_2[x, y]$. Then $B_{\leq} = \{1, x\}$, $L = \{x, y, x^2, xy\}$ and $\overline{L} = \{xy\}$. Only $d = 2$ is considered on Line 4 of Algorithm 2. The following matrices represent respectively $\mathscr{M}$ before and after applying Algorithm 1, M_1 and M_2:

$$\begin{array}{ccc|ccc|cc|cc|cc|cc} x^2 & xy & 1 & xy & x^2 & 1 & (x*) & 1 & x & (y*) & 1 & x \\ \hline 2 & 1 & 0 & 1 & 0 & -8 & 1 & 0 & -4 & 1 & 0 & 8 \\ 1 & 0 & 4 & 0 & 1 & 4 & x & 1 & 0 & x & -2 & 0 \end{array}.$$

2.4 Finite precision

We can now analyze the loss in precision when applying Algorithms 1 and 2. To prevent loss in precision to explode exponentially, we replace Line 5 of Algorithm 1 with the following two rows:

(1) By pivoting using the 'leading terms' of the rows j for $j > i$, eliminate all the coefficients possible of row i ;
(2) By pivoting with row i, eliminate all the coefficients on the i-th column.

The first row makes sense because by construction, all the rows of $\mathscr{M}$ have distinct leading terms, and this is kept unchanged during the pivoting process.

Proposition 2.6. *Let us assume that the matrix built on Line 17 of Algorithm 2 has coefficients in K known at precision $O(\pi^N)$. All rows have distinct leading terms, leading coefficient 1 and let us take Ξ be the smallest valuation of a coefficient of this matrix $\mathscr{M}$. We assume that $\Xi \leq 0$. Let $l = rank(\mathscr{M})$. We assume that $N > -l^2\Xi$. Then, after the application of Algorithm 1[1], the coefficients of the obtained matrix $\tilde{\mathscr{M}}$ are known at precision $O(\pi^{N+l^2\Xi})$, and the smallest valuation of a coefficient $\tilde{\mathscr{M}}$ is lower-bounded by $l\Xi$.*

Proof. After the reduction of row 1 by the other rows, the smallest valuation on row 1 is lower-bounded by $l\Xi$ and its coefficients are known at precision at least $O(\pi^{N+l\Xi})$. The coefficients of row 1 for the columns indexed by $\overline{L} \cup LT(G)$ are all zeros, except for its leading coefficient, which is $1 + O(\pi^{N+(l-1)\Xi})$. After the reduction of the other rows by row 1, on the rows of index > 1, the coefficients for the columns indexed by $\overline{L} \cup LT(G)$ are of valuation at least Ξ and known at precision $O(\pi^{N+l\Xi})$. The coefficients for the columns indexed by $B_{\leq}$ are of valuation at least $l\Xi$ and known at the same precision. The desired result follows by an easy induction argument. □

[1] using the modification presented just above this proposition

We then upper-bound the loss in precision for the whole computation of the multiplication matrices. Recall that: $D = 1 + \max_{x^\alpha \in B_{\leq}} |x^\alpha|$.

Proposition 2.7. *Let us assume that the smallest valuation of a coefficient of G is Ξ and that the coefficients of G are known at precision $O(\pi^N)$. As G is reduced, we get that $\Xi \leq 0$.*

Then the coefficients of the matrices $M_1, \ldots, M_n$ are of valuation at least $(n\delta)^D\,\Xi$, and are known at precision $O\left(\pi^{N+\left(\frac{(n\delta)^{2D+2}-1}{(n\delta)^2-1}\right)\Xi}\right)$.

Proof. This is a corollary to the previous proposition. There are at most D calls to the previous proposition, with matrices of ranks $l_1, \ldots, l_D$. Consequently, the upper bound on the valuation is $l_1 \ldots l_D\Xi$ and the precision is in $O(\pi^{N+(l_1^2+l_1^2l_2^2+\cdots+l_1^2\ldots l_D^2)\Xi})$ which is in $O(\pi^{N+D(l_1^2\ldots l_D^2)\Xi})$ As for all i, $l_i \leq n\delta$, we get the desired bounds. □

Remark 2.8. In the very favorable case where G is homogeneous and $w = [0, \ldots, 0]$, we get that $\Xi = 0$, and no loss in precision is happening. This is unfortunately not the most interesting case for polynomial system solving. Numerical data in Section 5 will show that loss in precision remain very reasonnable when using $w = [0, \ldots, 0]$ even in the affine case.

2.5 Using semi-stability

Following Huot's PhD thesis [Huo13], when Borel-fixedness (see Subsec. 3.2) or semi-stability properties are satisfied, many arithmetic operations can be avoided during the computation of the multiplication matrices. We begin with semi-stability.

Definition 2.9. I is said to be semi-stable for x_n if for all x^α such that $x^\alpha \in LM(I)$ and $x_n \mid x^\alpha$ we have for all $k \in [\![1, n-1]\!]$ $\frac{x_k}{x_n}x^\alpha \in LM(I)$.

Semi-stability's application is explained in Proposition 4.15, Theorem 4.16 and Corollary 4.19 of [Huo13] (see also Section 4 of [FGHR14]). We recall the main idea here with its adaptation to the tropical setting:

Proposition 2.10. *If I is semi-stable for x_n, M_n can be read from G and requires no arithmetic operation.*

Proof. The proof is the same as that of Theorem 8 of [FGHR14]. We prove that $\overline{L} \cap x_nB_{\leq} = \emptyset$. Let $x_nx^\alpha \in \overline{L} \cap x_nB_{\leq}$, with $x^\alpha \in B_{\leq}$. Then there is some monomial m and $g \in G$ such that $LM(mg) = x_nx^\alpha$. As $x^\alpha \in B_{\leq}$, we get that $x_n \nmid m$. Since $x_nx^\alpha \in \overline{L}$, then $|m| \geq 1$. Let $k < n$ be such that $x_k \mid m$. Then, by semi-stability for x_n, $x^\alpha = \frac{m}{x_k} \times \frac{x_k LM(g)}{x_n} \in LM(I)$, which is a contradiction. □

Thanks to Proposition 2.10, Algorithm 3 is correct, and its arithmetic cost is given by the following proposition.

Proposition 2.11. *Given a reduced GB G of the ideal I for $\leq$, a tropical term ordering, and assuming I is semi-stable for x_n, then M_n can be computed in $O(\delta^2)$ arithmetic operations, which are only computing opposites.*

To apply the previous result to compute a GB in shape position in Subsection 4.2, we need to also compute the $NF(x_i)$'s. The following lemma states that this is not costly.

Algorithm 3: Computing M_n, when semi-stable for x_n

input : A reduced GB G of the ideal I for $\leq$, a tropical term ordering, assuming I is semi-stable for x_n
output : M_n the matrix of the multiplication by x_n in A/I

Using $LT(G)$, computes $B_\leq$ (and $\delta = \sharp(B_\leq)$);
Define M_n as a zero matrix in $K^{\delta\times\delta}$, its rows and columns are indexed by the $x^\alpha \in B_\leq$;
Compute $L_n = \{x_n x^\alpha, \text{ for } x^\alpha \in B_\leq\}$. ;
for $x^\alpha \in L_n \cap B_\leq$ **do**
  Set $M_n[x^\alpha, \frac{x^\alpha}{x_n}] = 1$;
for $x^\alpha \in L_n \cap LT(G)$ **do**
  Take $g \in G$ such that g can be written $g = x^\alpha + \sum_{x^\beta \in B_<} g_{x^\beta} x^\beta$. **for** $x^\beta \in B_\leq$ **do**
    Set $M_n[x^\beta, \frac{x^\alpha}{x_i}] = -g_{x^\beta}$;
Return M_n ;

Lemma 2.12. *Given a reduced GB G of the ideal I for $\leq$, a tropical term ordering, then the $NF_\leq(x_i)$'s can be computed in $O(n\delta)$ arithmetic operations, which are only computing opposites.*

Proof. It is a consequence of the fact that $\leq$ is degree-compatible: for any i, x_i is either in $LT(G)$ or in $B_\leq$. □

Subsection 4.2 will apply the previous two results to obtain a fast algorithm to compute a shape-position basis.

Remark 2.13. For grevlex in the classical case, it is known that after a generic change of variable, I is semi-stable for x_n. The reason is that after a generic change of variable, $LT(I)$ is equal to the GIN of I (see Definition 4.1.3 of [HH11]) , which is known to be Borel-fixed, and Borel-fixedness implies semi-stability for x_n. In Section 3, we investigate whether this strategy is still valid in the tropical case.

3 GIN AND BOREL-FIXED INITIAL IDEAL

In this section, we introduce the tropical generic initial ideal of a 0-dimensional ideal analogously to the classical case, and study its properties of Borel-fixedness and semi-stability. The desired goal is to be able to use the fast Algorithm 3 after a (generic) change of variable.

3.1 Tropical GIN

We follow the lines of Chapter 4 of [HH11], and use the usual action of $\mathrm{GL}_n(K)$ on A: $(\eta, f(x)) \in \mathrm{GL}_n(K) \times A \mapsto \eta(f) := f(\eta^\top \cdot x)$.

Definition 3.1. An external product of monomials $x^{\alpha_1} \wedge \cdots \wedge x^{\alpha_k}$ is called a *standard exterior monomial* if $x^{\alpha_1} \geq \cdots \geq x^{a_k}$. If its monomial is standard, a term $cx^{\alpha_1} \wedge \cdots \wedge x^{\alpha_k}$ is called a *standard exterior term*. We define an ordering on standard exterior terms by setting that: $cx^{\alpha_1} \wedge \cdots \wedge x^{\alpha_k} \geq dx^{\beta_1} \wedge \cdots \wedge x^{\beta_k}$ if $\mathrm{val}(c) + \sum_{i=1}^k w \cdot \alpha_i < \mathrm{val}(d) + \sum_{i=1}^k w \cdot \beta_i$, or $\mathrm{val}(c) + \sum_{i=1}^k w \cdot \alpha_i = \mathrm{val}(d) + \sum_{i=1}^k w \cdot \beta_i$ and there exists $1 \leq j \leq k$ s.t. $x^{\alpha_j} > x^{\beta_j}$ and $x^{\alpha_i} = x^{\beta_i}$ for all $i < j$. We then define the leading term of an external product of polynomials $f_1 \wedge \cdots \wedge f_k$ as its largest term, and denote it by $LT(f_1 \wedge \cdots \wedge f_k)$. The monomial of the leading term is denoted by $LM(f_1 \wedge \cdots \wedge f_k)$.

Lemma 3.2. *Let $(f_1, \ldots, f_t) \in A^t$. If $LT(f_1) > \cdots > LT(f_t)$, then $LT(f_1 \wedge \cdots \wedge f_t) = LT(f_1) \wedge \cdots \wedge LT(f_t)$.*

Proof. Let c_i be the coefficient of $LM(f_i)$ in f_i. Then, $c = \prod c_i$ is the coefficient of $\Gamma = LT(f_1) \wedge \cdots \wedge LT(f_t)$ in $f_1 \wedge \cdots \wedge f_t$. We may assume that the f_i's are ordered such that $cLT(f_1) \wedge \cdots \wedge LT(f_t)$ is a standard exterior term. Let $\Delta = dv_1 \wedge \cdots \wedge v_t$ be another term in $f_1 \wedge \cdots \wedge f_t$ and d_i the coefficient of v_i in f_i. Let $x^{\alpha_i} = LM(f_i)$ and $x^{\beta_i} = v_i$. Since $c_i x^{\alpha_i}$ is the leading term of f_i, it follows that $\mathrm{val}(c_i) + w \cdot \alpha_i \leq \mathrm{val}(d_i) + w \cdot \beta_i$. Thus, $\sum_{i=1}^t (\mathrm{val}(c_i) + w \cdot \alpha_i) \leq \sum_{i=1}^t (\mathrm{val}(d_i) + w \cdot \beta_i)$. As $\mathrm{val}(c) = \sum_{i=1}^t c_i$ and $\mathrm{val}(d) = \sum_{i=1}^t d_i$, we obtain $\mathrm{val}(c) + \sum_{i=1}^k w \cdot \alpha_i \leq \mathrm{val}(d) + \sum_{i=1}^k w \cdot \beta_i$. If the inequality is strict then Γ is strictly bigger than any permutation of the monomials of Δ such that a standard exterior term is obtained. If equality holds. Then, for all i, $\mathrm{val}(c_i) + w \cdot \alpha_i = \mathrm{val}(d_i) + w \cdot \beta_i$ and $x^{\alpha_i} \geq x^{\beta_i}$. As Γ is a standard exterior term, we deduce that also in this case, Γ is strictly bigger than any permutation of the monomials of Δ such that a standard exterior term is obtained. □

Lemma 3.3. *Let $V \subset A$ be a t-dimensional K-vector space. Let $w_1, \ldots, w_t$ be monomials with $w_1 > \cdots > w_t$. Then the following conditions are equivalent.*

(1) the monomials $w_1, \ldots, w_t$ form a K-basis of $LT(V)$,
(2) if $(f_1, \ldots, f_t)$ is a K-basis of V, then $LM(f_1 \wedge \cdots \wedge f_t) = w_1 \wedge \cdots \wedge w_t$,
(3) there exists a K-basis $(f_1, \ldots, f_t)$ of V s.t. $LM(f_1 \wedge \cdots \wedge f_t) = w_1 \wedge \cdots \wedge w_t$.

Proof. (1) $\Rightarrow$ (2): We may assume that the f_j's are monic and $LT(f_1) > \cdots > LT(f_t)$. Since $LT(f_i) \in LT(V)$, there is $j(i)$ s.t. $LT(f_i) = w_{j(i)}$. As $w_1 >_1 \cdots >_1 w_t$, we obtain $j(i) = i$ and $LT(f_i) = w_i$ for all i. By Lemma 3.2, $LT(f_1 \wedge \cdots \wedge f_t) = LT(f_1) \wedge \cdots \wedge LT(f_t) = w_1 \wedge \cdots \wedge w_t$.

(2) $\Rightarrow$ (3): It is obvious by choosing a K-basis $f_1, \ldots, f_t$ of V.

(3) $\Rightarrow$ (1): Since $\dim(V) = \dim(LT(V))$ and $w_1, \ldots, w_t$ is linear independent, it is enough to show that $w_i \in LT(V)$. Let $f_1, \ldots, f_t$ be monic polynomials forming a K-basis of V with $LT(f_1) > \cdots > LT(f_t)$ and $LT(f_1 \wedge \cdots \wedge f_t) = w_1 \wedge \cdots \wedge w_t$. By Lemma 3.2, $LT(f_1 \wedge \cdots \wedge f_t) = LT(f_1) \wedge \cdots \wedge LT(f_t)$ and thus $w_i \in LT(V)$. □

Proposition 3.4. *Let $V \subset A_d$ be a t-dimensional K-vector space and $f_1, \ldots, f_t$ a basis of V. Let $cw_1 \wedge \cdots \wedge w_t$ be the largest (up to multiplication by an element of valuation 0) standard exterior term of $\bigwedge^t A_{\leq d}$ such that there exists $\eta \in \mathrm{GL}_n(R)$ with*

$$LT(\eta(f_1) \wedge \cdots \wedge \eta(f_t)) = cw_1 \wedge \cdots \wedge w_t.$$

Let $U_V = \{\eta \in \mathrm{GL}_n(R) \mid LT(\eta(f_1) \wedge \cdots \wedge \eta(f_t)) = \varepsilon \times cw_1 \wedge \cdots \wedge w_t,\ \mathrm{val}(\varepsilon) = 0\}$. Then, U_V is open in $\mathrm{GL}_n(R)$ and for any $\eta, \upsilon \in U_V$, $LT(\eta V) = LT(\upsilon V)$.

Proof. As only a finite amount of monomials are possible and $\mathrm{val}(R)$ is discrete and ≥ 0, U_V is well-defined. The valuation being discrete, U_V is open: $LT(\eta(f_1) \wedge \cdots \wedge \eta(f_t)) = \varepsilon \times cw_1 \wedge \cdots \wedge w_t$ amounts to $\mathrm{val}(q(\eta)) < \nu$ for carefully chosen $\nu \in \mathbb{R}$ and polynomial $q \in \mathbb{Z}[k^{n\times n}]$. The last statement follows from Lemma 3.3. □

From Lemma 3.3, $w_1 \wedge \cdots \wedge w_t$ in Prop 3.4 is independent of the choice of basis of V. For $d \in \mathbb{Z}_{\geq 0}$, let $I_{\leq d} = I \cap A_{\leq d}$.

Theorem 3.5. *Let I be a 0-dimensional ideal with $\delta = \dim_K K[X]/I$. We consider the finite dimensional K-vector space $I_{\leq\delta}$. Then the non-empty open set $U_I := U_{I_{\leq\delta}} \subset \mathrm{GL}_n(R)$ satisfies that $LT(\eta I) = LT(\upsilon I)$ for any $\eta, \upsilon \in U_I$.*

Proof. Let $\eta \in U_I$. We denote $LT(\eta I_{\leq d})$ by $J_{\leq d}$. Then $J_{\leq d} = LT(\upsilon I_{\leq d})$ for all $\upsilon \in U_I$ and $d > \delta$. Indeed, since $LT(\eta I_{\leq\delta})$ contains the initial terms in the reduced Gröbner basis G of ηI,

$$J_{\leq d} \subset A_{\leq d-\delta} LT(\eta I_{\leq\delta}) = A_{\leq d-\delta} LT(\upsilon I_{\leq\delta}) \subset LT(\upsilon I_{\leq d}).$$

As $\dim_K(J_d) = \dim_K(LT(\upsilon I_d))$, we obtain $J_d = LT(\upsilon I_d)$ for all $\upsilon \in U_I$. Since $LT(\eta I) = \bigcup_{d=\delta}^{\infty} J_{\leq d}$, then $LT(\eta I) = LT(\upsilon I)$ for any $\eta, \upsilon \in U_I$, which concludes the proof. □

Definition 3.6. We call $LM(\eta I)$, with $\eta \in U_I \subset \mathrm{GL}_n(R)$ as given in Theorem 3.5, the tropical generic initial ideal (tropical gin) of I.

Unfortunately, U_I is not a Zariski-open subset of $GL_n(R)$ in general, hence the *generic* in the name "tropical gin" is only given as a reference to the classical case. The following proposition is a consolation.

Proposition 3.7. *Assume k is infinite. Then*

$$U_I \bmod \pi := \{\eta \bmod \pi, \text{ for } \eta \in U_I\}$$

is a non-empty Zariski-open set of $GL_n(k)$.

Proof. Let q be the polynomial defining $U_{I_{\leq\delta}}$ in the proof of Theorem 3.5. One can replace q by some q/π^l so that $\overline{q} = q \bmod \pi$ is non-zero, and one can check that consequently, since k is infinite, $U_I \bmod \pi = \{\overline{x} \in \mathrm{GL}_n(k) : \overline{q}(\overline{x}) \neq 0\}$ and this is a non-empty Zariski-open set of $GL_n(k)$. □

Remark 3.8. If, *e.g.*, $R = \mathbb{R}[\![t]\!]$, and one takes $\eta \in GL_n(R)$ at random using a nonatomic distribution over $\mathbb{R}$, then η belongs to U_I with probability one.

3.2 Borel-fixedness

In classical cases, a generic initial ideal is Borel-fixed ideal i.e. it is fixed under the action of the Borel subgroup $\mathcal{B} \subset \mathrm{GL}_n(K)$, which is the subgroup of all nonsingular upper triangular matrices. In tropical cases, a generic initial ideal is not always Borel-fixed. However, it can be Borel-fixed under some conditions.

Example 3.9. Let $I = (x^2, y^2)$ and $K = \mathbb{Q}_2$ (using $w = [0, 0]$ and grevlex). Then in degree two, for a generic change of variables of $x^2 \wedge y^2$ by the matrix $\begin{bmatrix} a & b \\ c & d \end{bmatrix}$, we get in $K[x, y] \bigwedge K[x, y]$:

$$2(a^2bd - ab^2c)x^2 \wedge xy + (a^2d^2 - b^2c^2)x^2 \wedge y^2 + 2(acd^2 - bc^2d)xy \wedge y^2.$$

Hence the tropical GIN is $x^2 \wedge y^2$ for degree two, and is therefore not Borel-fixed, nor semi-stable for y.

Definition 3.10. Let $\mathfrak{B} \subset \mathrm{GL}_n(O_K)$ be the subgroup generated by nonsingular upper triangular matrices whose diagonal entries have valuation 0. We call $\mathfrak{B}$ a Borel subgroup. We say that a monomial ideal J is tropical Borel-fixed if J is fixed under the action of $\mathfrak{B}$.

A direct adaptation of Theorem 4.2.1 and Prop. 4.2.4 of [HH11] states that the usual properties of the GIN are preserved, under some conditions.

Proposition 3.11. *Let d be the maximal total degree of the reduced GB of the tropical generic initial ideal of I. If $K = \mathbb{Q}_p$ and $p \geq d$, or if $\mathrm{val}(\mathbb{Z} \setminus \{0\}) = \{0\}$, then the tropical generic initial ideal of I is tropical Borel-fixed and moreover, semi-stable for x_n.*

4 TROPICAL FGLM

In this section, we investigate the second part of the FGLM strategy. Namely, the multiplication matrices of A/I have been computed using the algorithms of Section 2, and we can now perform operations in A/I efficiently.

The strategy is then to go through projections in A/I of monomials and find linear relations among them. When done carefully, these relations provide polynomials in I, whose leading terms for the new term order can be read on the monomials defining the relation. When processed in the right order, we can obtain from these polynomials a minimal GB of I for our new term order.

4.1 Tropical to classical

We first begin with the easiest case of starting from a tropical GB and computing a classical GB.

It is clear that once the multiplication matrices are obtained, we can directly apply the classical FGLM algorithm (namely Algorithm 4.1 of [FGLM93], see also Algorithm 8 of [Huo13]), or its p-adic stabilized version: Algorithm 3 of [RV16]. This part is in $O(n\delta^3)$ arithmetic operations. We refer to Prop 3.6 of *loc. cit.* and obtain the following propositions.

Proposition 4.1. *The total complexity to compute a classical GB of I starting from a tropical GB is in $O(n^3\delta^3)$ arithmetic operations.*

Behavior regarding to precision can be stated the following way.

Proposition 4.2. *Let $\leq_1$ be a tropical term ordering and $\leq_2$ be a monomial ordering. Let G be an approximate reduced tropical GB for $\leq_1$ of the ideal I, with coefficients known up to precision $O(\pi^N)$. Let Ξ be the smallest valuation of a coefficient in G. Let $B_{\leq_1}$ and $B_{\leq_2}$ be the canonical bases of A/I for $\leq_1$ and $\leq_2$. Let M be the matrix whose columns are the $NF_{\leq_1}(x^\beta)$ for $x^\beta \in B_{\leq_2}$. Let $cond_{\leq_1,\leq_2}(I)$ be the biggest valuation of an invariant factor in the Smith Normal Form of M. Recall that $D = 1 + \max_{x^\alpha \in B_{\leq}} |x^\alpha|$.*

Then if $N > 2cond_{\leq_1,\leq_2}(I) - \left(\frac{(n\delta)^{2D+2}-1}{(n\delta)^2-1}\right)\Xi$, we can chain Algorithm 2 and Algorithm 3 of [RV16] to obtain an approximate GB G_2 of I for $\leq_2$. The coefficients of the polynomials of G_2 are known up to precision $O\left(\pi^{N+\left(\frac{(n\delta)^{2D+2}-1}{(n\delta)^2-1}\right)\Xi - 2cond_{\leq_1,\leq_2}(I)}\right)$.

4.2 Tropical to shape position

We can apply any classical FGLM algorithm if K is an exact field, or a stabilized variant using Smith Normal Form, as in Algorithm 6 of [RV16]. We refer to Prop. 4.5 of *loc. cit.*. Complexity is very favorable when we have the combination of Borel-fixedness and shape position.

Proposition 4.3. *If I is in shape position and semi-stable for x_n, then we can combine Algorithm 3 with Algorithm 6 of [RV16]). The time-complexity is in $O(n\delta^2) + O(\delta^3)$ arithmetic operations.*

Proposition 4.4. *Let G_1 be an approximate reduced GB of I, with coefficients known at precision $O(\pi^N)$. Let Ξ be the smallest valuation of a coefficient in G_1. If $\leq_2$ is lex, and if we assume that the ideal I is in shape position and $LM_{\leq_1}(I)$ is semi-stable for x_n, then the adapted FGLM in Algorithm 6 of [RV16]), computes an approximate GB G_2 of I for lex, in shape position. The coefficients of the polynomials of G_2 are known up to precision $O(\pi^{N-2cond_{\leq_1,\leq_2}+\delta\Xi})$. Moreover, we can read on M whether the precision was enough or not, and hence prove after the computation that the result is indeed an approximate GB.*

4.3 Tropical (or classical) to tropical

We conclude our series of algorithms with a new algorithm to compute a tropical GB of I of dimension 0 knowing the multiplication matrices of A/I.

In the classical case, the vanilla FGLM algorithm goes through the monomials x^α in ascending order for $\leq_2$, test whether x^α is in the vector space generated (in A/I) by the monomials x^β such that $x^\beta <_2 x^\alpha$, and if so, produce a polynomial in the GB in construction from the relation obtained by this linear relation.

In the tropical case, because of the fact that coefficients have to be taken into account, a relation (in A/I) between x^α and some monomials x^β such that $x^\beta <_2 x^\alpha$ is not enough to ensure that $x^\alpha \in LT_{\leq_2}(I)$. We deal with this issue by (1) taking all monomials of a given degree at the same time, in a big Macaulay matrix, and (2) reducing them with a special column-reduction algorithm so as to preserve the leading terms.

The linear algebra algorithm is presented in Algorithm 5, with the general tropical FGLM algorithm in Algorithm 4.

Algorithm 4: A tropical FGLM algorithm

input : $M_1, \ldots, M_n$ the multiplication matrices of A/I, in a basis $B_{\leq_1}$ for a tropical term ordering $\leq_1$, a tropical term ordering $\leq_2$.
output : A GB G of the ideal I for $\leq_2$.

1 $L \leftarrow \{1\}$, $G \leftarrow \emptyset$, $d \leftarrow 1$;
2 $M \leftarrow$ the matrix with δ rows and 0 columns ;
3 $P \leftarrow$ the matrix with 0 rows and 0 columns ;
4 **while** $L \neq \emptyset$ **do**
5 Stack on the right of M all the monomials in L of degree d, written in the basis $B_{\leq_1}$ using the multiplication matrices ;
6 Remove those monomials from L ;
7 Apply Algorithm 5 with M and $\leq_2$, to get a new M and update the pivoting matrix P ;
/* If M_0 is the matrix of the $NF_{\leq_1}(x^\alpha)$ for x^α indexing the columns of M, then $M = M_0 P$. */
8 For all the new columns indexed by x^α that reduced to zero, add to G the polynomial $x^\alpha - \sum_{\gamma \neq \alpha} P_{\gamma,\alpha} x^\gamma$, and remove the multiples of x^α from L ;
9 Add to L the $x_i x^\alpha$ for all i and for all x^α new column in M that did not reduce to zero, and remove the duplicates ;
10 $d \leftarrow d + 1$;
11 **Return** G

Algorithm 5: Column reduction for FGLM

input : M a $\delta \times l$ matrix over K, whose rows and columns are indexed by monomials. A tropical term ordering $\leq$. An invertible $s \times s$ matrix P.
output : A column-reduction of M compatible with $\leq$, an updated P.

1 **if** $M = 0$ **then** Return M, P ;
2 Find the coefficient $M[i,j]$ of row indexed by x^β and column indexed by x^α such that $M[i,j]^{-1} x^\alpha$ is smallest, and using smallest x^β to break ties ;
3 Use this non-zero coefficient to eliminate the other coefficients on the same row ;
4 Update P accordingly ;
5 Proceed recursively on the remaining rows and columns ;
6 **Return** M, P

The fact that Algorithm 5 computes a column-echelon form of the matrix (up to column-swapping) along with the pivoting matrix is clear. What is left to prove is the compatibility of the pivoting process with the computation of the normal forms and the leading terms according to $\leq_2$. It relies on the following loop-invariant.

Proposition 4.5. *At any point during the execution of Algorithm 5, for any x^α, the column of M indexed by x^α corresponds to the normal form $NF_{\leq_1}(H)$ (with respect to $\leq_1$) of some polynomial H with $LT_{\leq_2}(H) = x^\alpha$.*

Proof. It is true by construction for any column when entering Algorithm 5. Also by construction, all columns are labelled by distinct monomials. Now let us assume that on Line 4, we are eliminating a coefficient d on the column labelled by x^β using a coefficient c on the column labelled by x^α as pivot. Because of the choice of pivot on Line 3, we get that $c^{-1}x^\alpha <_2 d^{-1}x^\beta$. Let us assume that the column indexed by x^α corresponds to $NF_{\leq_1}(H)$ with $LT_{\leq_2}(H) = x^\alpha$, and the column indexed by x^β corresponds to $NF_{\leq_1}(Q)$ with $LT_{\leq_2}(Q) = x^\beta$. Please note that $x^\alpha \neq x^\beta$. Then after pivoting the second column corresponds to $NF_{\leq_1}(Q - dc^{-1}H)$. As $LT_{\leq_2}(dc^{-1}H) = dc^{-1}x^\alpha <_2 x^\beta$, the loop-invariant is then preserved, which is enough to conclude the proof. □

Theorem 4.6. *Algorithm 4 terminates and is correct: its output is a GB of the ideal I for $\leq_2$. It requires $O(n\delta^3)$ arithmetic operations.*

Proof. We use the following loop-invariant: after Line 9 is executed, $LT_{\leq_2}(G)$ contains all the minimal generators in $LT_{\leq_2}(I)$ of degree $\leq d$, they each correspond to a reduced-to-zero column of M, and the x^β corresponding to non-reduced-to-zero columns of M are all in $NS_{\leq_2}(I)$. The proof for this invariant is as follows. As $\leq_2$ is degree-compatible, it is clear by linear algebra that $rank(M) = \dim(A_{\leq d}/I_{\leq d})$. Thanks to Proposition 4.5, the polynomials added to G are in I, and more precisely, $f = x^\alpha - \sum_\gamma P_{\gamma,\alpha} x^\gamma$ as in Line 8 is a polynomial such that $LT_{\leq_2}(f) = x^\alpha$ and $NF_{\leq_1}(f) = 0$, as given in the Proposition. Their $LT_{\leq_2}$'s are minimal generators of $LT_{\leq_2}(I)$ by construction (all multiples of previous generators have been erased). By a dimension argument, no minimal generator is missing.

Once d is big enough for all minimal generators of $LT_{\leq_2}(I)$ to have been produced, no monomials can be left in L and the algorithm terminates. Termination and correctness are then clear.

As columns are labelled by some $x_i x^\alpha$ with $x^\alpha \in NS_{\leq_2}(I)$ then at most $n\delta$ columns are produced in the algorithm. As the rank of M is δ and so is also its number of rows, the column-reduction of a given column costs $O(\delta^2)$ arithmetic operations. Consequently, the total cost of the algorithm is in $O(n\delta^3)$ arithmetic operations. □

Remark 4.7. The previous algorithm remarkably bears the same asymptotic complexity as the vanilla classical FGLM algorithm ($O(n\delta^3)$ arithmetic operations), regardless of the more involved linear algebra part. Could fast linear algebra also be applied here?

Example 4.8. Let $(x + \frac{1}{2}y, y^2 + 1)$ be a GB of the ideal it spans, for $w = [0, -1]$ and grevlex. We compute a GB of the same ideal for $w = [0, 0]$ and grevlex. The following matrices are: the polynomials added to M (in three batches, by degree), the final state of M and the final P. In the end, we get $(y + 2x, x^2 + \frac{1}{4})$ as the output GB.

$$\begin{array}{c|cc} & 1 & x \\ \hline 1 & 1 & -2^{-2} \\ y & -2^{-1} & 1 \end{array}, \quad \begin{array}{c|cc} & y & x^2 \\ \hline 1 & 0 & 0 \\ y & 0 & 0 \end{array}, \quad P = \begin{array}{c|ccc} & 1 & x & y \\ \hline 1 & 1 & & 2^{-2} \\ x & & 1 & 2 \\ y & & & 1 \\ x^2 & & & 1 \end{array}$$

5 NUMERICAL DATA

A toy implementation of our algorithms in SageMath [Sage] is available on https://gist.github.com/TristanVaccon. The following arrays gather some numerical results. The timings are expressed in seconds of CPU time.[2]

5.1 Tropical to classical

For a given p, we take three polynomials with random coefficients in $\mathbb{Z}_p$ (using the Haar measure) in $\mathbb{Q}_p[x, y, z]$ of degrees $2 \leq d_1 \leq d_2 \leq d_3 \leq 4$. $D = d_1 + d_2 + d_3 - 2$ is the Macaulay bound. We first compute a tropical GB for the weight $w = [0, 0, 0]$ and the grevlex monomial ordering, and then apply Algorithms 2 and 4 to obtain a lex GB. We compare with the strategy of computing a classical grevlex GB and then applying FGLM to obtain a lex GB. For any given choice of d_i's, the experiment is repeated 50 times. Coefficients of the initial polynomials are given at high-enough precision $O(p^N)$ for no precision issue to appear (see [RV16] for more on FGLM at finite precision).

Coefficients of the output tropical GB or classical GB are known at individual precision $O(p^{N-m})$ (for some $m \in \mathbb{Z}$)). We compute the total mean and max on those m's on the obtained GB. In the first following array, we provide the mean and max for the tropical strategy. In the second, to compare classical and tropical, we provide couples for the mean on the 50 ratios of timing per execution (t), along with the arithmetic (Σ) and geometric (π) mean of the 50 ratios of mean loss in precision per execution. Data for $p = 101$ or 65519 are not worth for these ratios as the loss in precision is 0 most of the time.

In average the tropical strategy takes longer, but save a large amount of precision (for small p). While the ratio of saved precision may decrease with the degree, the abolute amount of saved precision is often still very large. We have also noted that the standard deviations for these ratios can be very large.

precision (trop.)	$D=4$		5		6		7		8		9	
$p=2$	11	103	25	278	60	509	176	1253	300	1783	652	3929
3	3	21	12	97	36	396	125	634	141	1002	282	2876
101	0	1	0	1	1	79	0	2	15	408	0	2
65519	0	0	0	0	0	0	0	0	0	0	0	0

trop. / classical	$D=4$			5			6			7			8			9		
	t	Σ	π	t	Σ	π	t	Σ	π	t	Σ	π	t	Σ	π	t	Σ	π
$p=2$	20	.4	.3	5	.4	.2	5	.5	.2	5	.6	.2	1.5	.8	.2	9	1	.2
3	6	.6	.2	6	.5	.2	5	.5	.2	2	.4	.1	1.2	.7	.1	.9	.9	.1

5.2 Tropical to tropical

We repeat the same experiments for mean and max loss in precision, but this time we compute a tropical GB for weight $w = [0, 0, 0]$ and then use Algorithm 4 to compute a tropical GB for weight $w = [-2, 4, -8]$ (grevlex for tie-breaks in both cases). Precision-wise, it seems that there is an intrinsic difficulty in computing a lex GB compared to a tropical GB.

precision loss	$D=4$		5		6		7		8		9	
$p=2$	2	18	2.5	14	2.6	14	2.9	16	3	17	3.5	19
3	1	9	1	7	1	9	1.4	14	1.4	11	2	13
101	0	1	0	1	0	1	0	2	0	2	0	2
65519	0	0	0	0	0	0	0	0	0	0	0	0

5.3 Semi-stability and shape position

We adapt our setting to $\mathbb{Q}((t))$, using entries with coefficients in $\mathbb{Z}[\![t]\!]$ given at precision 50 (using SageMath's built-in random function), and apply the ideas of Subsection 2.5 and Section 3. As $\mathbb{Q}$ is involved, computations are slow for $D \geq 7$ due to coefficients growth.

$w=[0,0,0]$+grevlex	$D=4$		5		6	
mean timing (F5 & FGLM)	2.8	9.4	3.9	102	10	1030
precision F5 (mean & max)	0	2	0	2	0	3
precision FGLM (mean & max)	0	0	0.1	8	0.4	34

[2] Everything was performed on a Ubuntu 16.04 with 2 processors of 2.6GHz and 16 GB of RAM.

REFERENCES

[CM19] Chan A., Maclagan D., Gröbner bases over fields with valuations, Math. Comp. 88 (2019), 467-483.

[FGHR14] Faugère, J.-C., Gaudry, P., Huot, L., Renault, G., Sub-cubic Change of Ordering for Gröbner Basis: A Probabilistic Approach, in Proceedings: ISSAC 2014. ACM, Kobe, Japon, pp. 170–177, 2014

[FGLM93] Faugère, J.-C., Gianni, P., Lazard, D., Mora, T., Efficient computation of zero-dimensional Gröbner bases by change of ordering, J. of Symbolic Computation 16 (4), 329–344, 1993

[GRZ19] Görlach, P, Ren, Y, Zhang, L., Computing zero-dimensional tropical varieties via projections, arXiv:1908.03486

[HH11] Herzog J., Hibi T., Monomial Ideals, Springer, 2001

[Huo13] Huot, L., Résolution de systèmes polynomiaux et cryptologie sur les courbes elliptiques, Ph.D. thesis, Université Pierre et Marie Curie (Paris VI), http://tel.archives-ouvertes.fr/tel-00925271

[JRS19] Jensen, A., Ren, Y., Schoenemann, H., The gfanlib interface in Singular and its applications, J. of Software for Algebra and Geometry 9 (2019), 81-87

[MS15] Maclagan, D. and Sturmfels, B., Introduction to tropical geometry, Graduate Studies in Mathematics, volume 161, AMS, Providence, RI, 2015

[MR19] Markwig, T. and Ren, Y., Computing Tropical Varieties Over Fields with Valuation, Foundations of Computational Mathematics, 2019

[RV16] Renault, G. and Vaccon, T. On the p-adic stability of the FGLM algorithm, arxiv:1602.00848

[Sage] SageMath, the Sage Mathematics Software System (Version 8.6), The Sage Development Team, 2018, http://www.sagemath.org

[Ser79] Serre, J.-P., Local fields, Vol. 67 of Graduate Texts in Mathematics. Springer-Verlag, New York-Berlin, translated from the French by Marvin Jay Greenberg

[Va15] Vaccon T., Matrix-F5 Algorithms and Tropical Gröbner Bases Computation, in Proceedings: ISSAC 2015, Bath, UK. Extended version in the J. of Symbolic Computation, Dec. 2017.

[VY17] Vaccon T., Yokoyama K., A Tropical F5 algorithm, in Proceedings: ISSAC 2017, Kaiserslautern, Germany.

[VVY18] Vaccon T., Verron T., Yokoyama K., On Affine Tropical F5 algorithm, in Proceedings: ISSAC 2018, New York, USA. Extended version to appear in the J. of Symbolic Computation.

Modular Techniques for Effective Localization and Double Ideal Quotient

Yuki Ishihara*
Graduate School of Science, Rikkyo University
Tokyo, Japan
yishihara@rikkyo.ac.jp

ABSTRACT

By double ideal quotient, we mean $(I : (I : J))$ where I and J are ideals. In our previous work [12], double ideal quotient and its variants are shown to be very useful for checking prime divisors and generating primary components. Combining those properties, we can compute "direct localization" effectively, comparing with full primary decomposition. In this paper, we apply modular techniques effectively to computation of such double ideal quotient and its variants, where first we compute them modulo several prime numbers and then lift them up over rational numbers by Chinese Remainder Theorem and rational reconstruction. As a new modular technique for double ideal quotient and its variants, we devise criteria for output from modular computations. Also, we apply modular techniques to intermediate primary decomposition. We examine the effectiveness of our modular techniques for several examples by preliminary computational experiments in Singular.

CCS CONCEPTS

• **Computing methodologies** → **Algebraic algorithms**.

KEYWORDS

Gröbner Basis, Primary Decomposition, Modular Method, Localization, Double Ideal Quotient

ACM Reference Format:
Yuki Ishihara. 2020. Modular Techniques for Effective Localization and Double Ideal Quotient. In *International Symposium on Symbolic and Algebraic Computation (ISSAC '20), July 20–23, 2020, Kalamata, Greece.* ACM, New York, NY, USA, 8 pages. https://doi.org/10.1145/3373207.3404017

1 NEW CONTRIBUTIONS

For proper ideals I and J, *double ideal quotient* is an ideal of shape $(I : (I : J))$. It and its variants are effective for localization and give us criteria for prime divisors (primary components) and ways to generate primary components. In [12], "Local Primary Algorithm" computes the specific primary component from given a prime ideal without full primary decomposition. However, they tend to be very time-consuming for computing Gröbner bases and ideal quotients in some cases. Also, there is another problem with a way to find candidates of prime divisors. As a solution of these problems, we propose a new method for computing double ideal quotient in the n variables polynomial ring with rational coefficients $\mathbb{Q}[X]$ by using "Modular Techniques", where $X = \{x_1, \ldots, x_n\}$. It is well-known that modular techniques are useful to avoid intermediate coefficient growth and have a good relationship with parallel computing (see [2, 5, 11, 15]). In this paper, we have the following contributions.

(1) Apply modular techniques to double ideal quotient. (Theorem 2.2.8 and Theorem 2.2.9)
(2) Extend criteria about prime divisor in [12]. (Theorem 2.1.4)
(3) Devise a new method for certain intermediate decomposition in some special cases. (Corollary 2.3.2, Proposition 2.3.4)

For a prime number p, let $\mathbb{Z}_{(p)} = \{a/b \in \mathbb{Q} \mid p \nmid b\}$ be the localized ring by p and $\mathbb{F}_p[X]$ the polynomial ring over the finite field. We denote by ϕ_p the canonical projection $\mathbb{Z}_{(p)}[X] \to \mathbb{F}_p[X]$. Given ideals I and J in the polynomial ring with rational coefficients $\mathbb{Q}[X]$, we first compute double ideal quotient of the image $\phi_p((I : (I : J)) \cap \mathbb{Z}_{(p)}[X])$ in $\mathbb{F}_p[X]$ for "lucky" primes p (we will discuss such luckiness later). Next, we lift them up to G_{can}, a candidate of Gröbner basis, from the computed Gröbner basis $\bar{G}$ of $\phi_p((I : (I : J)) \cap \mathbb{Z}_{(p)}[X])$ by using Chinese Remainder Theorem (CRT) and rational reconstruction (see [5]). Avoiding intermediate coefficient growth, this method is effective for several examples.

Also, we extend the criterion in [12] about prime divisor in order to compute certain "intermediate decomposition" of ideals and to find prime divisors in some special cases. For an ideal I and a prime ideal P, it follows that P is a prime divisor of I if and only if $P \supset (I : (I : P))$ (see Theorem 31 (Criterion 5), [12]). However, the projected image of a prime ideal may not be a prime ideal but an intersection of prime ideals in $\mathbb{F}_p[X]$. Thus, we generalize the criterion to a radical ideal $J \supset I$; it follows that every prime divisor P of J is associated with I if and only if $J \supset (I : (I : J))$. For such a radical ideal J, if J is unmixed, we can compute the intersection of primary components Q of I whose associated prime is a prime divisor of J by modular techniques. This ideal may be considered as an "intermediate component" of I. By gathering these intermediate components, we may obtain an "intermediate primary decomposition" (see Definition 2.3.1). For this computation, we can utilize maximal independent sets (see Section 2.3 for the definition of maximal independent set).

Primary decomposition of an ideal in a polynomial ring over a field is an essential tool of Commutative Algebra and Algebraic Geometry. Algorithms of primary decomposition have been much

ISSAC '20, July 20–23, 2020, Kalamata, Greece

ACM ISBN 978-1-4503-7100-1/20/07…$15.00
https://doi.org/10.1145/3373207.3404017

studied, for example, by [7, 9, 13, 19]. We apply double ideal quotient to check whether candidates of prime divisors from modular techniques are associated with the original ideal or not. It shall contribute the total efficiency of the whole process since our "intermediate decomposition" may divide a big task into small ones as a "divide-and-conquer" strategy.

This paper is organized as follows. In section 2.1, we introduce extended criteria for prime divisor and primary component based on double ideal quotient and its variants. In section 2.2, we apply modular techniques to double ideal quotient and its variants. In section 2.3, we sketch an outline of intermediate primary decomposition. In section 3, we see some effectiveness of modular method in several examples in a preliminary experiment. Its practicality will be examined by computing more detailed experiments.

2 MAIN THEOREMS

Here we show theoretical bases for our new techniques described in Section 1. We denote an arbitrary field by K and the ideal generated by $f_1, \ldots, f_s \in K[X]$ by $(f_1, \ldots, f_s)_{K[X]}$. If the base ring is obvious, we simply write $(f_1, \ldots, f_s)$. Also, we denote by $K[X]_P$ the localized ring by a prime ideal P and by I_P the ideal $IK[X]_P$ respectively. For an irredundant primary decomposition $\mathcal{Q}$ of I, we say that P is a prime divisor of I if there is a primary component $Q \in \mathcal{Q}$ s.t. $P = \sqrt{Q}$. For simplicity, we assume primary decomposition is irredundant. We denote the set of prime divisors of I by $\mathrm{Ass}(I)$ (see Definition 4.1.1 and Theorem 4.1.5 in [10]). For a prime ideal P, P_P is also prime.

2.1 Criteria for prime divisors and primary components

First, we recall criteria using double ideal quotient and its variants (see [12], Sect. 3.). In Proposition 2.1.1, the equivalence between (A) and (B) is originally described in [21]. In our previous work [12], we relate it with a variant of double ideal quotient $(I : (I : P^\infty))$. Double ideal quotient is also used to compute equidimensional hull in [7], which we use in Lemma 2.1.7 and Theorem 2.1.10 later. We will show that such double ideal quotient(s) can be computed efficiently by modular techniques in Section 2.2.

Proposition 2.1.1 ([12], Theorem 31). *Let I be an ideal and P a prime ideal. Then, the following conditions are equivalent.*
(A) $P \in \mathrm{Ass}(I)$,
(B) $P \supset (I : (I : P))$,
(C) $P \supset (I : (I : P^\infty))$.

Remark 2.1.2. *In Proposition 2.1.1, the condition $P \supset (I : (I : P))$ is equivalent to $P = (I : (I : P))$ since $P \subset (I : (I : P))$ always holds for any ideals I and P. Indeed, $P(I : P) \subset I$ from the definition of $(I : P)$ and thus $P \subset (I : (I : P))$.*

Remark 2.1.3. *The operations of double ideal quotient and localization by prime ideal are commutative. Indeed, for ideals I, J and a prime ideal P, $(I : J)_P = (I_P : J_P)$ from Corollary 3.15 in [1] and thus we obtain $(I : (I : J))_P = (I_P : (I : J)_P) = (I_P : (I_P : J_P))$. Similarly, we have $(I : (I : J^\infty))_P = (I_P : (I : J^\infty)_P) = (I_P : (I_P : J_P^\infty))$ as $(I : J^\infty) = (I : J^m)$ and $(I_P : J_P^\infty) = (I_P : J_P^m)$ for a sufficiently large integer m. Also, a prime ideal P is associated with an ideal I if and only if P_P is associated with I_P since there is a correspondence between primary decompositions of I and I_P (see Proposition 4.9 in [1]). Similarly, for a P-primary ideal Q, Q is a P-primary component of I if and only if Q_P is a P_P-primary component of I_P.*

Next, we introduce extended theorems about double ideal quotient and its variants toward intermediate primary decomposition in Section 2.3. Proposition 2.1.1 gives a relationship between an ideal I and a prime divisor P. It can be extended to one between an ideal I and an intersection of some prime divisors J. Thus, we consider a radical ideal J instead of a prime ideal P as follows.

Theorem 2.1.4. *Let I be an ideal and J a proper radical ideal. Then, the following conditions are equivalent.*
(A) $\mathrm{Ass}(J) \subset \mathrm{Ass}(I)$,
(B) $J \supset (I : (I : J))$,
(C) $J \supset (I : (I : J^\infty))$.

Proof. First, we show that (A) implies (B). Let $P \in \mathrm{Ass}(J) \subset \mathrm{Ass}(I)$. Then, $P \supset (I : (I : P))$ by Proposition 2.1.1. Thus, $P \supset (I : (I : P)) \supset (I : (I : J))$. Since $J = \bigcap_{P \in \mathrm{Ass}(J)} P$, we obtain $J \supset (I : (I : J))$. Next, we show that (B) implies (C). As $(I : J) \subset (I : J^\infty)$, we obtain $J \supset (I : (I : J)) \supset (I : (I : J^\infty))$. Finally, we show that (C) implies (A). Let $P \in \mathrm{Ass}(J)$. Then, $J_P \supset (I : (I : J^\infty))_P = (I_P : (I_P : J_P^\infty))$ from Remark 2.1.3 and thus $J_P = P_P \in \mathrm{Ass}(I_P)$ from Proposition 2.1.1. Hence, $P \in \mathrm{Ass}(I)$ by Remark 2.1.3. □

Example 2.1.5. *Let $I = (x) \cap (x^3, y) \cap (x^2 + 1)$ and $J = (x, y) \cap (x^2 + 1)$. Then, $(I : (I : J)) = (x, y) \cap (x^2 + 1) = J$ and $\mathrm{Ass}(J) = \{(x, y), (x^2 + 1)\} \subset \mathrm{Ass}(I) = \{(x), (x, y), (x^2 + 1)\}$. In addition, we have $(I : (I : J^\infty)) = (x^2, y) \cap (x^2 + 1) \subset J$.*

To generate primary component, the following lemma is well-known. Here, for a d-dimensional ideal I, equidimensional hull $\mathrm{hull}(I)$ is the intersection of its d-dimensional primary components.

Lemma 2.1.6 ([7], Section 4. [14], Remark 10). *Let I be an ideal and P a prime divisor of I. For a sufficiently large integer m, $\mathrm{hull}(I + P^m)$ is a P-primary component appearing in a primary decomposition of I.*

Here, we generalize Lemma 2.1.6 to an intersection of equidimensional prime divisors as follows.

Lemma 2.1.7. *Let I be an ideal and J an intersection of prime divisors of I. Suppose J is unmixed i.e. $\dim(P) = \dim(J)$ for any $P \in \mathrm{Ass}(J)$. Then, for a sufficiently large integer m, $\mathrm{hull}(I + J^m)$ is an intersection of primary components appearing in a primary decomposition of I i.e. $\mathrm{hull}(I + J^m) = \bigcap_{P \in \mathrm{Ass}(J)} Q(P)$ where $Q(P)$ is a P-primary component of I.*

Proof. Let m be a positive integer. First, we note that, for each $P \in \mathrm{Ass}(J)$, $I \subset \mathrm{hull}(I + J^m)_P \cap K[X] \subset \mathrm{hull}(I + P^m)$ since

$$I \subset I + J^m \subset \mathrm{hull}(I + J^m) \subset \mathrm{hull}(I + J^m)_P \cap K[X]$$
$$\subset \mathrm{hull}(I + P^m)_P \cap K[X] = \mathrm{hull}(I + P^m)$$

where the last equality comes from the fact that $\sqrt{I + P^m} = P$ and P is the unique isolated prime divisor of $I + P^m$. By Lemma 2.1.6, there exist a sufficiently large integer $m(P)$ and a primary decomposition $\mathcal{Q}$ of I such that $\mathrm{hull}(I + P^{m(P)}) \in \mathcal{Q}$. Then,

$$I \subset \bigcap_{P \in \mathrm{Ass}(J)} \mathrm{hull}(I + J^{m(P)})_P \cap K[X] \subset \bigcap_{P \in \mathrm{Ass}(J)} \mathrm{hull}(I + P^{m(P)})$$

and, by intersecting $\bigcap_{Q\in\mathcal{Q},\sqrt{Q}\notin\mathrm{Ass}(J)} Q$ with them, we obtain

$$I \subset \left(\bigcap_{P\in\mathrm{Ass}(J)} \mathrm{hull}(I+J^{m(P)})_P \cap K[X]\right) \cap \bigcap_{Q\in\mathcal{Q},\sqrt{Q}\notin\mathrm{Ass}(J)} Q$$
$$\subset \left(\left(\bigcap_{P\in\mathrm{Ass}(J)} \mathrm{hull}(I+P^{m(P)})\right) \cap \bigcap_{Q\in\mathcal{Q},\sqrt{Q}\notin\mathrm{Ass}(J)} Q = I.\right.$$

Thus, $\left(\bigcap_{P\in\mathrm{Ass}(J)} \mathrm{hull}(I+J^{m(P)})_P \cap K[X]\right) \cap \bigcap_{Q\in\mathcal{Q},\sqrt{Q}\notin\mathrm{Ass}(J)} Q = I$ and $\mathrm{hull}(I+J^{m(P)})_P \cap K[X]$ is a P-primary component of I. Since J is unmixed, $\sqrt{I+J^m} = \sqrt{J} = \bigcap_{P\in\mathrm{Ass}(J)} P$ and $\mathrm{Ass}(\mathrm{hull}(I+J^m)) = \mathrm{Ass}(J)$ i.e. $\mathrm{hull}(I+J^m) = \bigcap_{P\in\mathrm{Ass}(J)} \mathrm{hull}(I+J^m)_P \cap K[X]$. Thus, for $m \geq \max\{m(P) \mid P \in \mathrm{Ass}(J)\}$, $\mathrm{hull}(I+J^m)$ is an intersection of primary components of a primary decomposition of I. □

Using variants of double ideal quotient, we devise a criterion for primary component and generate isolated primary components. We remark that Theorem 2.1.8 holds for any Noetherian rings.

Theorem 2.1.8 ([12], Theorem 26 (Criterion 1)). *Let I be an ideal and P a prime divisor of I. For a P-primary ideal Q, assume $Q \not\supset (I : P^\infty)$ and let $J = (I : P^\infty) \cap Q$. Then, the following conditions are equivalent.*

(A) *Q is a P-primary component for some primary decomposition of I.*
(B) *$(I : (I : J)^\infty) = J$.*

We also generalize Theorem 2.1.8 to intersection of primary components as follows. We can check whether m appearing in Lemma 2.1.7 is large enough or not by Theorem 2.1.9.

Theorem 2.1.9. *Let I be an ideal and J an intersection of prime divisors of I. Suppose J is unmixed. For an unmixed ideal L with $\sqrt{L} = J$, assume $\sqrt{(L : (I : J^\infty))} = J$ and let $Z = (I : J^\infty) \cap L$. Then, the following conditions are equivalent.*

(A) *$L = \bigcap_{P\in\mathrm{Ass}(J)} Q(P)$ where $Q(P)$ is a P-primary components of I.*
(B) *$(I : (I : Z)^\infty) = Z$.*

Proof. First, we show (A) implies (B). From (A), it is easy to see that $\mathcal{T} = \mathrm{Ass}((I : J^\infty)) \cup \mathrm{Ass}(L)$ is *an isolated set* (see Definition 5 in [12]). Indeed, for $P' \in \mathrm{Ass}(I)$, if there exists $P \in \mathcal{T}$ s.t. $P' \subset P$, then $P' \in \mathcal{T}$ since $\mathrm{Ass}((I : J^\infty)) = \{P' \in \mathrm{Ass}(I) \mid J \not\subset P'\}$ and $\mathrm{Ass}(L) = \mathrm{Ass}(J)$. Thus, for $S = K[X] \setminus (\bigcup_{P\in\mathcal{T}} P)$, we obtain $Z = IK[X]_S \cap K[X]$ from Lemma 6 in [12] and $\mathcal{T} = \mathrm{Ass}(Z)$. By Lemma 25 in [12], we obtain $(I : (I : Z)^\infty) = Z$.

Second, we show (B) implies (A). Let $P \in \mathrm{Ass}(J)$. Then, we obtain $P_P = J_P$ and $\sqrt{L_P} = (\sqrt{L})_P = J_P = P_P$. Thus, L_P is a P_P-primary ideal and $Z_P = (I : J^\infty)_P \cap L_P = (I_P : J_P^\infty) \cap L_P$. Since $\sqrt{(L : (I : J^\infty))} = J$, $\sqrt{(L_P : (I_P : J_P^\infty))} = P_P$ and thus $L_P \not\supset (I_P : J_P^\infty)$; otherwise we get $\sqrt{(L_P : (I_P : J_P^\infty))} = K[X]_P \neq P_P$. Here, $(I_P : (I_P : Z_P)^\infty) = Z_P$ for all $P \in \mathrm{Ass}(J)$ since $(I : (I : Z)^\infty) = Z$ and $(I_P : (I_P : Z_P)^\infty) = (I : (I : Z)^\infty)_P$. Thus, by Theorem 2.1.8, L_P is a primary component of I_P. Since L is unmixed and $L = \sqrt{J}$, it follows that $L = \bigcap_{P\in\mathrm{Ass}(J)} L_P \cap K[X]$. From Remark 2.1.3, $L_P \cap K[X]$ is a P-primary component of I if and only if L_P is a P_P-primary component of I_P. Finally, we obtain the equivalence. □

Also, we can compute the isolated primary component from its associated prime by a variant of double ideal quotient.

Theorem 2.1.10 ([12], Theorem 36). *Let I be an ideal and P an isolated prime divisor of I. Then*

$$\mathrm{hull}((I : (I : P^\infty)^\infty))$$

is the isolated P-primary component of I.

We generalize Theorem 2.1.10 as follows.

Theorem 2.1.11. *Let I be an ideal and J an intersection of isolated prime divisors of I. Suppose J is unmixed. Then*

$$\mathrm{hull}((I : (I : J^\infty)^\infty)) = \bigcap_{P\in\mathrm{Ass}(J)} Q(P)$$

where $Q(P)$ is the isolated P-primary component of I.

Proof. Let $\mathcal{Q}$ be a primary decomposition of I. By Proposition 22 in [12], we obtain

$$(I : (I : J^\infty)^\infty) = \bigcap_{Q\in\mathcal{Q}, J\subset\sqrt{IK[X]_{\sqrt{Q}}\cap K[X]}} Q.$$

Since $J \subset \sqrt{IK[X]_{\sqrt{Q(P)}} \cap K[X]} = \sqrt{Q(P)} = P$ for $P \in \mathrm{Ass}(J)$, it follows that

$$(I : (I : J^\infty)^\infty) = \bigcap_{P\in\mathrm{Ass}(J)} Q(P) \cap \bigcap_{Q\in\mathcal{Q}, J\subset\sqrt{IK[X]_{\sqrt{Q}}\cap K[X]}, \sqrt{Q}\notin\mathrm{Ass}(J)} Q.$$

As J is unmixed, each $Q(P)$ has the same dimension for $P \in \mathrm{Ass}(J)$. Then, $\dim(\bigcap_{Q\in\mathcal{Q}, J\subset\sqrt{IK[X]_{\sqrt{Q}}\cap K[X]}, \sqrt{Q}\notin\mathrm{Ass}(J)} Q) < \dim(J)$ from the fact that for $Q \in \mathcal{Q}$ with $J \subset \sqrt{IK[X]_{\sqrt{Q}} \cap K[X]}$ and $\sqrt{Q} \notin \mathrm{Ass}(J)$, there exists $P \in \mathrm{Ass}(J)$ s.t. $P \subsetneq \sqrt{Q}$. Since J is an intersection of isolated prime divisors of I, we obtain

$$\mathrm{hull}((I : (I : J^\infty)^\infty)) = \bigcap_{P\in\mathrm{Ass}(J)} Q(P).$$

□

2.2 Modular techniques for double ideal quotient

We propose modular techniques for double ideal quotient. For a prime number p, let $\mathbb{Z}_{(p)} = \{a/b \in \mathbb{Q} \mid p \nmid b\}$ be the localized ring by p and $\mathbb{F}_p[X]$ the polynomial ring over the finite field. We denote by ϕ_p the canonical projection $\mathbb{Z}_{(p)}[X] \to \mathbb{F}_p[X]$. For $F \subset \mathbb{Q}[X]$, we denote by $I(F)$ the ideal generated by F. For $F \subset \mathbb{Z}_{(p)}[X]$, we denote $\langle \phi_p(F) \rangle$ by $I_p(F)$ and $\phi_p(I(F) \cap \mathbb{Z}_{(p)}[X])$ by $I_p^0(F)$ respectively.

We recall the outline of "modular algorithm for ideal operation" (see [16]) as Algorithm 1. Given ideals I, J, ideal operations $AL(*,*)$ over $\mathbb{Q}[X]$ and $AL_p(*,*)$ over $\mathbb{F}_p[X]$ as inputs, we compute $AL(I,J)$ as the output by using modular computations. First, we choose a list of random prime numbers $\mathcal{P}$, which satisfies certain computable condition primeTest. For example, primeTest is to check whether p is permissible (see Definition 2.2.1) for Gröbner bases of I and J or not. Next, we compute modular operations $H_p = AL_p(I,J)$ for each $p \in \mathcal{P}$. After omitting expected unlucky primes by deleteUnluckyPrimes, we lift H_p's up to H_{can} by CRT

and rational reconstruction. Finally, we check H_{can} is really the correct answer by finalTest. If finalTest says False, then we enlarge $\mathcal{P}$ and continue from the first step. In this paper, we introduce new finalTest for ideal quotient and double ideal quotient. We remark that the termination of this modular algorithm is ensured by the finiteness of unlucky prime numbers. For example, for a given ideals I, J and an algorithm for the ideal quotient $(I : J)$ over the rational numbers, there are only finite many steps from the inputs to the outputs and thus the number of coefficients is also finite; hence we can project the computations onto those over finite fields $\mathbb{F}_p$ for all prime numbers p except those appearing in coefficients (see Lemma 6.1 in [16] for details).

Algorithm 1 Modular Algorithm for Ideal Operation

Input: I, J: ideals, $AL(*,*)$: an ideal operation over $\mathbb{Q}[X]$, $AL_p(*,*)$: an ideal operation over $\mathbb{F}_p[X]$,
Output: $AL(I, J)$ over $\mathbb{Q}[X]$

```
choose 𝒫 as a list of random primes satisfying primeTest;
ℋ𝒫 = ∅;
while do
  for p ∈ 𝒫 do
    compute H_p = AL_p(I, J);
    ℋ𝒫 = ℋ𝒫 ∪ {H_p};
  end for
  (ℋ𝒫_lucky, 𝒫_lucky) = deleteUnluckyPrimes(ℋ𝒫, 𝒫);
  lift ℋ𝒫_lucky to H_can by CRT and rational reconstruction;
  if H_can passes finalTest then
    return H_can
  end if
  enlarge 𝒫 with prime numbers not used so far;
end while
```

First, we introduce some notions of *good* primes as follows.

Definition 2.2.1 ([16], Definition 2.1). *Let p be a prime number, $F \subset \mathbb{Q}[X]$ and $\prec$ a monomial ordering. Let G be the reduced Gröbner basis of $I(F)$ with respect to $\prec$. Here, we denote by $\mathrm{lc}_\prec(f)$ the leading coefficient of a polynomial f with respect to $\prec$.*

(1) *p is said to be* weak permissible *for F, if $F \subset \mathbb{Z}_{(p)}[X]$.*
(2) *p is said to be* permissible *for F and $\prec$, if p is weak permissible for $F \subset \mathbb{Q}[X]$ and $\phi_p(\mathrm{lc}_\prec(f)) \neq 0$ for all f in F.*
(3) *p is said to be* compatible *with F if p is weak permissible for F and $I_p^0(F) = I_p(F)$.*
(4) *p is said to be* effectively lucky *for F and $\prec$, if p is permissible for $(G, \prec)$ and $\phi_p(G)$ is the reduced Gröbner basis of $I_p(G)$.*

Remark 2.2.2. *If p is effectively lucky for F and $\prec$, then p is compatible with F (see Lemma 3.1 (3) in [16]).*

Next, the notion of p-compatible Gröbner basis candidate is very useful for easily computable tests toward finalTest in modular techniques .

Definition 2.2.3 ([16], Definition 4.1). *Let G_{can} be a finite subset of $\mathbb{Q}[X]$ and $F \subset \mathbb{Q}[X]$. We call G_{can}* a p-compatible Gröbner basis candidate *for F and $\prec$, if p is permissible for G_{can} and $\phi_p(G_{can})$ is a Gröbner basis of $I_p^0(F)$ with respect to $\prec$.*

The following can be used to finalTest in modular techniques.

Lemma 2.2.4 ([16], Proposition 4.1). *Suppose that G_{can} is a p-compatible Gröbner basis candidate for $(F, \prec)$, and $G_{can} \subset I(F)$. Then G_{can} is a Gröbner basis of $I(F)$ with respect to $\prec$.*

We introduce the following easily computable tests for ideal quotient and saturation in modular techniques, appearing in [16].

Lemma 2.2.5 ([16], Lemma 6.2 and Lemma 6.4). *Suppose that a prime number p is compatible with $(F, \prec)$ and permissible for $(f, \prec)$. For a finite subset $H_{can} \subset \mathbb{Q}[X]$, H_{can} is a Gröbner basis of $(I(F) : f)$ with respect to $\prec$, if the following conditions hold;*

(1) *p is permissible for $(H_{can}, \prec)$,*
(2) *$\phi_p(H_{can})$ is a Gröbner basis of $(I_p(F) : \phi_p(f))$ with respect to $\prec$,*
(3) *$H_{can} \subset (I(F) : f)$.*

For a finite subset $L_{can} \subset \mathbb{Q}[X]$, L_{can} is a Gröbner basis of $(I(F) : f^\infty)$ with respect to $\prec$, if the following conditions hold;

(1) *p is permissible for $(L_{can}, \prec)$,*
(2) *$\phi_p(L_{can})$ is a Gröbner basis of $(I_p(F) : \phi_p(f)^\infty)$ with respect to $\prec$,*
(3) *$L_{can} \subset (I(F) : f^\infty)$.*

We generalize Lemma 2.2.5 by replacing f into an ideal J as follows. We recall that $I_p(G) = \langle \phi_p(G) \rangle_{\mathbb{F}_p[X]}$ where p is weak permissible for G.

Lemma 2.2.6. *Suppose that a prime number p is compatible with $(F, \prec)$ and permissible for $(G, \prec)$. For a finite subset $H_{can} \subset \mathbb{Q}[X]$, H_{can} is a Gröbner basis of $(I(F) : I(G))$ with respect to $\prec$, if the following conditions hold;*

(1) *p is permissible for $(H_{can}, \prec)$,*
(2) *$\phi_p(H_{can})$ is a Gröbner basis of $(I_p(F) : I_p(G))$ with respect to $\prec$,*
(3) *$H_{can} \subset (I(F) : I(G))$.*

Proof. Since p is permissible for $(H_{can}, \prec)$, we can consider $I_p(H_{can}) = \langle \phi_p(H_{can}) \rangle$. It is enough to show $I_p(H_{can}) = \phi_p((I(F) : I(G)) \cap \mathbb{Z}_{(p)}[X])$ since the equation implies H_{can} is a p-compatible Gröbner basis candidate for $(I(F) : I(G))$ with respect to $\prec$ and a Gröbner basis of $(I(F) : I(G))$ with respect to $\prec$ from $H_{can} \subset (I(F) : I(G))$ and Lemma 2.2.4.

It is clear that $I_p(H_{can}) \subset \phi_p((I(F) : I(G)) \cap \mathbb{Z}_{(p)}[X])$ as $H_{can} \subset (I(F) : I(G))$. To show the inverse inclusion, we pick $h \in (I(F) : I(G)) \cap \mathbb{Z}_{(p)}[X]$. Then, $hG \subset I(F) \cap \mathbb{Z}_{(p)}[X]$ where $hG = \{hg \mid g \in G\}$ since p is permissible for h and G. Thus,

$$\begin{aligned} \phi_p(h) I_p(G) = \phi_p(h) \langle \phi_p(G) \rangle &= \langle \phi_p(hG) \rangle \\ &\subset \langle \phi_p(I(F) \cap \mathbb{Z}_{(p)}[X]) \rangle = I_p^0(F) = I_p(F) \end{aligned}$$

by the compatibility of F; we obtain $\phi_p(h) \in (I_p(F) : I_p(G)) = I_p(H_{can})$. Hence $I_p(H_{can}) \supset \phi_p((I(F) : I(G)) \cap \mathbb{Z}_{(p)}[X])$. □

Remark 2.2.7. *We can check whether $H_{can} \subset (I(F) : I(G))$ or not, by checking whether $I(H_{can})I(G) \subset I(F)$ or not.*

We apply this lemma to double ideal quotient as follows.

Theorem 2.2.8. *Suppose that a prime number p is compatible with $(F, \prec)$ and permissible for $(G, \prec)$. Assume p satisfies $(I_p(F) : I_p(G)) = \phi_p((I(F) : I(G)) \cap \mathbb{Z}_{(p)}[X])$. For a finite subset $K_{can} \subset$*

$\mathbb{Q}[X]$, *K_{can} is a Gröbner basis of* $(I(F) : (I(F) : I(G)))$ *with respect to* $\prec$ *if the following conditions hold;*

(1) *p is permissible for* $(K_{can}, \prec)$,
(2) *$\phi_p(K_{can})$ is a Gröbner basis of* $(I_p(F) : (I_p(F) : I_p(G)))$ *with respect to* $\prec$,
(3) $K_{can} \subset (I(F) : (I(F) : I(G)))$.

Proof. Since p is permissible for $(K_{can}, \prec)$, we can consider $I_p(K_{can}) = \langle \phi_p(K_{can}) \rangle$. By Lemma 2.2.4, it is enough to show that K_{can} is a p-compatible Gröbner basis candidate of $(I(F) : (I(F) : I(G)))$. Since $K_{can} \subset (I(F) : (I(F) : I(G)))$, $I_p(K_{can}) \subset \phi_p((I(F) : (I(F) : I(G))) \cap \mathbb{Z}_{(p)}[X])$ holds. Thus, we show the other inclusion. Let $h \in (I(F) : (I(F) : I(G))) \cap \mathbb{Z}_{(p)}[X]$. Then,

$$\phi_p(h)\phi_p((I(F) : I(G)) \cap \mathbb{Z}_{(p)}[X]) \subset \phi_p(I(F) \cap \mathbb{Z}_{(p)}[X]) = I_p^0(F) = I_p(F).$$

Since $\phi_p((I(F) : I(G)) \cap \mathbb{Z}_{(p)}[X]) = (I_p(F) : I_p(G))$, we obtain $\phi_p(h) \in (I_p(F) : (I_p(F) : I_p(G))) = I_p(K_{can})$. Hence, $I_p(K_{can}) \supset \phi_p((I(F) : (I(F) : I(G))) \cap \mathbb{Z}_{(p)}[X])$. □

To check the conditions $(I_p(F) : I_p(G)) = \phi_p((I(F) : I(G)) \cap Z_{(p)}[X])$ and $K_{can} \subset (I(F) : (I(F) : I(G)))$, we need a Gröbner basis H of $(I(F) : I(G))$ in general (the former by $I_p(H) = (I_p(F) : I_p(G))$ and the latter by $I(K_{can})I(H) \subset I(F)$, respectively). However, as to the latter, in a special case that P is an associated prime divisor of I, we confirm it more easily. Setting $I(G) = P$ for a prime ideal P, we devise the following "Associated Test" using modular techniques.

Theorem 2.2.9 (Associated Test). *Let I be an ideal and P a prime ideal. Let F and G be Gröbner bases of I and P respectively. Suppose p is permissible for F, G and satisfies $(I_p(F) : I_p(G)) = \phi_p((I(F) : I(G)) \cap \mathbb{Z}_{(p)}[X])$. Let K_{can} be a finite subset of $\mathbb{Q}[X]$. Then, P is a prime divisor of I if the following conditions hold;*

(1) *p is permissible for* $(K_{can}, \prec)$,
(2) *$\phi_p(K_{can})$ is a Gröbner basis of* $(I_p(F) : (I_p(F) : I_p(G)))$ *with respect to* $\prec$,
(3) $(I_p(F) : (I_p(F) : I_p(G))) = I_p(G)$,
(4) $K_{can} \subset P$.

Proof. To prove this, we use Theorem 2.2.8. If all conditions of Theorem 2.2.8 hold, then K_{can} is a Gröbner basis of $(I : (I : P))$ and thus $(I : (I : P)) \subset P$ by the condition $K_{can} \subset P$; hence, P is a prime divisor of I by Proposition 2.1.1. Now, we show that all conditions of Theorem 2.2.8 hold. Since we have directly (1) and (2) in Theorem 2.2.8, it is enough to check the condition $K_{can} \subset (I(F) : (I(F) : I(G)))$. Indeed, we obtain $K_{can} \subset P \subset (I(F) : (I(F) : I(G)))$ by Remark 2.1.2 and (4). □

In above associated test, K_{can} will be G if P is a prime divisor of I. Thus, we omit CRT and rational reconstruction as follows. Also, we minimize the number of prime numbers we use since we can check the number is large enough comparing with the following $\|G\|$. For a finite set G of $\mathbb{Q}[X]$, we define

$$\|G\| = \max\{a^2+b^2 \mid \frac{a}{b} \text{ is a coefficient in a term of an element of } G\}.$$

Corollary 2.2.10 (Associated Test without CRT, Algorithm 2). *Let I be an ideal and P a prime ideal. Let F and G be Gröbner bases of I and P respectively. Let $\mathcal{P}$ be a finite set of prime numbers. Suppose every $p \in \mathcal{P}$ is permissible for F, G and satisfies $(I_p(F) : I_p(G)) = \phi_p((I(F) : I(G)) \cap \mathbb{Z}_{(p)}[X])$. Then, P is a prime divisor of I if the following conditions hold;*

(1) *$(I_p(F) : (I_p(F) : I_p(G))) = I_p(G)$ for every $p \in \mathcal{P}$,*
(2) *$\prod_{p \in \mathcal{P}} p$ is larger than $\|G\|$.*

Proof. Since $\prod_{p \in \mathcal{P}} p$ is larger than coefficients appearing in G for the rational reconstruction (see Lemma 4.2. in [5]), G is a Gröbner basis candidate itself and we can set $K_{can} = G$ in Theorem 2.2.9. Then, K_{can} satisfies all conditions of the theorem. □

Algorithm 2 Associated Test without CRT

Input: F: a Gröbner basis of an ideal I, G: a Gröbner basis of a prime ideal P, H: a Gröbner basis of $(I(F) : I(G))$.
Output: True if P is a prime divisor of I

```
choose P as a list of random primes satisfying primeTest (p ∈ P
is permissible for F, G and H) and ∏_{p∈P} p > ‖G‖;
RESTART;
while do
  for p ∈ P do
    if (I_p(F) : (I_p(F) : I_p(G))) ≠ I_p(G) then
      delete p from P;
    end if
  end for
  if ∏_{p∈P} p ≤ ‖G‖ then
    enlarge P with prime numbers not used so far and go back
    to RESTART;
  end if
  if (I_p(F) : I_p(G)) = I_p(H) for every p ∈ P then
    return True
  end if
  enlarge P with prime numbers not used so far and go back to
  RESTART;
end while
```

Also, we devise a non-associated test as follows. The test is useful since it does not need a condition $(I_p(F) : I_p(G)) = \phi_p((I(F) : I(G)) \cap \mathbb{Z}_{(p)}[X])$.

Theorem 2.2.11 (Non-Associated Test, Algorithm 3). *Let I be an ideal and P a prime ideal. Let F and G be Gröbner bases of I and P respectively. Suppose p is permissible for F and G. Let $K_{can} \subset \mathbb{Q}[X]$ and we assume p is permissible for K_{can}. Then, P is not a prime divisor of I if the following conditions hold;*

(1) *$\phi_p(K_{can})$ is a Gröbner basis of* $(I_p(F) : (I_p(F) : I_p(G)))$ *with respect to* $\prec$,
(2) $K_{can} \subset (I : (I : P))$,
(3) $(I_p(F) : (I_p(F) : I_p(G))) \neq I_p(G)$.

Proof. Suppose P is a prime divisor of I. Then, $(I : (I : P)) = P$ from Remark 2.1.2 and

$$\begin{aligned}\phi_p(K_{can}) &\subset \phi_p((I : (I : P)) \cap \mathbb{Z}_{(p)}[X]) \\ &\subset \phi_p(P \cap \mathbb{Z}_{(p)}[X]) = I_p^0(G) = I_p(G).\end{aligned}$$

Since $\langle \phi_p(K_{can}) \rangle = (I_p(F) : (I_p(F) : I_p(G))) \supset I_p(G)$, we obtain $(I_p(F) : (I_p(F) : I_p(G)) = I_p(G)$. This contradicts $(I_p(F) : (I_p(F) : I_p(G))) \neq I_p(G)$. □

Algorithm 3 Non-Associated Test

Input: F: a Gröbner basis of an ideal I, G: a Gröbner basis of a prime ideal P, H: a Gröbner basis of $(I(F) : I(G))$.
Output: False if P is NOT a prime divisor of I

```
choose 𝒫 as a list of random primes satisfying primeTest;
𝒦𝒫 = ∅;
while do
  for p ∈ 𝒫 do
    compute K_p = (I_p(F) : (I_p(F) : I_p(G)));
    if (I_p(F) : (I_p(F) : I_p(G))) = I_p(G) then
      delete p from 𝒫;
    else
      𝒦𝒫 = 𝒦𝒫 ∪ {K_p};
    end if
  end for
  (𝒦𝒫_lucky, 𝒫_lucky) = deleteUnluckyPrimes(𝒦𝒫, 𝒫);
  lift 𝒦𝒫_lucky to K_can by CRT and rational reconstruction;
  if I(K_can)I(H) ⊂ I then
    return False
  end if
  enlarge 𝒫 with prime numbers not used so far;
end while
```

Next, we consider modular saturation. Since $(I : J^m) = (I : J^\infty)$ for a sufficiently large m, the following holds from Lemma 2.2.6.

Lemma 2.2.12. *Suppose that a prime number p is compatible with $(F, \prec)$ and permissible for $(G, \prec)$. For a finite subset $H_{can} \subset \mathbb{Q}[X]$, H_{can} is a Gröbner basis of $(I(F) : I(G)^\infty)$ with respect to $\prec$, if the following conditions hold;*

1. *p is permissible for $(H_{can}, \prec)$,*
2. *$\phi_p(H_{can})$ is a Gröbner basis of $(I_p(F) : I_p(G)^\infty)$ with respect to $\prec$,*
3. *$H_{can} \subset (I(F) : I(G)^\infty)$.*

To check $H_{can} \subset (I(F) : I(G)^\infty)$, we can use the following.

Lemma 2.2.13. *Let H_{can}, F and G be finite sets of $K[X]$. For $G = \{f_1, \ldots, f_k\}$ and a positive integer m, we denote $\{f_1^m, \ldots, f_k^m\}$ by $G^{[m]}$. Then, the following conditions are equivalent.*

(A) $H_{can} \subset (I(F) : I(G)^\infty)$,
(B) $I(H_{can})I(G)^m \subset I(F)$ for some m,
(C) $I(H_{can})I(G^{[m]}) \subset I(F)$ for some m.

Proof. $[(A) \Rightarrow (B)]$ This is obvious from the definition of $(I(F) : I(G)^\infty)$. $[(B) \Rightarrow (C)]$ Since $I(G^{[m]}) \subset I(G)^m$, $I(H_{can})I(G^{[m]}) \subset I(H_{can})I(G)^m \subset I(F)$. $[(C) \Rightarrow (A)]$ As $I(G)^{km} \subset I(G^{[m]})$, we obtain $I(H_{can})I(G)^{km} \subset I(H_{can})I(G^{[m]}) \subset I(F)$ and $H_{can} \subset (I(F) : I(G)^\infty)$. □

Since the number of generators of $I(G^{[m]})$ is less than that of $I(G)^m$, it is better to check whether $I(H_{can})I(G^{[m]}) \subset I(F)$ or not.

Finally, we introduce modular techniques for double saturation.

Theorem 2.2.14. *Suppose that a prime number p is compatible with $(F, \prec)$ and permissible for $(G, \prec)$. Assume p satisfies $(I_p(F) : I_p(G)^\infty) = \phi_p((I(F) : I(G)^\infty) \cap \mathbb{Z}_{(p)}[X])$. For a finite subset $K_{can} \subset \mathbb{Q}[X]$, K_{can} is a Gröbner basis of $(I(F) : (I(F) : I(G)^\infty)^\infty)$ with respect to $\prec$ if the following conditions hold;*

1. *p is permissible for $(K_{can}, \prec)$,*
2. *$\phi_p(K_{can})$ is a Gröbner basis of $(I_p(F) : (I_p(F) : I_p(G)^\infty)^\infty)$ with respect to $\prec$,*
3. *$K_{can} \subset (I(F) : (I(F) : I(G)^\infty)^\infty)$.*

Proof. For a sufficiently large integer m, $(I(F) : I(G)^\infty) = (I(F) : I(G)^m)$ and $(I_p(F) : I_p(G)^\infty) = (I_p(F) : I_p(G)^m)$. Thus, we can prove this by the similar way of Theorem 2.2.8. □

2.3 Intermediate primary decomposition

In this section, we introduce intermediate primary decomposition as a bi-product of modular localizations devised in Section 2.2. We give a rough outline of possible "intermediate primary decomposition via MIS". In general, modular primary decomposition is very difficult to compute since primary component may be different over infinite many finite fields. For example, $I = (x^2+1) \cap (x+1)$ is a primary decomposition in $\mathbb{Q}[X]$, however, it is not one in $\mathbb{F}_p[X]$ for every prime number p of type $p = 4n + 1$. Thus, we propose *intermediate primary decomposition via MIS* instead of full primary decomposition. For a subset of variables X and an ideal I, we call U a maximal independent set (MIS) of I if $K[U] \cap I = \{0\}$ (see Definition 3.5.3 in [10]). Then, for a subset $U \subset X$, we define

$$\mathrm{Ass}_U(I_p(F)) = \{\bar{P}_p \in \mathrm{Ass}(I_p(F)) \mid U \text{ is a MIS of } \bar{P}_p\}.$$

where p is permissible for F. Also, we denote the set of prime divisors of I which have the same MIS U by

$$\mathrm{Ass}_U(I) = \{P \in \mathrm{Ass}(I) \mid U \text{ is a MIS of } P\}.$$

We note that U is a MIS of $I(F)$ if U is one of the initial ideal $in_\prec(I(F))$ (see Exercise 3.5.1 in [10]). Thus, if p is effective lucky for $(F, \prec)$ and U is a MIS of $in_\prec(I(F))$ then U is also a MIS of $I(F)$ and $I_p(F)$. Here, we define intermediate primary decomposition in general setting as follows (a certain generalization of one in [19]).

Definition 2.3.1. *Let I be an ideal. Then, a set of ideals $\mathcal{Q}$ is called an* intermediate primary decomposition (IPD) *of I if*

(a) *for all $Q \in \mathcal{Q}$,* $\mathrm{Ass}(Q) \subset \mathrm{Ass}(I)$,
(b) $\bigcap_{Q \in \mathcal{Q}} Q = I$.

We call $Q \in \mathcal{Q}$ an intermediate primary component of I. In particular, when there is a subset U of X s.t. $\mathrm{Ass}(Q) = \mathrm{Ass}_U(I)$, *we call Q an intermediate component of I via U.*

We remark that $\bigcup_{Q \in \mathcal{Q}} \mathrm{Ass}(Q) = \mathrm{Ass}(I)$. For computing intermediate primary decomposition, the following Corollary is very useful to generate prime divisors.

Corollary 2.3.2. *Let F be a Gröbner basis of I and p a permissible prime number for F. Let U be a subset of X such that $\mathrm{Ass}_U(I_p(F))$ is not empty, and $\bar{H}$ a Gröbner basis of $\bar{J} = \bigcap_{P_p \in \mathrm{Ass}_U(I_p(F))} P_p$. Let H_{can} be a Gröbner basis candidate constructed from $\bar{H}$ and $J = I(H_{can})$. Assume p is permissible for H_{can}. Suppose H_{can} is a Gröbner basis of J and p is effectively lucky for the reduced Gröbner basis L of $(I : J)$ with $I_p(L) = (I_p(F) : I_p(H_{can}))$. If J is a prime ideal then J is a prime divisor of I.*

Proof. To apply Theorem 2.2.9 for I and J, we check the conditions. First, since p is effectively lucky for L, p is compatible with L by Remark 2.2.2. Thus, $\phi_p((I(F) : I(H_{can})) \cap \mathbb{Z}_{(p)}[X]) = I_p^0(L) = I_p(L) = (I_p(F) : I_p(H_{can}))$. From the assumption, p is permissible

for H_{can}. As $I_p(H_{can}) = \bar{J}$ is an intersection of prime divisors of $I_p(F)$, it follows that $(I_p(F) : (I_p(F) : I_p(H_{can})) = I_p(H_{can})$ by Theorem 2.1.4. Thus, $\phi_p(H_{can}) = \bar{H}$ is a Gröbner basis of $(I_p(F) : (I_p(F) : I_p(H_{can}))$. It is obvious that $H_{can} \subset J$. Hence, all conditions in Theorem 2.2.9 hold and thus J is a prime divisor of I. □

When J is not prime, we can check the radicality of J by the following lemma. For any effectively lucky p for H_{can}, if $\langle \bar{H} \rangle$ is radical then $\langle H_{can} \rangle$ is also radical.

Lemma 2.3.3 ([16], Lemma 6.7). *Suppose that H_{can} is the output of our CRT modular computation, that is, it satisfies the following:*

(1) *p is permissible for $(H_{can}, \prec)$,*
(2) *$\phi_p(H_{can})$ coincides with the reduced Gröbner basis of $\sqrt{I_p(F)}$*
(3) *$H_{can} \subset \sqrt{I(F)}$*

Then H_{can} is the reduced Gröbner basis of $\sqrt{I(F)}$ with respect to $\prec$.

We can extend Corollary 2.3.2 to intersection of prime divisors by using Theorem 2.1.4 as Proposition 2.3.4. We can ensure that the lifted ideal $I(H_{can})$ is radical from Lemma 2.3.3 and an intersection of prime divisors I from Theorem 2.1.4 and Theorem 2.2.8.

Proposition 2.3.4. *Under the conditions of Corollary 2.3.2 (except the primality of J), if J is a radical ideal then J is some intersections of prime divisors of I.*

We note that, if $\mathrm{Ass}_U(I_p(F))$ consist of one prime, that is, $\bar{J}$ is prime, then we check if J is prime or not more easily. Moreover, if $\mathrm{Ass}_U(I_p(F))$ consists of two prime ideals $\bar{P}_1$ and $\bar{P}_2$ and then we combine those prime divisors and apply the criterion for radical to the lifting of $\bar{P}_1 \cap \bar{P}_2$. We also make the same argument for $\bar{P}_1 \cap \bar{P}_2 \cap \bar{P}_3$, $\bar{P}_1 \cap \bar{P}_2 \cap \bar{P}_3 \cap \bar{P}_4$ and so on.

Example 2.3.5. *Let $I = (x) \cap (x^3, y) \cap (x^2 + 1)$ Let $F = \{x^3y + xy, x^5 + x^3\}$ be the reduced Gröbner basis of I. We consider two prime numbers $p = 3, 5$. Then,* $\mathrm{Ass}(I_3(F)) = \{(x), (x, y), (x^2 + 1)\}$ *and* $\mathrm{Ass}(I_5(F)) = \{(x), (x, y), (x + 2), (x + 3)\}$. *For $U_1 = \{y\}$ and $U_2 = \emptyset$,* $\mathrm{Ass}_{U_1}(I_3(F)) = \{(x), (x^2 + 1)\}$ *and* $\mathrm{Ass}_{U_2}(I_3(F)) = \{(x, y)\}$. *Similarly,* $\mathrm{Ass}_{U_1}(I_5(F)) = \{(x), (x + 2), (x + 3)\}$ *and* $\mathrm{Ass}_{U_2}(I_5(F)) = \{(x, y)\}$. *For $J_p(U) = \bigcap_{P_p \in \mathrm{Ass}_U(I_p(F))} P_p$, it follows that $J_3(U_1) = (x^3 + x)$, $J_5(U_1) = (x^3 + x)$, $J_3(U_2) = (x, y)$ and $J_5(U_2) = (x, y)$. By using CRT, we may compute radicals of intermediate primary components $J_{can}(U_1) = (x^3+x)$ and $J_{can}(U_2) = (x, y)$. Finally, we obtain an intermediate primary decomposition $\{(x^3 + x), (x^3, y)\}$ of I from Lemma 2.1.7 and Theorem 2.1.11.*

Finally, we sketch an outline of intermediate primary decomposition via MIS as follows. Its termination comes from the finiteness of unlucky primes for computation of associated prime divisors and primary components.

Intermediate Primary Decomposition via MIS

Input: F: a Gröbner basis of an ideal I.
Output: $\{Q(U)\}$: an IPD via MIS of I.

(Step 1) choose $\mathcal{P}$ as a list of random primes satisfying primeTest
(Step 2) compute $\mathrm{Ass}(I_p(F))$ for $p \in \mathcal{P}$ and choose a set of MISs $\mathcal{U}$ from $\mathrm{Ass}(I_p(F))$
(Step 3) compute $J_p(U) = \bigcap_{P_p \in \mathrm{Ass}_U(I_p(F))} P_p$ for each $U \in \mathcal{U}$ and let $\mathcal{JP}(U) = \mathcal{JP}(U) \cup \{J_p(U)\}$
(Step 4) delete unlucky p for $\mathcal{JP}(U)$ and obtain $\mathcal{JP}_{lucky}(U)$
(Step 5) lift $\mathcal{JP}_{lucky}(U)$ to $J_{can}(U)$ by CRT and rational reconstruction. If $J_{can}(U)$ is unmixed then go to Step 6; otherwise RESTART
(Step 6) if $J_{can}(U)$ passes finalTest (Proposition 2.3.4) then go to Step 7: otherwise RESTART
(Step 7) compute an intersection of primary components $Q(U)$ by $\mathrm{hull}(I + J_{can}(U)^m)$ (Lemma 2.1.7 and 2.1.9) or $\mathrm{hull}((I : (I : J_{can}(U)^\infty)^\infty))$ (Theorem 2.1.11) for isolated cases
(Step 8) if $\bigcap_{U \in \mathcal{U}} Q(U) = I$ then return $\{Q(U)\}$; otherwise RESTART

RESTART: enlarge $\mathcal{P}$ with prime numbers not used so far and go back to Step 2

3 EXPERIMENTS

In this section, we see some naive experiments on Singular [6]. Timings (in seconds) are measured in real time and on a PC with Intel Core i7-8700B CPU with 32GB memory. We see several examples with intermediate coefficient growth. The source code for several algorithms (`modQuotient`, `modSat` and `modDiq`) is open in https://github.com/IshiharaYuki/moddiq.

To implement modular algorithms for (double) ideal quotient and saturation, we use the library `modular.lib`. A function `modular` returns a candidate from modular computations by CRT and rational reconstruction. As the optional arguments, the function has `primeTest`, `deleteUnluckyPrimes`, `pTest` and `finalTest`. In this paper, we implemented `primeTest`, `pTest` and `finalTest` for ideal quotient and saturation. Also, we use Singular implemented functions `quotient` and `sat` to compute $(I : J)$ and $(I : J^\infty)$ respectively (about computations of ideal quotient and saturation, see [10]). We explain some details of our implementations. First, `modQuotient` computes ideal quotient by modular techniques based on Lemma 2.2.6. Second, `modSat` computes saturation by modular techniques based on Lemma 2.2.12 and Lemma 2.2.13. Third, `diq` computes double ideal quotient by using `quotient` twice and `modDiq` computes double ideal quotient based on Theorem 2.2.8. The function `modDiq` uses `modQuotient` to check the condition that $(I_p(F) : I_p(G)) = \phi_p((I(F) : I(G)) \cap Z_{(p)}[X])$ and $K_{can} \subset (I(F) : (I(F) : I(G)))$ in Theorem 2.2.8. Of course, we can compute double ideal quotient by using `modQuotient` twice.

Here, we use the degree reverse lexicographical ordering (`dp` on Singular) .We tested our implementation by "cyclic ideal", where $cyclic(n)$ is defined in $\mathbb{Q}[x_1, \ldots, x_n]$ (see the definition in [4]). We let $P_1 = (-15x_5 + 16x_6^3 - 60x_6^2 + 225x_6 - 4, 2x_5^2 - 7x_5 + 2x_6^2 - 7x_6 + 28, (4x_6 - 1)x_5 - x_6 + 4, 4x_1 + x_5 + x_6, 4x_2 + x_5 + x_6, 4x_3 + x_5 + x_6, 4x_4 + x_5 + x_6)$ and $P_2 = (x_2^2 + 4x_2 + 1, x_1 + x_2 + 4, x_3 - 1, x_4 - 1, x_5 - 1, x_6 - 1)$ be prime divisors of $cyclic(6)$. Let $Q_1 = ((-15x_5 + 16x_6^3 - 60x_6^2 + 225x_6 - 4)^2, (2x_5^2 - 7x_5 + 2x_6^2 - 7x_6 + 28)^2, (4x_6 - 1)x_5 - x_6 + 4, 4x_1 + x_5 + x_6, 4x_2 + x_5 + x_6, 4x_3 + x_5 + x_6, 4x_4 + x_5 + x_6)$ be a P_1-primary ideal. Also, we let $I_1 = (8x^2y^2 + 5xy^3 + 3x^3z + x^2yz, x^5 + 2y^3z^2 + 13y^2z^3 + 5yz^4, 8x^3 + 12y^3 + xz^2, 7x^2y^4 + 18xy^3z^2 + y^3z^3)$ be a modification of an ideal appeared in [3] and $I_2 = (xw_{11} - yw_{10}, yw_{12} - zw_{11}, -w_{11}w_{20} + w_{21}w_{10}, -w_{21}w_{12} + w_{22}w_{11})$ be $A_{2,3,3}$ (see [20]). As inputs, we used their Gröbner bases.

In Table 1, we can see that `modQuotient` is very effective for computation of such ideals. In table 2, we compare timings of computations of saturation in each method. To consider ideals with non-prime components, we take an intersection or products of

ideals. We can see that modSat is very effective even when multiplicities of target primary components are large. In table 3, we see results of prime divisors checks by double ideal quotient in each method. We can see that modular methods "double `modQuotient`" and `modDiq` are very efficient, comparing with the rational `diq`. In almost cases in the table, `modDiq` is faster than `modQuotient` since the final test (Theorem 2.2.8) may have some effectiveness for efficient computations.

As a whole, we examined the efficiency of modular techniques for ideal quotients by computational experiments.

ideal quotient	`quotient`	`modQuotient`
$(cyclic(6) : P_1)$	35.0	11.2
$(cyclic(6) : P_2)$	15.1	7.65
$(I_1^2 : I_1)$	7.80	0.32
$(I_1^3 : I_1)$	255	7.67
$(I_1^4 : I_1)$	2137	68.8
$(I_1 I_2 : I_2)$	0.88	0.72

Table 1: Ideal quotient

saturation	`sat`	`modSat`
$((cyclic(6) \cap Q_1) : P_1^\infty)$	86.9	16.4
$(I_1 I_2^2 : I_2^\infty)$	1264	21.9
$((I_1 \cdot (x^{100}, xy)) : (x, y)^\infty)$	0.33	0.13
$((I_1 \cdot (x^{500}, xy)) : (x, y)^\infty)$	27.3	1.18
$((I_1 \cdot (x^{1000}, xy)) : (x, y)^\infty)$	201	4.25

Table 2: Saturation

[ideal, prime divisor]	`diq`	double `modQuotient`	`modDiq`
$[cyclic(6), P_1]$	37.0	28.9	17.8
$[cyclic(6), P_2]$	15.3	9.36	11.3
$[I_1^3, (x, y)]$	13.1	8.96	5.32
$[I_1^4, (x, y)]$	254	81.7	41.4
$[I_1^2 I_2, (x, y, z)]$	143	80.7	29.1

Table 3: Double ideal quotient

4 CONCLUSION AND REMARKS

In this paper, we apply modular techniques to effective localization and double ideal quotient. Double ideal quotient and its variants are used to prime divisor check and generate primary component. Modular techniques can avoid intermediate coefficient growth and thus we can compute double ideal quotient and its variants efficiently. We also devise new algorithms for modular prime divisor check and intermediate primary decomposition. We have already implemented `modQuotient`, `modSat` and `modDiq` on Singular, and we can see that modular techniques are very effective for several examples in experiments.

We are on the way to implement Associated Check (Algorithm 2, 3) and complete an efficient algorithm of Intermediate Primary Decomposition via MIS. In particular, we can expect that Algorithm 2 will also be efficient for examples we see in the experiments of `modDiq`. Combining Algorithm 2 and Algorithm 3, we may have a new test for prime divisors as follows. First, we choose a list of random primes $\mathcal{P}$ and check $(I_p(F) : (I_p(F) : I_p(G))) = I_p(G)$ for each $p \in \mathcal{P}$, where $I(F)$ is an ideal and $I(G)$ is a prime ideal as inputs. Second, if prime numbers s.t. $(I_p(F) : (I_p(F) : I_p(G))) = I_p(G)$ are majority then we go to Algorithm 2; otherwise, go to Algorithm 3. Finally, we continue to enlarge $\mathcal{P}$ until we pass the associated test (Corollary 2.2.10) or the non-associated test (Theorem 2.2.11). Also, we can compute a Gröbner basis of $(I(F) : I(G))$ at the same time during the algorithms.

As our future work, we continue to improve the implementations and extend experiments to other examples. Also, we are thinking about intermediate primary decomposition in another way e.g. by double saturation.

ACKNOWLEDGEMENTS

This work has been advanced during the author's research stay at Technische Universität Kaiserslautern, supported by Overseas Challenge Program for Young Researchers of Japan Society for the Promotion of Science. The author is very grateful to the Singular team for fruitful discussions and kind hospitality there. In particular, he is very thankful to Wolfram Decker and Hans Schönemann for helpful advice of modular techniques and programming on Singular at Kaiserslautern. He appreciates the kind support of the computational facility by Masayuki Noro. He would like to thank his supervisor, Kazuhiro Yokoyama, for constructive comments and suggestions for the paper.

REFERENCES

[1] Atiyah, M.F., MacDonald, I.G.: Introduction to Commutative Algebra. Addison-Wesley Series in Mathematics. Avalon Publishing, New York (1994)
[2] Afzal, D., Kanwal, F., Pfister, G., Steidel, S.: Solving via Modular Methods. In: Bridging Algebra, Geometry, and Topology, Springer Proceedings in Mathematics & Statistics, vol. 96, 1-9 (2014)
[3] Arnold, E.: Modular algorithms for computing Gröbner bases. J. Symb. Comput. 35, 403-419 (2003)
[4] Backelin, J., Fröberg, R. How we prove that there are exactly 924 cyclic 7-roots. In: Proceedings of ISSAC 91, ACM Press, 103-111 (1991)
[5] Böhm, J., Decker, W., Fieker, C., Pfister, G.: The use of bad primes in rational reconstruction. Math. Comput. 84, 3013-3027 (2015)
[6] Decker, W.; Greuel, G.-M.; Pfister, G.; Schönemann, H.: Singular 4-1-2 — A computer algebra system for polynomial computations. http://www.singular.uni-kl.de (2019).
[7] Eisenbud, D., Huneke, C., Vasconcelos, W.: Direct methods for primary decomposition. Inventi. Math.110 (1), 207-235 (1992)
[8] Gräbe, H.: On lucky primes. J. Symb. Comput. 15, 199-209 (1993)
[9] Gianni, P., Trager, B., Zacharias, G.: Gröbner bases and primary decomposition of polynomial ideals. J. Symb. Comput. 6(2), 149-167 (1988)
[10] Greuel, G.-M., Pfister, G.: A Singular Introduction to Commutative Algebra. Springer, Heidelberg (2002). https://doi.org/10.1007/978-3-662-04963-1
[11] Idrees, N., Pfister, G., Steidel, S.: Parallelization of modular algorithms. J. Symb. Comput. 46, 672-684 (2011)
[12] Ishihara Y., Yokoyama K.: Effective Localization Using Double Ideal Quotient and Its Implementation. In: Computer Algebra in Scientific Computing CASC 2018, LNCS, vol. 11077, Springer, pp.272-287 (2018)
[13] Kawazoe, T., Noro, M.: Algorithms for computing a primary ideal decomposition without producing intermediate redundant components. J. Symb. Comput. 46(10), 1158-1172 (2011)
[14] Matzat, B.H., Greuel, G.-M., Hiss, G.: Primary decomposition: algorithms and comparisons. In: Matzat, B.H., Greuel, G.M., Hiss, G. (eds.) Algorithmic Algebra and Number Theory, pp. 187-220. Springer, Heidelberg (1999). https://doi.org/10.1007/978-3-642-59932-3
[15] Noro, M.: Modular algorithms for computing a generating set of the syzygy module. In: Computer Algebra in Scientific Computing CASC 2009, LNCS, vol. 5743, pp. 259-268. Springer (2009)
[16] Noro, M., Yokoyama, K. Usage of Modular Techniques for Efficient Computation of Ideal Operations. Math.Comput.Sci. 12(1): 1-32, (2018)
[17] Pauer, F.: On lucky ideals for Gröbner bases computations. J. Symb. Comput. 14, 471-482 (1992)
[18] Pfister, G.: On modular computation of standard basis. Anal. Stiint. Univ. Ovidius Constanta 15, 129-138 (2007)
[19] Shimoyama, T., Yokoyama, K.: Localization and primary decomposition of polynomial ideals. J. Symb. Comput. 22(3), 247-277 (1996)
[20] Sturmfels, B.: Solving systems of polynomial equations. In: CBMS Regional Conference Series. American Mathematical Society, no. 97 (2002)
[21] Vasconcelos, W.: Computational Methods in Commutative Algebra and Algebraic Geometry. Algorithms and Computation in Mathematics. Springer, Heidelberg (2004)

How Many Zeros of a Random Sparse Polynomial Are Real?

Gorav Jindal
gorav.jindal@gmail.com
Department of Computer Science, Aalto University
Espoo, Finland

Anurag Pandey
apandey@mpi-inf.mpg.de
Max Planck Institut für Informatik,
Saarland Informatics Campus
Saarbrücken, Germany

Himanshu Shukla
hshukla.math04@gmail.com
Max Planck Institut für Informatik,
Saarland Informatics Campus
Saarbrücken, Germany

Charilaos Zisopoulos
zisopoulos@cs.uni-saarland.de
Department of Computer Science, Saarland University,
Saarland Informatics Campus
Saarbrücken, Germany

ABSTRACT

We investigate the number of real zeros of a univariate k-sparse polynomial f over the reals, when the coefficients of f come from independent standard normal distributions. Recently Bürgisser, Ergür and Tonelli-Cueto showed that the expected number of real zeros of f in such cases is bounded by $O(\sqrt{k}\log k)$. In this work, we improve the bound to $O(\sqrt{k})$ and also show that this bound is tight by constructing a family of sparse support whose expected number of real zeros is lower bounded by $\Omega(\sqrt{k})$. Our main technique is an alternative formulation of the Kac integral by Edelman-Kostlan which allows us to bound the expected number of zeros of f in terms of the expected number of zeros of polynomials of lower sparsity. Using our technique, we also recover the $O(\log n)$ bound on the expected number of real zeros of a dense polynomial of degree n with coefficients coming from independent standard normal distributions.

CCS CONCEPTS

• **Theory of computation → Algebraic complexity theory**;
• **Mathematics of computing → Continuous functions**.

KEYWORDS

Sparse Polynomials, Real Tau conjecture, Random polynomials

ACM Reference Format:
Gorav Jindal, Anurag Pandey, Himanshu Shukla, and Charilaos Zisopoulos. 2020. How Many Zeros of a Random Sparse Polynomial Are Real?. In *International Symposium on Symbolic and Algebraic Computation (ISSAC '20), July 20–23, 2020, Kalamata, Greece.* ACM, New York, NY, USA, 8 pages. https://doi.org/10.1145/3373207.3404031

ISSAC '20, July 20–23, 2020, Kalamata, Greece

ACM ISBN 978-1-4503-7100-1/20/07…$15.00
https://doi.org/10.1145/3373207.3404031

1 INTRODUCTION

Understanding the number of real zeros of a given real univariate polynomial has always been of interest, both from a theoretical as well as an application point of view in science, engineering and mathematics.

1.1 Zeros of Sparse Polynomials

A lot of the polynomials that we encounter in applications are *sparse*, i.e., their degree is considerably larger than their number of monomials. This motivates studying the question for the sparse polynomials. Descartes' famous rule of signs from the 17th century [6] already sheds some light by bounding the number of non-zero real zeros of a k-sparse $f \in \mathbb{R}[x]$ [1] by $2k-2$. There are polynomials which achieve this bound too. Having some understanding on the number of real roots of k-sparse polynomials, it makes sense to ask the same question for generalizations.

In this spirit, Kushnirenko initiated a systematic study of the number of real zeros of systems of multivariate sparse polynomial equations. He coined the term "fewnomials" for sparse polynomials and made a series of hypotheses connecting the number of real zeros of a system of multivariate polynomial equations to the complexity of symbolic description of the same system. We refer the readers to a letter [17] which he wrote to Frank Sottile telling about the story of the genesis of this study. Since the formulation of the hypotheses in late 1970s, there has been a lot of work on bounding the number of real zeros of a system of sparse polynomials, most notably [1] and [10]. See [24] and [9] for surveys on the topic.

In the setting of a single univariate polynomial, however, our understanding seems very limited. For instance, if we consider the first non-trivial generalization, i.e. if we consider polynomials of the form $fg+1$, where f and g are both k-sparse, to the best of our knowledge, no bound better than the one given by Descartes' rule of sign is known. In particular, no sub-quadratic bound is known. We also do not know of any example where the number of real roots of $fg+1$ is super-linear in k.

[1] throughout this article, polynomials considered are over reals and have degree n with $n >> k$.

1.2 Connections to algebraic complexity theory: Real Tau Conjecture

Koiran [14] provided a strong motivation for computer scientists to consider generalizations like the ones above in 2011, when he considered the number of real zeros of the sum of products of sparse polynomials. He formulated *the real τ-conjecture* claiming that if a polynomial is given as

$$f = \sum_{i=1}^{m} \prod_{j=1}^{t} f_{ij}$$

where all f_{ij}'s are k-sparse, then the number of real zeros of f is bounded by a polynomial in $O(mkt)$. Thus the conjecture claims that a univariate polynomial computed by a depth-4 arithmetic circuit (see [21, 22] for background on arithmetic circuits) with the fan-in of gates at the top three layers being bounded by m, t and k respectively will have $O((mkt)^c)$ real zeros for some positive constant c. Notice that applying Descartes' bound only gives an exponential bound on the number of real zeros of f, since a-priori the sparsity bound that we can achieve for f is only $O(mk^t)$.

What is of particular interest is the underlying connection of this conjecture to the central question of algebraic complexity theory. Koiran showed that the conjecture implies a superpolynomial lower bound on the arithmetic circuit complexity of the permanent, hence establishing the importance of the question of understanding real roots of sparse polynomials from the perspective of theory of computation as well. In fact this connection is what inspired the authors to investigate the problems considered in this article.

The real τ-conjecture itself was inspired by the Shub and Smale's τ-conjecture [23] which asserts that the number of integer zeros of a polynomial with arithmetic circuit complexity bounded by s will be bounded by a polynomial in s. This conjecture also implies a superpolynomial lower bound on the arithmetic circuit size of the permanent [4] and also implies $\mathrm{P}_{\mathbb{C}} \neq \mathrm{NP}_{\mathbb{C}}$ in the Blum-Shub-Smale model of computation (see [2, 23]). Koiran's motivation was to connect the complexity theoretic lower bounds to the number of real zeros instead of the number of integer zeros, because the latter takes one to the realm of number theory where problems become notoriously hard very quickly.

While the real τ-conjecture remains open (see [11, 15, 16] for some works towards it), Briquel and Bürgisser [3] showed that the conjecture is true in the average case, i.e. they show that when the coefficients involved in the description of f are independent Gaussian random variables, then the expected number of real zeros of f is bounded by $O(mk^2t)$.

1.3 Zeros of random sparse univariate polynomials

In order to gain a better understanding of the behavior of the number of real zeros for sparse polynomials and its generalizations, we study the case of a single univariate sparse random polynomial. In this article, we only consider the case when the coefficients are identically distributed independent standard normal random variables.

With respect to this consideration, the dense case, where there are no restrictions on the sparsity, thus we have a polynomial f of degree n with all its $n+1$ coefficients as standard normal random variables, has been extensively studied and is well understood. It has been considered among others by Littlewood, Offord, Erdős, Kac, Edelman, Kostlan for various distribution since the 30s (see for instance [7, 8, 12, 18]). For this article, the works in [7, 12] are most relevant, since it was Kac [12] who showed the first $O(\log n)$ bound for the expected number of real zeros for the dense case when the coefficients are standard normal random variables. It seems very surprising that there are so few real zeros in the random case. Edelman and Kostlan [7] gave an alternative, simpler derivation for the same bound, in addition to providing essential insights to the integral and numerous generalizations in a variety of cases.

In the sparse case, there is a line of work considering the case of the multivariate system of random equations (for instance see [13, 19, 20]). However their focus is different and we are not aware of any useful adaptations to the univariate case. In fact, we do not know of any such progress until the recent work of Bürgisser, Ergür and Tonelli-Cueto [5] which showed that for a random k-sparse univariate polynomial, the expected number of real roots in the standard normal case, is bounded by $\frac{4}{\pi}\sqrt{k}\log k$, where the base of the logarithm is e, as will be everywhere else in this article unless stated otherwise. Thus they show that in this setting, the number of real zeros is much less than the Descartes bound.

Before we state our results we set up some notations. Consider a set $S = \{e_1, \ldots, e_k\} \subseteq \mathbb{N}$ of natural numbers. For such a set S, one asks how many roots (in expectation) of the random polynomial $f_S = \sum_{i=1}^{k} a_i x^{e_i}$ (here a_i's are independent standard normals) are real. For an open interval $I \subseteq \mathbb{R}$, we use z_S^I to denote the expected number of roots of f_S in I. To avoid some degeneracy issues, we always assume $0 \notin I$, this assumption allows us to assume that the smallest element of S is zero. In this paper, we are only concerned with the case when $I = (0, 1)$. See Remark 1 on why this is sufficient. When $I = (0, 1)$, we simply use z_S to denote z_S^I.

Our main contribution is the improvement on the bound on the expected number of real zeros of a random k-sparse polynomial f and proving that this is the best one can do.

Theorem 1. *Let $S \subseteq \mathbb{N}$ be any set as above with $|\ S\ | = k$, then we have $z_S \leq \frac{2}{\pi}\sqrt{k-1}$.*

Remark 1. Since our bound in Theorem 1 only depends on the size of S, and not on the structure of S, we get that $z_S^{\mathbb{R}^*} = 4z_S^{(0,1)}$. For $S = \{e_1, \ldots, e_k\}$, $z_S^{(1,\infty)}$ is equal to $z_{S'}^{(0,1)}$ for $S' = \{n - e_1, \ldots, n - e_k\}$ by replacing x by $\frac{1}{x}$ and multiplying by x^n, where n is the degree of f_S. Also $z_S^{(-\infty,0)} = z_S^{(0,\infty)}$ by replacing x by $-x$.

Theorem 2. *There exists a sequence of sets $S_k \subset \mathbb{N}$ with $|S_k| = k+2$ such that for $k \geq 3$, $z_{S_k} \geq \frac{\pi-\sqrt{3}}{16\pi}\sqrt{k} + \frac{1}{7}$.*

Theorem 2 shows that the bound obtained in Theorem 1 is tight and cannot be reduced further for an arbitrary, in terms of just the size of S, $S \subset \mathbb{N}$.

Using our techniques, we confirm the intuition from the dense case that in expectation, all the roots are concentrated around 1 i.e. for any small constant $\epsilon > 0$, the expected number of roots in $(0, 1-\epsilon)$ is bounded by a constant independent of n and k.

Theorem 3. *For a fixed $\epsilon > 0$ and any $S \subseteq \mathbb{N}$ as above, we have*

$$z_S^{(0,1-\epsilon)} \leq \frac{1}{2\pi}\left(\log\left(\frac{2}{\epsilon}\right) + \frac{4}{\sqrt{\epsilon}} - 4\right).$$

1.4 Proof ideas

Our main technical contribution is an alternative formulation of the Kac integral by Edelman-Kostlan, that we call the *Edelman-Kostlan integral* and is presented in detail in Section 2.

The formulation allows us to bound $z_{S_1 \uplus S_2}$ in terms of the bounds on z_{S_1} and z_{S_2} (presented in Subsection 2.2). Thus we can build our k-sparse polynomial monomial-by-monomial. We show that every time we add a monomial, we do not increase the expected number of roots by a lot. A careful application of this idea yields the desired $O(\sqrt{k})$ bound (presented in Section 3).

We also obtain a bound on $z_{S_1+S_2}$ in terms of z_{S_1} and z_{S_2}, where $S_1 + S_2$ is the set obtained as a result of the addition of elements of S_1 and S_2, that is, the so-called Minkowski sum of sets S_1 and S_2 (presented in Subsection 2.1). Combining the bounds on $z_{S_1+S_2}$ and $z_{S_1 \uplus S_2}$ allows us to recover the $O(\log n)$ bound for the dense case i.e. $S = \{0, 1, \ldots, n\}$, where we build up our set S as a combination of unions and Minkowski sums of sets (presented in the full version).

Further, the proof that all the roots are concentrated around 1 follows from the analysis of an approximation of the Edelman-Kostlan integral. This approximation which is inspired by the one used in [5] makes the analysis of the integral simpler.

Finally in Section 5, we show that we cannot obtain a better bound for an arbitrary $S \subset \mathbb{N}$. We show this by applying the idea of monomial-wise construction of a polynomial (presented in Section 2.2) on a carefully chosen monomial sequence, thus proving Theorem 2.

1.5 Previous work: known bounds on z_S^I

In this subsection, we present the state of the art prior to this work for z_S^I.

For $S = \{0, 1, 2, \ldots, n\}$ and $I = \mathbb{R}$, z_S^I is known to be bounded by $O(\log n)$.

Theorem 4 ([7, 12])**.** *If $S = \{0, 1, 2, \ldots, n\}$ then*

$$z_S^{\mathbb{R}^*} = \frac{2}{\pi}\log(n) + C_1 + \frac{2}{n\pi} + O\left(\frac{1}{n^2}\right).$$

Here $C_1 \approx 0.6257358072\ldots$.

Determining the value of z_S^I for arbitrary sets S remains an open problem. Towards this the best bound known was the following result by [5].

Theorem 5 ([5, Theorem 1.3])**.** *Let $S \subseteq \mathbb{N}$ be any set as above with $|\ S\ | = k$ then we have*

$$z_S \leq \frac{1}{\pi}\sqrt{k}\log(k).$$

2 PRELIMINARIES

Since our method builds upon the Edelman-Kostlan method [7] by a novel approach on analyzing their integral, it is essential to look at the method. In order to compute z_S for $S = \{e_1, \ldots, e_k\}$, define a generalization of the moment curve v_S as $v_S(t) := (t^{e_1}, t^{e_2}, \ldots, t^{e_k})$. This allows the following expression for z_S^I:

Theorem 6 ([7], Theorem 3,1)**.** *For all sets $S \subseteq \mathbb{N}$, we have the following equality for z_S^I*

$$z_S^I = \frac{1}{\pi}\int_I \frac{\sqrt{(\|v_S(t)\|_2 \cdot \|v_S'(t)\|_2)^2 - (v_S(t)\cdot v_S'(t))^2}}{(\|v_S(t)\|_2)^2}\mathrm{d}t. \quad (2.1)$$

We refer to the above integral as the *Edelman-Kostlan integral.*

The strength of this method is that the integral is parameterized by the support S and the interval I, thus allowing one to estimate the expected number of real zeros for any such arbitrary support and interval. In their paper, they compute the integral for $S = \{0, 1, \ldots, k\}$ and $I = (0, 1)$ and for these values show that z_S^I is bounded by $O(\log k)$. However, for arbitrary S of cardinality k, the integral becomes quite complicated to analyze.

In [5], they get around this difficulty by upper bounding the integral. This is achieved by ignoring the negative term of the numerator and through some elementary norm inequalities leads to the $O(\sqrt{k}\log k)$ bound. In order to further improve this bound, we believe it is necessary to analyze the above integral in new ways.

We now give an alternative formulation of the Edelman-Kostlan integral on which our proofs build upon.

Definition 1. For a set $S = \{e_1, e_2, \ldots, e_k\} \subseteq \mathbb{N}$, we define

$$g_S(t) := (\|v_S(t)\|_2)^2 = \sum_{i=1}^{k} t^{2e_i}$$

In the following lemma, we show that we can express z_S^I entirely in terms of $g_S(t)$ and its derivatives. Hence we define:

Definition 2. Let $g : \mathbb{R} \to \mathbb{R}_{\geq 0}$ be differentiable function such that $g^{-1}(0)$ is finite. Define the function $\mathcal{I}(g) : \mathbb{R} \to \mathbb{R}$,

$$\mathcal{I}(g) := \left(\frac{g'(t)}{g(t)}\right)' + \frac{g'(t)}{tg(t)} = (\log(g(t)))'' + \frac{(\log(g(t)))'}{t}.$$

Note that whenever the Edelman-Kostlan integral is well-defined, the conditions on g which make $\mathcal{I}(g)$ well-defined and non-negative are also satisfied. We now give our alternative formulation.

Lemma 1. *For all sets $S \subseteq \mathbb{N}$, we have the following equality for z_S^I*

$$z_S^I = \frac{1}{2\pi}\int_I \sqrt{\mathcal{I}(g_S(t))}\mathrm{d}t.$$

Proof. We can rewrite Equation (2.1) as

$$z_S^I = \frac{1}{\pi}\int_I \frac{\sqrt{(g_S(t)\cdot(\|v_S'(t)\|_2)^2 - (v_S(t)\cdot v_S'(t))^2}}{g_S(t)}\mathrm{d}t.$$

Now note the following equality for $v_S(t)\cdot v_S'(t)$.

$$v_S(t)\cdot v_S'(t) = \sum_{i=1}^{k} e_i t^{2e_i-1} = \frac{g_S'(t)}{2}$$

We also have the following equality for $(\|v'_S(t)\|_2)^2$.

$$\begin{aligned}(\|v'_S(t)\|_2)^2 &= \sum_{i=1}^{k} e_i^2 t^{2e_i-2} = \frac{1}{4}\left(\sum_{i=1}^{k} 4e_i^2 t^{2e_i-2}\right)\\ &= \frac{1}{4}\left(\sum_{i=1}^{k}((2e_i(2e_i-1))+2e_i)\cdot t^{2e_i-2}\right)\\ &= \frac{1}{4}\left(\sum_{i=1}^{k}(2e_i(2e_i-1)\cdot t^{2e_i-2}\right) + \frac{1}{4}\left(\sum_{i=1}^{k} 2e_i\cdot t^{2e_i-2}\right)\\ &= \frac{1}{4}g''_S(t) + \frac{1}{4t}g'_S(t).\end{aligned}$$

Therefore we can rewrite z_S^I as

$$\begin{aligned}z_S^I &= \frac{1}{\pi}\int_I \sqrt{\frac{1}{4}\left(\frac{g_S(t)\cdot(g''_S(t)+\frac{1}{t}g'_S(t)) - (g'_S(t))^2}{(g_S(t))^2}\right)}\mathrm{d}t\\ &= \frac{1}{2\pi}\int_I \sqrt{\frac{g''_S(t)}{g_S(t)} - \left(\frac{g'_S(t)}{g_S(t)}\right)^2 + \frac{g'_S(t)}{tg_S(t)}}\mathrm{d}t\\ &= \frac{1}{2\pi}\int_I \sqrt{\left(\frac{g'_S(t)}{g_S(t)}\right)' + \frac{g'_S(t)}{tg_S(t)}}\mathrm{d}t\\ &= \frac{1}{2\pi}\int_I \sqrt{\mathcal{I}(g_S(t))}\mathrm{d}t.\end{aligned}$$

□

The formulation in Definition 2 yields the following lemma:

Lemma 2. *For two non-negative functions $g_1, g_2 : \mathbb{R} \to \mathbb{R}_{>0}$, we have that $\sqrt{\mathcal{I}(g_1\cdot g_2)} \le \sqrt{\mathcal{I}(g_1)} + \sqrt{\mathcal{I}(g_2)}$.*

Proof. Consider:

$$\begin{aligned}\mathcal{I}(g_1\cdot g_2) &= (\log(g_1(t)\cdot g_2(t)))'' + \frac{(\log(g_1(t)\cdot g_2(t)))'}{t}\\ &= (\log(g_1(t)))'' + \frac{(\log(g_1(t)))'}{t}\\ &\quad + (\log(g_2(t)))'' + \frac{(\log(g_2(t)))'}{t}\\ &= \mathcal{I}(g_1) + \mathcal{I}(g_2).\end{aligned}$$

Now the claim follows by using the fact that $\sqrt{x+y} \le \sqrt{x} + \sqrt{y}$ for non-negative x, y. □

Lemma 2 allows us to give a bound on the integral when $S = S_1 * S_2$, where $*$ corresponds to the operation of either union or Minkowski sum of sets. This bound depends on the integrals associated to the corresponding sets S_1 and S_2.

2.1 Minkowski sum of sets

In this subsection, we upper bound the number of zeroes z_S when S is the Minkowski sum of two collision-free sets $S = A + B$ by the sum of the number of zeroes for the two summands.

Definition 3. For sets $A, B \subseteq \mathbb{N}$, we define the Minkowski sum of A, B as: $A + B := \{a + b : a \in A, b \in B\}$. We say two sets $A, B \subseteq \mathbb{N}$ are *collision-free* if $|\ A + B\ | = |\ A\ | \cdot |\ B\ | = |\ A \times B\ |$, i.e., when all the "$a + b : a \in A, b \in B$" are distinct.

Now we show how to apply this definition in the context of the above formulation of z_S^I and $\mathcal{I}(g)$.

Lemma 3. *If $S_1, S_2 \subseteq \mathbb{N}$ are two collision-free sets, then $z^I_{S_1+S_2} \le z^I_{S_1} + z^I_{S_2}$.*

Proof. It is easy to see from the definition of g_S, when S_1, S_2 are collision-free, we have

$$g_{S_1+S_2}(t) = g_{S_1}(t)\cdot g_{S_2}(t)$$

Therefore we obtain

$$\begin{aligned}z^I_{S_1+S_2} &= \frac{1}{2\pi}\int_I \sqrt{\mathcal{I}(g_{S_1+S_2}(t))}\mathrm{d}t = \frac{1}{2\pi}\int_I \sqrt{\mathcal{I}(g_{S_1}(t)\cdot g_{S_2}(t))}\mathrm{d}t\\ &\le \frac{1}{2\pi}\int_I \sqrt{\mathcal{I}(g_{S_1}(t))}\mathrm{d}t + \frac{1}{2\pi}\int_I \sqrt{\mathcal{I}(g_{S_2}(t))}\mathrm{d}t\\ &= z^I_{S_1} + z^I_{S_2}.\end{aligned}$$

, where the last inequality follows from Lemma 2. □

2.2 Union of sets

In this subsection, we provide an upper bound for another set operation on the support S. Specifically, we want to find upper bounds for $z_{S_1 \uplus S_2}$, here $S_1 \uplus S_2$ denotes the disjoint union of S_1 and S_2. First we state the following proposition which is easy to verify.

Proposition 1. *If $S_1, S_2 \subseteq \mathbb{N}$ are two disjoint sets then $g_{S_1\uplus S_2}(t) = g_{S_1}(t) + g_{S_2}(t)$.*

We need the following definition to give our result for expressing $z_{S_1\uplus S_2}$ in terms of z_{S_1} and z_{S_2}.

Definition 4. Let $S_1, S_2 \subseteq \mathbb{N}$ be two disjoint sets with $\left(\frac{g_{S_1}}{g_{S_2}}\right)' \ge 0$ at zero. Let $c_1, \ldots, c_m$ (with $c_i \le c_{i+1}$) be the critical points of odd multiplicity of $\frac{g_{S_1}}{g_{S_2}}$ in $(0,1)$. Define $c_0 := 0$ and $c_{m+1} := 1$. We define the following quantities, here $0 \le i \le m$ and $c \in (0,1)$.

$$\begin{aligned}\gamma_{S_1,S_2}(c) &= \sqrt{\frac{g_{s_1}(c)}{g_{s_2}(c)}}\\ T^i_{S_1,S_2} &:= (-1)^i\left(\arctan(\gamma_{S_1,S_2}(c_{i+1})) - \arctan(\gamma_{S_1,S_2}(c_i))\right)\\ R_{S_1,S_2} &:= \sum_{i=0}^{m} T^i_{S_1,S_2}.\end{aligned}$$

We also state a basic easy to verify technical proposition which will be useful in the proof of the main theorem.

Proposition 2. *The following identity is true for all a, b, c, d:*

$$\left(\frac{a+c}{b+d}\right)^2 = \left(\frac{b}{b+d}\right)\left(\frac{a}{b}\right)^2 + \left(\frac{d}{b+d}\right)\left(\frac{c}{d}\right)^2 - \frac{1}{bd}\left(\frac{bc-ad}{b+d}\right)^2.$$

We may now state the key result of this section.

Lemma 4. *Let $S_1, S_2 \subseteq \mathbb{N}$ be two disjoint sets. Assume that $\left(\frac{g_{S_1}}{g_{S_2}}\right)'$ is non-negative at zero. Note that at least one of $\left(\frac{g_{S_1}}{g_{S_2}}\right)'$ and $\left(\frac{g_{S_2}}{g_{S_1}}\right)'$ is non-negative at zero. Thus, we can always rename accordingly S_1 and S_2 to ensure this is the case. Then we have*

$$z_{S_1\uplus S_2} \le z_{S_1} + z_{S_2} + \frac{1}{\pi}R_{S_1,S_2}.$$

Proof. By using Proposition 1, we know that

$$\begin{aligned}\mathcal{I}(g_{S_1 \uplus S_2}) &= \mathcal{I}(g_{S_1} + g_{S_2}) \\ &= \frac{g''_{S_1} + g''_{S_2}}{g_{S_1} + g_{S_2}} - \left(\frac{g'_{S_1} + g'_{S_2}}{g_{S_1} + g_{S_2}}\right)^2 + \frac{1}{t}\left(\frac{g'_{S_1} + g'_{S_2}}{g_{S_1} + g_{S_2}}\right) \\ &= \frac{g_{S_1}}{g_{S_1} + g_{S_2}} \cdot \mathcal{I}(g_{S_1}) + \frac{g_{S_2}}{g_{S_1} + g_{S_2}} \cdot \mathcal{I}(g_{S_2}) \\ &\quad + \frac{1}{g_{S_1} g_{S_2}} \left(\frac{g_{S_1} {g_{S_2}}' - g_{S_2} {g_{S_1}}'}{g_{S_1} + g_{S_2}}\right)^2\end{aligned}$$

The last equality follows by applying Proposition 2 on $g'_{S_1} = a, g_{S_1} = b, g'_{S_2} = c, g_{S_2} = d$. In order to simplify the notations we denote $\frac{1}{g_{S_1} g_{S_2}} \left(\frac{g_{S_1} g'_{S_2} - g_{S_2} g'_{S_1}}{g_{S_1} + g_{S_2}}\right)^2$ by W^2. Therefore we have

$$\begin{aligned}z_{S_1 \uplus S_2} &= \frac{1}{2\pi} \int_0^1 \sqrt{\mathcal{I}(g_{S_1 \uplus S_2}(t))}\mathrm{d}t \\ &= \frac{1}{2\pi} \int_0^1 \sqrt{\frac{g_{S_1}}{g_{S_1} + g_{S_2}} \cdot \mathcal{I}(g_{S_1}) + \frac{g_{S_2}}{g_{S_1} + g_{S_2}} \cdot \mathcal{I}(g_{S_2}) + W^2}\mathrm{d}t \\ &\leq \frac{1}{2\pi}\left(\int_0^1 \sqrt{\mathcal{I}(g_{S_1}(t))}\mathrm{d}t + \int_0^1 \sqrt{\mathcal{I}(g_{S_2}(t))}\mathrm{d}t + \int_0^1 |W|\,\mathrm{d}t\right) \\ &= z_{S_1} + z_{S_2} + \frac{1}{2\pi}\int_0^1 \left|\frac{1}{\sqrt{g_{S_1} g_{S_2}}}\left(\frac{g_{S_2} g'_{S_1} - g_{S_1} g'_{S_2}}{g_{S_1} + g_{S_2}}\right)\right| \mathrm{d}t.\end{aligned}$$

Now we just need to upper bound the definite integral

$$J := \int_0^1 \left|\frac{1}{\sqrt{g_{S_1} g_{S_2}}}\left(\frac{g_{S_2} g'_{S_1} - g_{S_1} g'_{S_2}}{g_{S_1} + g_{S_2}}\right)\right| \mathrm{d}t$$

The value of J in a sub-interval (α, β) of $(0, 1)$ depends upon the condition whether $g_{S_2} g'_{S_1} - g_{S_1} g'_{S_2}$ is positive or negative in (α, β). So we divide $(0, 1)$ in the intervals where $g_{S_2} g'_{S_1} - g_{S_1} g'_{S_2}$ is positive or negative. Note that $g_{S_2} g'_{S_1} - g_{S_1} g'_{S_2}$ is positive if and only if $\left(\frac{g_{S_1}}{g_{S_2}}\right)'$ is positive. Therefore $g_{S_2} g'_{S_1} - g_{S_1} g'_{S_2}$ changes sign exactly on the critical points of odd multiplicity of $\frac{g_{S_1}}{g_{S_2}}$. Suppose (α, β) is some sub-interval of $(0, 1)$ where $\left(\frac{g_{S_1}}{g_{S_2}}\right)'$ is non-negative. Let us look at the integral J in the interval (α, β). We have:

$$\begin{aligned}J_{\alpha,\beta} &:= \int_\alpha^\beta \frac{1}{\sqrt{g_{S_1} g_{S_2}}}\left(\frac{g_{S_2} g'_{S_1} - g_{S_1} g'_{S_2}}{g_{S_2}^2}\right) \cdot \left(\frac{g_{S_2}^2}{g_{S_1} + g_{S_2}}\right)\mathrm{d}t \\ &= \int_\alpha^\beta \sqrt{\frac{g_{S_2}}{g_{S_1}}} \cdot \left(\frac{g_{S_2} g'_{S_1} - g_{S_1} g'_{S_2}}{g_{S_2}^2}\right) \cdot \left(\frac{g_{S_2}}{g_{S_1} + g_{S_2}}\right)\mathrm{d}t \\ &= 2\int_\alpha^\beta \left(\sqrt{\frac{g_{S_1}}{g_{S_2}}}\right)' \cdot \left(\frac{1}{1 + \left(\sqrt{\frac{g_{S_1}}{g_{S_2}}}\right)^2}\right)\mathrm{d}t = 2\int_\gamma^\eta \left(\frac{1}{1+u^2}\right)\mathrm{d}u\end{aligned}$$

$$\left(\text{substituting } u := \sqrt{\frac{g_{S_1}}{g_{S_2}}}. \text{ Here } \gamma = \sqrt{\frac{g_{S_1}(\alpha)}{g_{S_2}(\alpha)}} \text{ and } \eta = \sqrt{\frac{g_{S_1}(\beta)}{g_{S_2}(\beta)}}\right)$$

Therefore $J_{\alpha,\beta} = 2(\arctan(\eta) - \arctan(\gamma))$. For intervals where $\left(\frac{g_{S_1}}{g_{S_2}}\right)'$ is negative, we obtain the same result by the substitution $u = \sqrt{\frac{g_{S_2}}{g_{S_1}}}$ instead, which is reflected on the definition of $T^i_{S_1,S_2}$ above. Now the claimed inequality for $z_{S_1 \uplus S_2}$ follows by using the quantities defined in Definition 4. □

3 PROOF OF THEOREM 1: $O(\sqrt{k})$ BOUND

We begin with considering the cases where either $|S| = 1$ or $|S| = 2$. This will be the base to construct an inductive argument for the general case, using Lemma 4.

Lemma 5. *For any singleton set S, we have $\mathcal{I}(g_S) = 0$.*

Proof. Suppose $S = \{a\}$, therefore $g_S(t) = t^{2a}$. Hence

$$\mathcal{I}(g_S) = (2a\log(t))'' + \frac{(2a\log(t))'}{t} = -\frac{2a}{t^2} + \frac{2a}{t^2} = 0 \qquad \square$$

Lemma 6. *For all sets S of size two, $z_S = \frac{1}{4}$.*

Proof. Without loss of generality we can assume that $S = \{0, a\}$. An easy calculation shows that $\sqrt{\mathcal{I}(g_S(t))} = \frac{2at^{a-1}}{1+t^{2a}}$. Therefore

$$z_S = \frac{2}{2\pi}\int_0^1 \frac{at^{a-1}}{1+t^{2a}}\mathrm{d}t = \frac{1}{4}. \qquad \square$$

Now we show that if we increase the sparsity of a polynomial f by adding a monomial of degree higher than the degree of f, we can bound the expected number of real zeros of the resulting polynomial in terms of the bound for the same quantity for f.

Lemma 7. *Let $S \subseteq \mathbb{N}$ be a set with $0 \in S$ and $|\ S\ | = k$. If $a \in \mathbb{N}$ is such that $a > \max(S)$ then*

$$z_{S\cup\{a\}} \leq z_S + \frac{1}{\pi}\arctan\left(\frac{1}{\sqrt{k}}\right)$$

Proof. Let us first analyze the derivative of $\frac{g_{\{a\}}}{g_S}$. We have

$$\left(\frac{g_{\{a\}}}{g_S}\right)' = \frac{1}{g_S^2}\left(2ax^{2a-1}\sum_{e\in S} x^{2e} - x^{2a}\sum_{e\in S} 2ex^{2e-1}\right) > 0 \qquad (3.1)$$

Therefore $\frac{g_{\{a\}}}{g_S}$ is always increasing in $(0, 1)$. Abusing the notation slightly, let W be such that

$$W^2 = \frac{1}{g_S g_{\{a\}}}\left(\frac{g'_{\{a\}} g_S - g'_S g_{\{a\}}}{g_S + g_{\{a\}}}\right)^2$$

(similar to Lemma 4). Hence, we have

$$\begin{aligned}\sqrt{\mathcal{I}(g_{S\cup\{a\}}(t))} &= \sqrt{\frac{g_S \cdot \mathcal{I}(g_S)}{g_S + g_{\{a\}}} + \frac{g_{\{a\}} \cdot \mathcal{I}(g_{\{a\}})}{g_S + g_{\{a\}}} + W^2} \\ &\leq \sqrt{\mathcal{I}(g_S(t))} + 0 + |W|\end{aligned}$$

By substituting into the formula for $z_{S\cup\{a\}}$, we get

$$z_{S\cup\{a\}} = \frac{1}{2\pi}\int_0^1 \sqrt{I(g_{S\cup\{a\}}(t))}\mathrm{d}t$$
$$\leq \frac{1}{2\pi}\cdot\left(\int_0^1 \sqrt{I(g_S(t))}\mathrm{d}t \quad + \right.$$
$$\left.\int_0^1 \frac{1}{\sqrt{g_S g_{\{a\}}}}\left(\frac{g'_{\{a\}}g_S - g'_S g_{\{a\}}}{g_S + g_{\{a\}}}\right)\mathrm{d}t\right)$$
$$= z_S + \frac{1}{2\pi}\int_0^1 \frac{1}{\sqrt{g_S g_{\{a\}}}}\left(\frac{g'_{\{a\}}g_S - g'_S g_{\{a\}}}{g_S + g_{\{a\}}}\right)\mathrm{d}t$$

Now we use the substitution $u = \sqrt{\frac{g_{\{a\}}}{g_s}}$ to obtain

$$\int_0^1 \frac{1}{\sqrt{g_S g_{\{a\}}}}\left(\frac{g'_{\{a\}}g_S - g'_S g_{\{a\}}}{g_S + g_{\{a\}}}\right)\mathrm{d}t = 2\int_\alpha^\beta \left(\frac{1}{1+u^2}\right)\mathrm{d}u,$$

where $\alpha = \sqrt{\frac{g_{\{a\}}(0)}{g_s(0)}} = 0$ and $\beta = \sqrt{\frac{g_{\{a\}}(1)}{g_s(1)}} = \frac{1}{\sqrt{k}}$. Note that the integrand is the derivative of $\arctan(u)$, we now have

$$2\int_\alpha^\beta \left(\frac{1}{1+u^2}\right)\mathrm{d}u = 2\left(\arctan\left(\frac{1}{\sqrt{k}}\right) - \arctan(0)\right) = 2\arctan\left(\frac{1}{\sqrt{k}}\right)$$

Hence

$$z_{S\cup\{a\}} \leq z_S + \frac{1}{2\pi}\int_0^1 \frac{1}{\sqrt{g_S g_{\{a\}}}}\left(\frac{g'_{\{a\}}g_S - g'_S g_{\{a\}}}{g_S + g_{\{a\}}}\right)\mathrm{d}t$$
$$= z_S + \frac{1}{\pi}\arctan\left(\frac{1}{\sqrt{k}}\right). \qquad \square$$

We may now prove a slightly stronger version of Theorem 1.

Theorem 7 (Theorem 1 restated)**.** *Let $S \subseteq \mathbb{N}$ be a set with $0 \in S$ and $|\ S\ | = k$. Then $z_S \leq \frac{1}{4} + \frac{2}{\pi}(\sqrt{k-1}-1) \leq \frac{2}{\pi}\cdot\sqrt{k-1}$.*

Proof. If $k \leq 2$ then the results follows from Lemma 6. So assume $k > 2$. By using Lemmas 6 and 7, we may always add the highest element iteratively and obtain that

$$z_S \leq \frac{1}{4} + \frac{1}{\pi}\sum_{i=2}^{k-1}\arctan\left(\frac{1}{\sqrt{i}}\right)$$

We use the following well-known inequality

$$\arctan(x) < x \text{ for all } x > 0.$$

This implies that

$$z_S - \frac{1}{4} \leq \frac{1}{\pi}\sum_{i=2}^{k-1}\frac{1}{\sqrt{i}} \leq \frac{1}{\pi}\sum_{i=2}^{k-1}\frac{1}{\sqrt{i}}$$
$$\leq \frac{1}{\pi}\int_1^{k-1}\sqrt{\frac{1}{x}}\mathrm{d}x = \frac{2}{\pi}(\sqrt{k-1}-1).$$

Hence the claimed bound follows. $\square$

4 PROOF OF THEOREM 3: ROOTS CONCENTRATE AROUND 1

Here we want to show that most of the roots are near 1. First we need the following proposition useful in the analysis.

Proposition 3. *For all $t \in (0,1)$, we have*

$$\sqrt{\sum_{e>0} e^2 t^{2e-2}} \leq \frac{1}{1-t^2} + \frac{2t}{(1-t^2)^{\frac{3}{2}}}$$

Proof. First use the following well-known equality

$$\frac{1}{1-t^2} = \sum_{e\geq 0} t^{2e}.$$

to obtain that

$$\left(\frac{1}{1-t^2}\right)'' = \sum_{e>0} 2e(2e-1)t^{2e-2} = \frac{2(1+3t^2)}{(1-t^2)^3}$$

Therefore

$$\sum_{e>0} e(2e-1)t^{2e-2} = \frac{(1+3t^2)}{(1-t^2)^3}.$$

Clearly

$$\sqrt{\sum_{e>0} e^2 t^{2e-2}} \leq \sqrt{\sum_{e>0} e(2e-1)t^{2e-2}} \leq \sqrt{\frac{(1+3t^2)}{(1-t^2)^3}}$$

$$= \sqrt{\frac{1}{(1-t^2)^2} + \frac{4t^2}{(1-t^2)^3}} \leq \frac{1}{1-t^2} + \frac{2t}{(1-t^2)^{\frac{3}{2}}} \qquad \square$$

We now give the proof of Theorem 3.

Proof of Theorem 3. Without loss of generality , we assume that $0 \in S$, therefore $\|v_S(t)\|_2 \geq 1$ for all $t \in \mathbb{R}$. By using the equality in Theorem 6 and also by ignoring the second term in the numerator in Equation (2.1), we get the following inequality for z_S

$$z_S^{(0,1-\epsilon)} \leq \frac{1}{\pi}\int_0^{1-\epsilon} \frac{\sqrt{(\|v_S(t)\|_2 \cdot \left\|v'_S(t)\right\|_2)^2}}{(\|v_S(t)\|_2)^2}\mathrm{d}t$$
$$= \frac{1}{\pi}\int_0^{1-\epsilon} \frac{\left\|v'_S(t)\right\|_2}{\|v_S(t)\|_2}\mathrm{d}t \leq \frac{1}{\pi}\int_0^{1-\epsilon}\left\|v'_S(t)\right\|_2\mathrm{d}t$$

By using Proposition 3, we have: $\left\|v'_S(t)\right\|_2 = \sqrt{\sum_{e\in S} e^2 t^{2e-2}} \leq \frac{1}{1-t^2} + \frac{2t}{(1-t^2)^{\frac{3}{2}}}$. Therefore

$$z_S^{(0,1-\epsilon)} \leq \frac{1}{\pi}\int_0^{1-\epsilon}\left\|v'_S(t)\right\|_2\mathrm{d}t \leq \frac{1}{\pi}\int_0^{1-\epsilon}\left(\frac{1}{1-t^2} + \frac{2t}{(1-t^2)^{\frac{3}{2}}}\right)\mathrm{d}t$$
$$= \frac{1}{\pi}\left(\int_0^{1-\epsilon}\frac{1}{1-t^2}\mathrm{d}t + \frac{1}{\pi}\int_0^{1-\epsilon}\frac{2t}{(1-t^2)^{\frac{3}{2}}}\mathrm{d}t\right)$$
$$= \frac{1}{\pi}\left(\left[\frac{1}{2}\log\left(\frac{1+t}{1-t}\right)\right]_0^{1-\epsilon} + \left[\frac{2}{\sqrt{1-t^2}}\right]_0^{1-\epsilon}\right)$$

$$= \frac{1}{\pi}\left(\frac{1}{2}\log\left(\frac{2-\epsilon}{\epsilon}\right) + \frac{2}{\sqrt{\epsilon(2-\epsilon)}} - 2\right)$$
$$\leq \frac{1}{2\pi}\left(\log\left(\frac{2}{\epsilon}\right) + \frac{4}{\sqrt{\epsilon}} - 4\right). \qquad \square$$

5 THE LOWER BOUND

In this section we will come up with a sequence of sets $(S_k)_{k\geq 1}$ such that the expected number of real zeros of the corresponding polynomials is lower bounded by $\Omega(\sqrt{k})$, for large enough k.

Lemma 8. *Suppose $S = \{e_1, e_2, \ldots, e_k\}$ with $e_k = \max(S)$ and $\epsilon > 0$, then $z_S^{(1-\epsilon,1)} \leq \frac{\epsilon\sqrt{k}e_k}{\pi}$.*

Proof. We have:

$$z_S^{(1-\epsilon,1)} \leq \frac{1}{\pi}\int_{1-\epsilon}^{1} \frac{\sqrt{(\|v_S(t)\|_2 \cdot \|v_S'(t)\|_2)^2}}{(\|v_S(t)\|_2)^2}\mathrm{d}t = \frac{1}{\pi}\int_{1-\epsilon}^{1} \frac{\|v_S'(t)\|_2}{\|v_S(t)\|_2}\mathrm{d}t$$
$$\leq \frac{1}{\pi}\int_{1-\epsilon}^{1} \|v_S'(t)\|_2\,\mathrm{d}t = \frac{1}{\pi}\int_{1-\epsilon}^{1}\left(\sum_{i=1}^{k}\left(e_i^2 t^{2e_i-2}\right)\right)^{\frac{1}{2}}\mathrm{d}t$$
$$\leq \frac{1}{\pi}\int_{1-\epsilon}^{1}\left(ke_k^2\right)^{\frac{1}{2}}\mathrm{d}t = \frac{1}{\pi}\int_{1-\epsilon}^{1}\left(\sqrt{k}e_k\right)\mathrm{d}t = \frac{\epsilon}{\pi}\left(\sqrt{k}e_k\right) \qquad \square$$

Remark 2. Thus, we can have z_S^I arbitrarily small, for a small enough ϵ. This fact will be crucial in the proof of Theorem 2. Further, Lemma 8 can be viewed as a supplementary result to Theorem 3. Theorem 3 implies that most of the roots lie in $(0, 1-\epsilon)$, if ϵ is allowed to be arbitrarily small. Lemma 8 gives a precise formulation of this fact.

5.1 Proof of Theorem 2

From now on we will assume that $S = \{0,1\}\bigcup\{2^{2^i} \mid 1 \leq i \leq k-1\}$ and $a = 2^{2^k}$. The following lemma essentially will imply that one cannot avoid summing over $\sqrt{\frac{1}{k}}$ as in the proof of Theorem 7.

Lemma 9. *Let W be as in the proof of Lemma 4, then we have $\int_{1-\frac{1}{2a}}^{1} |W|\,\mathrm{d}t \geq 2\left(\arctan\left(\frac{1}{4\sqrt{k}}\right)\right)$.*

Proof. Using the computation in the proof of Lemma 4 we have

$$\int_{1-\frac{1}{2a}}^{1} |W|\,\mathrm{d}t = 2\left(\arctan\left(\frac{1}{\sqrt{k+1}}\right) - \arctan\left(\sqrt{\frac{g_{\{a\}}(1-\frac{1}{2a})}{g_S(1-\frac{1}{2a})}}\right)\right).$$

We now upper bound the value of $\arctan\left(\sqrt{\frac{g_{\{a\}}(1-\frac{1}{2a})}{g_S(1-\frac{1}{2a})}}\right)$ by giving a lower bound on $g_S(1-\frac{1}{2a})$ and an upper bound on $g_{\{a\}}(1-\frac{1}{2a})$. Using well-known inequalities $\left(1-\frac{1}{n}\right)^n \leq \frac{1}{e}$ (for any $n \in \mathbb{N}$) and $(1+x)^r \geq 1+rx$ if $x \geq -1$ and $r > 1$, we have, for large enough k

$$g_S\left(1-\frac{1}{2a}\right) = \sum_{i=1}^{k+1}\left(1-\frac{1}{2a}\right)^{2e_i}$$
$$\geq \sum_{i=1}^{k+1}\left(1-\frac{2e_i}{2a}\right) \geq k+1-\left(\sum_{i=1}^{k+1}2^{-k}\right) \geq k$$

Therefore, $\arctan\left(\sqrt{\frac{g_{\{a\}}(1-\frac{1}{2a})}{g_S(1-\frac{1}{2a})}}\right) \leq \arctan\left(\sqrt{\frac{\frac{1}{e}}{k}}\right)$. which gives

$$2\left(\arctan\left(\frac{1}{\sqrt{k+1}}\right) - \arctan\left(\sqrt{\frac{g_{\{a\}}(1-\frac{1}{2a})}{g_S(1-\frac{1}{2a})}}\right)\right)$$
$$\geq 2\arctan\left(\frac{1}{\sqrt{k+1}}\right) - 2\arctan\left(\sqrt{\frac{\frac{1}{e}}{k}}\right)$$
$$\geq 2\left(\arctan\left(\frac{\frac{1}{\sqrt{k+1}} - \frac{1}{e\sqrt{k}}}{1+\frac{1}{e\sqrt{k(k+1)}}}\right)\right) \geq 2\left(\arctan\left(\frac{1}{4\sqrt{k}}\right)\right) \qquad \square$$

For proving Theorem 2 we will again resort to our idea of monomial-wise construction of the polynomial. The monomial sequence we choose is $e_{i+2} = 2^{2^i}$ for $i \geq 1$ with $e_1 = 0, e_2 = 1$. Before we begin the proof, recall from the proof of Lemma 4 that

$$z_{S\cup\{a\}} = \frac{1}{2\pi}\int_0^1 \sqrt{\frac{g_S}{g_S+g_{\{a\}}}\cdot\mathcal{I}(g_S) + \frac{g_{\{a\}}}{g_S+g_{\{a\}}}\cdot\mathcal{I}(g_{\{a\}}) + W^2}\mathrm{d}t$$

The key idea is to write $z_{S\cup\{a\}}$ as a sum of two integrals over disjoint intervals such that $\mathcal{I}(g_S)$ dominates in one interval while W dominates in the other, then lower bound both integrals.

Proof of Theorem 2. Recall from Lemma 5 that $\mathcal{I}(g_{\{a\}}) = 0$. Therefore we have

$$z_{S\cup\{a\}} = \frac{1}{2\pi}\left(\int_0^1 \sqrt{\frac{g_S}{g_S+g_{\{a\}}}\cdot\mathcal{I}(g_S) + 0 + W^2}\mathrm{d}t\right)$$
$$\geq \frac{1}{2\pi}\left(\int_0^{1-\frac{1}{2a}} \sqrt{\frac{g_S}{g_S+g_{\{a\}}}\cdot\mathcal{I}(g_S)}\mathrm{d}t + \int_{1-\frac{1}{2a}}^{1}|W|\,\mathrm{d}t\right)$$
$$= \frac{1}{2\pi}\int_0^1 \sqrt{\frac{g_S}{g_S+g_{\{a\}}}\cdot\mathcal{I}(g_S)}\mathrm{d}t$$
$$- \frac{1}{2\pi}\int_{1-\frac{1}{2a}}^{1} \sqrt{\frac{g_S}{g_S+g_{\{a\}}}\cdot\mathcal{I}(g_S)}\mathrm{d}t + \frac{1}{2\pi}\int_{1-\frac{1}{2a}}^{1}|W|\,\mathrm{d}t$$
$$\geq \frac{1}{2\pi}\sqrt{\frac{k+1}{k+2}}\int_0^1 \sqrt{\mathcal{I}(g_S)}\mathrm{d}t - \frac{1}{2\pi}\int_{1-\frac{1}{2a}}^{1}\sqrt{\mathcal{I}(g_S)}\mathrm{d}t$$
$$+ \frac{1}{2\pi}\int_{1-\frac{1}{2a}}^{1}|W|\,\mathrm{d}t \qquad \left(\frac{g_{\{a\}}}{g_S} \text{ is increasing (Equation (3.1))}\right)$$
$$= \sqrt{\frac{k+1}{k+2}}\,z_S + \frac{1}{2\pi}\left(-\int_{1-\frac{1}{2a}}^{1}\sqrt{\mathcal{I}(g_S)}\mathrm{d}t + \int_{1-\frac{1}{2a}}^{1}|W|\mathrm{d}t\right)$$

Now by using Lemma 8 with $\epsilon = \frac{1}{2a}$, Lemma 9 and the inequality $\frac{\pi}{4}x < \arctan(x)$ for $0 < x < 1$, we have

$$
\begin{aligned}
z_{S\cup\{a\}} &\geq \sqrt{\frac{k+1}{k+2}}\, z_S + \frac{1}{2\pi}\int_{1-\frac{1}{2a}}^{1} |W|\,\mathrm{d}t - z_S^{(1-\frac{1}{2a},1)} \\
&\geq \sqrt{\frac{k+1}{k+2}}\, z_S + \frac{1}{\pi}\arctan\left(\frac{1}{4\sqrt{k}}\right) - \frac{\sqrt{k+1}}{2\pi 2^{2^{k-1}}} \\
&\geq \sqrt{\frac{k+1}{k+2}}\, z_S + \frac{1}{\sqrt{k}}\left(\frac{1}{16} - \frac{\sqrt{k}\sqrt{k+1}}{2\pi 2^{2^{k-1}}}\right) \\
&\geq \sqrt{\frac{k+1}{k+2}}\, z_S + \frac{\pi-\sqrt{3}}{16\pi}\frac{1}{\sqrt{k}.} \qquad \text{(assuming } k \geq 3\text{)} \\
&\geq \sqrt{\frac{2}{k+2}}\cdot z_{\{0,1\}} + \frac{\pi-\sqrt{3}}{16\pi}\left(\sum_{j=0}^{k-1}\frac{1}{\sqrt{k-j}}\cdot\sqrt{\frac{k+2-j}{k+2}}\right) \\
&\qquad \text{(iterating k-1 times)} \\
&\geq \sqrt{\frac{2}{k+2}}\cdot z_{\{0,1\}} + \frac{\pi-\sqrt{3}}{16\pi}\left(\sum_{j=0}^{k-1}\frac{1}{\sqrt{k+2-j}}\cdot\sqrt{\frac{k+2-j}{k+2}}\right) \\
&\geq \frac{\pi-\sqrt{3}}{16\pi}\sqrt{k} + \frac{1}{7} \qquad \left(\text{using } z_{\{0,1\}} = \tfrac{1}{4}\right) \quad \square
\end{aligned}
$$

6 CONCLUSION

We settle the bound on the expected number of real zeros of a random k-sparse polynomial when the coefficients are independent standard normal random variables. We first showed an $O(\sqrt{k})$ upper bound for an arbitrary set of size k, and then gave an example of set where this bound is tight. We see this as another step towards understanding the number of real zeros of sparse polynomials and related generalizations.

In this article, we considered random variables following independent standard normal distributions. It would be interesting to study other distributions on the coefficients, although we expect the analysis to become increasingly difficult as the distributions become more complex.

We also mentioned how the real τ-conjecture is connected to the problem we study and its importance in algebraic complexity. Towards resolving the conjecture, consider the simple setting where f and g are both k-sparse polynomials and we wish to study the number of real zeros of $fg + 1$. This is essentially the first case which is non-trivial, unfortunately very little is known and prior techniques seem to fail so far.

Also, there is a vast number of restricted arithmetic circuit models. We invite experts to consider the number of real zeros of univariate polynomials under such restrictions and explore their connections with complexity theoretic lower bounds. It is conceivable that one can find a restriction for which the behavior of the expected number of real zeros is easier to understand than the sparse case and which may lead to new insights towards resolving the aforementioned generalizations, such as the ones considered in the real τ-conjecture.

ACKNOWLEDGMENTS

We thank our advisor Markus Bläser for his constant support throughout the work. We thank Vladimir Lysikov for many insightful discussions. AP thanks Sébastien Tavenas for hosting him at Université Savoie Mont Blanc and for encouraging discussions there.

REFERENCES

[1] Frédéric Bihan and Frank Sottile. 2007. New fewnomial upper bounds from Gale dual polynomial systems. *Mosc. Math. J.* 7, 3 (2007), 387–407, 573. https://doi.org/10.17323/1609-4514-2007-7-3-387-407

[2] Lenore Blum, Felipe Cucker, Michael Shub, and Steve Smale. 1998. *Complexity and real computation.* Springer-Verlag, New York. xvi+453 pages. https://doi.org/10.1007/978-1-4612-0701-6 With a foreword by Richard M. Karp.

[3] Irénée Briquel and Peter Bürgisser. 2020. The real tau-conjecture is true on average. *Random Structures & Algorithms* (2020). https://doi.org/10.1002/rsa.20926 arXiv:https://onlinelibrary.wiley.com/doi/pdf/10.1002/rsa.20926

[4] Peter Bürgisser. 2009. On defining integers and proving arithmetic circuit lower bounds. *Comput. Complexity* 18, 1 (2009), 81–103. https://doi.org/10.1007/s00037-009-0260-x

[5] Peter Bürgisser, Ergür Alperen A., and Josué Tonelli-Cueto. 2019. On the Number of Real Zeros of Random Fewnomials. *SIAM Journal on Applied Algebra and Geometry* 3, 4 (2019), 721–732.

[6] René Descartes. 1886. *La géométrie.* Hermann.

[7] Alan Edelman and Eric Kostlan. 1995. How many zeros of a random polynomial are real? *Bull. Amer. Math. Soc. (N.S.)* 32, 1 (1995), 1–37. https://doi.org/10.1090/S0273-0979-1995-00571-9

[8] Paul Erdös and A. C. Offord. 1956. On the number of real roots of a random algebraic equation. *Proc. London Math. Soc. (3)* 6 (1956), 139–160. https://doi.org/10.1112/plms/s3-6.1.139

[9] Boulos El Hilany. 2016. *Géométrie Tropicale et Systèmes Polynomiaux.* Ph.D. Dissertation. LAMA, Universite Savoie Mont Blanc et de Universite Grenoble Alpes.

[10] A. G. Hovanskiĭ. 1980. A class of systems of transcendental equations. *Dokl. Akad. Nauk SSSR* 255, 4 (1980), 804–807.

[11] Pavel Hrubes. 2013. On the Real τ-Conjecture and the Distribution of Complex Roots. *Theory of Computing* 9 (2013), 403–411. https://doi.org/10.4086/toc.2013.v009a010

[12] M. Kac. 1943. On the average number of real roots of a random algebraic equation. *Bull. Amer. Math. Soc.* 49 (1943), 314–320. https://doi.org/10.1090/S0002-9904-1943-07912-8

[13] A. G. Khovanskiĭ. 1991. *Fewnomials.* Translations of Mathematical Monographs, Vol. 88. American Mathematical Society, Providence, RI. viii+139 pages. Translated from the Russian by Smilka Zdravkovska.

[14] Pascal Koiran. 2011. Shallow circuits with high-powered inputs. In *Innovations in Computer Science - ICS 2010, Tsinghua University, Beijing, China, January 7-9, 2011. Proceedings.* 309–320. http://conference.iiis.tsinghua.edu.cn/ICS2011/content/papers/5.html

[15] Pascal Koiran, Natacha Portier, and Sébastien Tavenas. 2015. A Wronskian approach to the real τ-conjecture. *J. Symb. Comput.* 68 (2015), 195–214. https://doi.org/10.1016/j.jsc.2014.09.036

[16] Pascal Koiran, Natacha Portier, Sébastien Tavenas, and Stéphan Thomassé. 2015. A τ-Conjecture for Newton Polygons. *Foundations of Computational Mathematics* 15, 1 (2015), 185–197. https://doi.org/10.1007/s10208-014-9216-x

[17] A. Kushnirenko. 26 February 2008. Letter to Frank Sottile. www.math.tamu.edu/~sottile/research/pdf/kushnirenko.pdf.

[18] J. E. Littlewood and A. C. Offord. 1938. On the Number of Real Roots of a Random Algebraic Equation. *J. London Math. Soc.* 13, 4 (1938), 288–295. https://doi.org/10.1112/jlms/s1-13.4.288

[19] Gregorio Malajovich and J. Maurice Rojas. 2004. High probability analysis of the condition number of sparse polynomial systems. *Theoret. Comput. Sci.* 315, 2-3 (2004), 524–555. https://doi.org/10.1016/j.tcs.2004.01.006

[20] J. Maurice Rojas. 1996. On the average number of real roots of certain random sparse polynomial systems. In *The mathematics of numerical analysis (Park City, UT, 1995).* Lectures in Appl. Math., Vol. 32. Amer. Math. Soc., Providence, RI, 689–699.

[21] Ramprasad Saptharishi. 2015. A survey of lower bounds in arithmetic circuit complexity. *Github survey* (2015).

[22] Amir Shpilka and Amir Yehudayoff. 2010. Arithmetic Circuits: A survey of recent results and open questions. *Foundations and Trends in Theoretical Computer Science* 5, 3-4 (2010), 207–388. https://doi.org/10.1561/0400000039

[23] Michael Shub and Steve Smale. 1995. On the intractability of Hilbert's Nullstellensatz and an algebraic version of "NP ≠ P?". *Duke Math. J.* 81 (1995), 47–54 (1996). https://doi.org/10.1215/S0012-7094-95-08105-8 A celebration of John F. Nash, Jr.

[24] Frank Sottile. 2011. *Real solutions to equations from geometry.* University Lecture Series, Vol. 57. American Mathematical Society, Providence, RI. x+200 pages. https://doi.org/10.1090/ulect/057

On the Geometry and the Topology of Parametric Curves

Christina Katsamaki *
christina.katsamaki@inria.fr

Fabrice Rouillier *
Fabrice.Rouillier@inria.fr

Elias Tsigaridas
elias.tsigaridas@inria.fr
* INRIA Paris
Sorbonne Université and Paris Université
F-75005, Paris, France

Zafeirakis Zafeirakopoulos
zafeirakopoulos@gtu.edu.tr
Institute of Information Technologies
Gebze Technical University, Turkey

ABSTRACT

We consider the problem of computing the topology and describing the geometry of a parametric curve in $\mathbb{R}^n$. We present an algorithm, PTOPO, that constructs an abstract graph that is isotopic to the curve in the embedding space. Our method exploits the benefits of the parametric representation and does not resort to implicitization.

Most importantly, we perform all computations in the parameter space and not in the implicit space. When the parametrization involves polynomials of degree at most d and maximum bitsize of coefficients τ, then the worst case bit complexity of PTOPO is $\widetilde{O}_B(nd^6 + nd^5\tau + d^4(n^2 + n\tau) + d^3(n^2\tau + n^3) + n^3d^2\tau)$. This bound matches the current record bound $\widetilde{O}_B(d^6 + d^5\tau)$ for the problem of computing the topology of a planar algebraic curve given in implicit form. For planar and space curves, if $N = \max\{d, \tau\}$, the complexity of PTOPO becomes $\widetilde{O}_B(N^6)$, which improves the state-of-the-art result, due to Alcázar and Díaz-Toca [CAGD'10], by a factor of N^{10}. However, visualizing the curve on top of the abstract graph construction, increases the bound to $\widetilde{O}_B(N^7)$. We have implemented PTOPO in MAPLE for the case of planar curves. Our experiments illustrate its practical nature.

CCS CONCEPTS

• **Computing methodologies** → **Symbolic and algebraic algorithms**; • **Mathematics of computing** → **Computations on polynomials**.

KEYWORDS

Parametric curve, topology, bit complexity, polynomial systems

ACM Reference Format:
Christina Katsamaki, Fabrice Rouillier, Elias Tsigaridas, and Zafeirakis Zafeirakopoulos. 2020. On the Geometry and the Topology of Parametric Curves. In *International Symposium on Symbolic and Algebraic Computation (ISSAC '20), July 20–23, 2020, Kalamata, Greece.* ACM, New York, NY, USA, 8 pages. https://doi.org/10.1145/3373207.3404062

ISSAC '20, July 20–23, 2020, Kalamata, Greece

ACM ISBN 978-1-4503-7100-1/20/07…$15.00
https://doi.org/10.1145/3373207.3404062

1 INTRODUCTION

Parametric curves constitute a classical and important topic in computational algebra and geometry [37] that constantly receives attention, e.g., [11, 13, 34, 38]. The interest in efficient algorithms for computing with parametric curves has been motivated, among others, by the omnipresence of parametric representations in computer modeling and computer aided geometric design, e.g., [16].

We focus on computing the topology of a real parametric curve, that is, the computation of an abstract graph that is isotopic [7, p. 184] to the curve in the embedding space. We design a complete algorithm, PTOPO, that applies directly to parametric curves of any dimension. We consider different characteristics of the parametrization, like properness and normality, before computing the singularities and other interesting points on the curve. These points are necessary for representing the geometry of the curve, as well as for producing a certified visualization of planar and space curves.

Previous work. A common strategy when dealing with parametric curves is implicitization. There has been great research effort, e.g., [11, 35] and references therein, in designing algorithms to compute the implicit equations describing the curve. However, it is also important to manipulate parametric curves directly, without converting them to implicit form.

The study of the topology of a real parametric curve is a topic that has not received much attention in the literature, in contrast to its implicit counterpart [14, 22]. It requires special treatment, since for instance it is not always easy to choose a parameter interval such that when we plot the curve over it, we include all the important topological features [3]. Moreover, while visualizing the curve using symbolic computational tools, the problem of missing points and branches may arise [31, 36]. Alcázar and Díaz-Toca [3] study the topology of real parametric curves without implicitizing. They work directly with the parametrization and address both planar and space real rational curves. Our algorithm to compute the topology is to be juxtaposed to their work; we refer to the next paragraph for more details. We also refer to [12] and [2] for other approaches based on computations by values and subdivision, respectively.

To compute the topology of a curve it is essential to detect its singularities. This is an important and well studied problem [3, 22, 32] of independent interest. Apart from classical approaches [17, 41] that work in the implicit representation, we can also compute the singularities using directly the parametrization. For instance, necessary and sufficient conditions to identify cusps and inflection points are expressed in the form of determinants, e.g., [23, 26].

On computing the singularities of a parametric curve, a line of work related to our approach, does so by means of a univariate resultant [1, 19, 28, 30, 32]. Notably in [32] the authors work on rational parametric curves in affine n-space; they use generalized resultants to find the parameters of the singular points. Moreover, they characterize the singularities and compute their multiplicities.

Cox [13] uses the syzygies of the ideal generated by the polynomials that give the parameterization to compute the singularities and their structure. There are state-of-the-art approaches that exploit this idea and relate the problem of computing the singularities with the notion of the μ-basis of the parametrization, e.g., [21] and references therein. Another method is used in [6], where they compute and characterize the singularities using factorization of resultants. In [5] they use the projection from the rational normal curve to the curve and its relation with the secant varieties to the normal curve.

Overview of our contributions. We introduce PTOPO, a complete, exact, and efficient algorithm (Alg. 2) for computing the geometric properties and the topology of parametric curves in $\mathbb{R}^n$. Unlike other algorithms, e.g. [3], it makes no assumptions on the input curves, such as the absence of axis-parallel asymptotes, and it does not perform any projections and liftings when $n \geq 2$. For this, it is applicable to any dimension. Nevertheless, it does not handle knots for space curves.

If the (proper) parametrization of the curve consists of polynomials of degree d and bitsize τ, then PTOPO outputs a graph isotopic [7, p.184] to the curve in the embedding space, by performing

$$\widetilde{O}_B(nd^6 + nd^5\tau + d^4(n^2 + n\tau) + d^3(n^2\tau + n^3) + n^3d^2\tau)$$

bit operations in the worst case (Thm. 5.5). We also provide a Las Vegas variant with expected complexity

$$\widetilde{O}_B(d^6 + d^5(n + \tau) + d^4(n^2 + n\tau) + d^3(n^2\tau + n^3) + n^3d^2\tau).$$

If $n = O(1)$, the bounds become $\widetilde{O}_B(N^6)$, where $N = \max\{d, \tau\}$. The vertices of the output graph correspond to special points on the curve, in whose neighborhood the topology is not trivial, given by their parameter values. Each edge of the graph is associated with two parameter values and corresponds to a unique smooth parametric arc. For an embedding isotopic to the curve, we map every edge of the abstract graph to the corresponding parametric arc.

For planar and space curves, our bound improves the previously known one due to Alcázar and Díaz-Toca [3] by a factor of $\widetilde{O}_B(N^{10})$. The latter algorithm [3] performs computations in the implicit space. On the contrary, PTOPO is a fundamentally different approach since we work exclusively in the parameter space; we do not use a sweep-line algorithm to construct the isotopic graph. We handle only the parameters that give important points on the curve and thus we avoid performing operations such as univariate root isolation in an extension field or evaluation of a polynomial at an algebraic number.

Computing singular points is an essential part of PTOPO (Lem. 4.7). We chose not to exploit recent methods, e.g., [6], for this task, but to employ older techniques, e.g., [3, 30, 32], that rely on a bivariate polynomial system, Eq. (2). We take advantage of this system's symmetry and of nearly optimal algorithms for bivariate system solving and for computations with real algebraic numbers [8, 14, 15, 27]. In particular, we introduce an algorithm for isolating the roots of over-determined bivariate polynomial systems by exploiting the Rational Univariate Representation (RUR) [8–10] that has worst case and expected bit complexity that matches the ones for square systems (Thm. 4.6). These are definitive steps for obtaining the complexity bounds of Thm. 5.4 and Thm. 5.5.

Moreover, our bound matches the current state-of-the-art complexity bound, $\widetilde{O}_B(d^6 + d^5\tau)$ or $\widetilde{O}_B(N^6)$, for computing the topology of implicit plane curves [14, 22]. However, if we want to visualize the graph in 2D or 3D, then we have to compute a characteristic box (Lem. 5.1) that contains all the the topological features of the curve and the intersections of the curve with its boundary. In this case, the complexity of PTOPO becomes $\widetilde{O}_B(N^7)$ (Thm. 5.4).

A preprocessing step of PTOPO consists in finding a proper reparametrization of the curve (if it is not proper). We present explicit bit complexity bounds (Lem. 3.2) for the algorithm of Pérez [29] to compute a proper parametrization. Another preprocessing step is to ensure that there are no singularities at infinity; Lem. 3.3 handles this task and provides explicit complexity estimates.

Last but not least, we provide a certified implementation[1] of PTOPO in MAPLE. So far, the implementation handles the topology computation and visualization of planar curves. We also provide an amplified adaptation[2] where omitted proofs can be found.

2 NOTATION AND ALGEBRAIC TOOLS

For a polynomial $f \in \mathbb{Z}[x]$, its infinity norm is equal to the maximum absolute value of its coefficients. We denote by $\mathcal{L}(f)$ the logarithm of its infinity norm. We also call the latter the bitsize of the polynomial. A univariate polynomial is of size (d, τ) when its degree is at most d and has bitsize τ. The bitsize of a rational function is the maximum of the bitsizes of the numerator and the denominator. We represent an algebraic number $\alpha \in \mathbb{C}$ by the *isolating interval representation*. When $\alpha \in \mathbb{R}$ (resp. $\mathbb{C}$), it includes a square-free polynomial which vanishes at α and a (rational) interval (resp. Cartesian products of intervals) containing α and no other root of this polynomial. We denote by O, resp. O_B, the arithmetic, resp. bit, complexity and we use $\widetilde{O}$, resp. $\widetilde{O}_B$, to ignore (poly-)logarithmic factors. We denote by $\mathtt{res}_x(f, g)$ the resultant of the polynomials f, g with respect to x. For $t \in \mathbb{C}$, we denote by $\bar{t}$ its complex conjugate. We use $[n]$ to signify the set $\{1, \ldots, n\}$.

We present some results needed for our analysis. Following [39], Lem. 2.1 is an application of Cauchy's bound for polynomials.

LEMMA 2.1. *Let $A = \sum_{i=0}^{m} a_i X^i, B = \sum_{i=0}^{n} b_i X^i \in \mathbb{Z}[X]$ of degrees m and n and of bitsizes τ and σ respectively. Let $\alpha_1, \ldots, \alpha_m$ be the complex roots of A, counting multiplicities. Then, for any $\kappa = 1, \ldots, m$ it holds that*

$$2^{-m\sigma - n\tau - (m+n)\log(m+n)} < |B(\alpha_\kappa)| < 2^{m\sigma + n\tau + (m+n)\log(m+n)}.$$

Lemmata 2.2, 2.3 restate known results on the gcd computation of various univariate and bivariate polynomials obtained by combining [10, Lem.4], [8, Lem. 5], [40, Thm. 6.46, Ex.10.21, Cor. 11.12] and [33].

[1] https://webusers.imj-prg.fr/~christina.katsamaki/ptopo/
[2] https://hal.archives-ouvertes.fr/hal-02573423

Lemma 2.2. *Let $f_1(X), \ldots, f_n(X) \in \mathbb{Z}[X]$ of sizes (δ, L). We can compute their* gcd *in worst case complexity $\widetilde{O}_B(n(\delta^3 + \delta^2 L))$, or with a Monte Carlo algorithm in $\widetilde{O}_B(\delta^2 + \delta L)$, or with a Las Vegas algorithm in $\widetilde{O}_B(n(\delta^2 + \delta L))$.*

Lemma 2.3. *Let $f_1(X, Y), \ldots, f_n(X, Y) \in \mathbb{Z}[X, Y]$ of bidegrees (δ, δ) and $\mathcal{L}(f_i) = L$. We can compute their* gcd *in worst case complexity $\widetilde{O}_B(n(\delta^5 + \delta^4 L))$, or with a Monte Carlo algorithm in $\widetilde{O}_B(\delta^3 + \delta^2 L)$, or with a Las Vegas algorithm in $\widetilde{O}_B(n(\delta^3 + \delta^2 L))$.*

3 RATIONAL CURVES

Following closely [3], we introduce basic notions for rational curves. Let $\widetilde{C}$ be an algebraic curve over $\mathbb{C}^n$, parametrized by the map

$$\phi: \quad \mathbb{C} \dashrightarrow \widetilde{C}$$
$$t \mapsto \big(\phi_1(t), \ldots, \phi_n(t)\big) = \Big(\frac{p_1(t)}{q_1(t)}, \ldots, \frac{p_n(t)}{q_n(t)}\Big), \qquad (1)$$

where $p_i, q_i \in \mathbb{Z}[t]$ are of size (d, τ) for $i \in [n]$, and $\widetilde{C}$ is the Zariski closure of $\mathrm{Im}(\phi)$. We call $\phi(t)$ a *parametrization* of $\widetilde{C}$.

We study the real trace of $\tilde{C}$, that is $C := \tilde{C} \cap \mathbb{R}^n$. A parametrization ϕ is chatacterized by means of *properness* (Sec. 3.1) and *normality* (Sec. 3.2). To ensure these properties, one can reparametrize the curve, i.e., apply a rational change of parameter to the given parametrization. We refer to [38, Ch. 6] for more details on reparameterization.

Without loss of generality, we assume that no component of the parametrization ϕ is constant; otherwise we could embed $\tilde{C}$ in a lower dimensional space. We consider that ϕ is in *reduced form*, i.e., $\gcd(p_i(t), q_i(t)) = 1$, for all $i \in [n]$. The point at infinity, $\mathbf{p}_\infty$, is the point on C we obtain for $t \to \pm\infty$ (if it exists). For a parametrization ϕ, we consider the following system of bivariate polynomials:

$$h_i(s, t) = \frac{p_i(s)q_i(t) - q_i(s)p_i(t)}{s - t}, \quad \text{for } i \in [n]. \qquad (2)$$

Remark 1. *The h_i's are polynomials since (s, s) is a root of the numerator for every s. Also, $h_i(t, t) = \phi_i'(t)q_i^2(t)$ for $i \in [n]$ [20, Lem. 1.7].*

3.1 Proper parametrization

A parametrization is proper if $\phi(t)$ is injective for almost all points on $\widetilde{C}$. In other words, almost every point on $\widetilde{C}$ is the image of exactly one parameter value (real or complex). For other equivalent definitions of properness we refer to [38, Ch. 4], [32]. The following condition [3, Thm. 1] leads to an algorithm for checking properness: a parametrization is proper if and only if $\deg(\gcd(h_1(s, t), \ldots, h_n(s, t))) = 0$. By applying Lem. 2.3 we get the following:

Lemma 3.1. *There is an algorithm that checks if a parametrization ϕ is proper in worst-case bit complexity $\widetilde{O}_B(n(d^5 + d^4\tau))$ and in expected bit complexity $\widetilde{O}_B(n(d^3 + d^2\tau))$.*

If ϕ is a not a proper parametrization, then there always exists a parametrization $\psi \in \mathbb{Z}(t)^n$ and $R(t) \in \mathbb{Z}(t)$ such that $\psi(R(t)) = \phi(t)$ and ψ is proper [38, Thm. 7.6]. There are various algorithms for obtaining a proper parametrization, e.g., [18, 19, 29, 34, 38]. We consider the algorithm in [29] for its simplicity; its pseudo-code is in Alg. 1.

Lemma 3.2. *Consider a non-proper parametrization of a curve C, consisting of univariate polynomials of size (d, τ). Alg. 1 computes a proper parametrization of C, involving polynomials of degree at most d and bitsize $O(d^2 + d\tau)$, in $\widetilde{O}_B(n(d^5 + d^4\tau))$, in the worst case.*

Proof. The algorithm first computes the bivariate polynomials $H_1, \ldots H_n$. They have bi-degree at most (d, d) and bitsize at most $2\tau + 1$. Then, we compute their gcd, that we denote by H, in $\widetilde{O}_B(n(d^5 + d^4\tau))$ (Lem. 2.3). By [25] and [4, Prop. 10.12] we have that $\mathcal{L}(H) = O(d + \tau)$, which is also the case for $C_j(s)$.

If the degree of H is one, then the parametrization is already proper and we have nothing to do. Otherwise, we consider H as a univariate polynomial in s and we find two of its coefficients that are relatively prime, using exact division. The complexity of this operation is $m^2 \cdot \widetilde{O}_B(d^2 + d\tau) = \widetilde{O}_B(d^4 + d^3\tau)$ [40, Ex. 10.21].

Subsequently, we perform n resultant computations to get $L_1, \ldots L_n$. From these we obtain the rational functions of the new parametrization. We focus on the computation of L_1. The same arguments hold for all L_i. The bi-degree of $L_1(s, x)$ is (d, d) [4, Prop. 8.49] and $\mathcal{L}(L_1) = O(d^2 + d\tau)$ [4, Prop. 8.50]; the latter dictates the bitsize of the new parametrization.

To compute L_1, we consider F_1 and G as univariate polynomials in t and we apply a fast algorithm for computing the univariate resultant based on subresultants [24]; it performs $\widetilde{O}(d)$ operations. Each operation consists of multiplying bivariate polynomials of bi-degree (d, d) and bitsize $O(d^2 + d\tau)$; so it costs $\widetilde{O}_B(d^4 + d^3\tau)$. We compute the resultant in $\widetilde{O}_B(d^5 + d^4\tau)$. We multiply the latter bound by n to conclude the proof. □

Algorithm 1: `Make_Proper`(ϕ)

Input: A parametrization $\phi \in \mathbb{Z}(t)^n$ as in Eq. (1)
Output: A proper parametrization $\psi = (\psi_1, \ldots, \psi_n) \in \mathbb{Z}(t)^n$

for $i \in [n]$ **do** $H_i(s, t) \leftarrow p_i(s)q_i(t) - p_i(t)q_i(s) \in \mathbb{Z}[s, t]$;
$H \leftarrow \gcd(H_1, \ldots, H_n) = C_m(t)s^m + \cdots + C_0(t) \in (\mathbb{Z}[t])[s]$
if $m = 1$ **then** RETURN $\phi(t)$;
Find $k, l \in [m]$ such that:
  $\deg(\gcd(C_k(t), C_l(t))) = 0$ and $\frac{C_k(t)}{C_l(t)} \notin \mathbb{Q}$
$R(t) \leftarrow \frac{C_k(t)}{C_l(t)}$
$r \leftarrow \deg(R) = \max\{\deg(C_k), \deg(C_l)\}$
$G \leftarrow s\,C_l(t) - C_k(t)$
for $i \in [n]$ **do**
  $F_i \leftarrow xq_i(t) - p_i(t)$
  $L_i(s, x) \leftarrow \mathtt{res}_t(F_i(t, x), G(t, s)) = (\tilde{q}_i(s)x - \tilde{p}_i(s))^r$
RETURN $\psi(t) = \big(\frac{\tilde{p}_1(t)}{\tilde{q}_1(t)}, \ldots, \frac{\tilde{p}_n(t)}{\tilde{q}_n(t)}\big)$

3.2 Normal parametrization

Normality of the parametrization concerns the surjectivity of the map ϕ. The parametrization $\phi(t)$ is $\mathbb{R}$-normal if for all points $\mathbf{p}$ on C there exists $t_0 \in \mathbb{R}$ such that $\phi(t_0) = \mathbf{p}$. When the parametrization is not $\mathbb{R}$-normal, the points that are not in the image of ϕ for $t \in \mathbb{R}$ are $\mathbf{p}_\infty$ (if it exists) and the isolated points that we obtain for complex values of t [31, Prop. 4.2]. An $\mathbb{R}$-normal reparametrization does not always exist. We refer to [38, Sect. 7.3] for further details.

However, if $\mathbf{p}_\infty$ exists, then we reparametrize the curve to avoid possible singularities at infinity. The point $\mathbf{p}_\infty$ exists if $\deg(p_i) \leq \deg(q_i)$, for all $i \in [n]$.

Lemma 3.3. *If $\mathbf{p}_\infty$ exists, then we can reparametrize the curve with a linear function to ensure that $\mathbf{p}_\infty$ is not a singular point, using a Las Vegas algorithm in expected time $\widetilde{O}_B(n(d^2 + d\tau))$. The new parametrization involves polynomials of size $(d, \widetilde{O}(d + \tau))$.*

Proof Sketch. We choose t_0 uniformly at random from a large enough set of integers and we reparametrize as $t \mapsto \frac{t_0 t+1}{t-t_0}$. With good probability $\phi(t_0)$ is not singular, so the point at infinity of the new parametrization is not singular. For a Las Vegas algorithm, we ensure that $\phi(t_0)$ is neither a cusp, i.e., $\phi'(t_0) \neq 0$, nor a multiple point, i.e., $\deg(\gcd(\phi_1(t_0)q_1(t) - p_1(t), \ldots, \phi_1(t_0)q_1(t) - p_1(t))) = 0$ (Lem. 2.2).

Remark 2. *Since the reparametrizing function in the previous lemma is linear, it does not affect properness [38, Thm. 6.3].*

4 SPECIAL POINTS ON THE CURVE

We consider a parametrization ϕ of C as in Eq. (1), such that ϕ is proper and there are no singularities at infinity. We highlight the necessity of these assumptions when needed. We detect the *parameters* that generate the *special points* of C, namely the singular, the isolated, and the extreme points. We identify the values of the parameter for which ϕ is not defined, namely the poles (see Def. 1); in presence of poles, C consists of multiple components.

Definition 1. *The parameters for which $\phi(t)$ is not defined are the poles of ϕ. The sets of poles over the complex and the reals are:*

$$\mathsf{T}_P^{\mathbb{C}} = \{t \in \mathbb{C} : \prod_{i\in[n]} q_i(t) = 0\} \text{ and } \mathsf{T}_P^{\mathbb{R}} = \mathsf{T}_P^{\mathbb{C}} \cap \mathbb{R}.$$

We consider the solution set S of system (2) over $\mathbb{C}^2$:

$$S = \{(s,t) \in \mathbb{C}^2 : h_i(s,t) = 0 \text{ for all } i \in [n]\}.$$

Remark 3. *Notice that when ϕ is in reduced form, if $(s,t) \in S$ and $(s,t) \in (\mathbb{C} \setminus \mathsf{T}_P^{\mathbb{C}}) \times \mathbb{C}$, then also $t \notin \mathsf{T}_P^{\mathbb{C}}$ [32, (in the proof of) Lem. 9].*

Next, we present some well known results [32, 38] that we adapt in our notation.

Singular points. Quoting [26], "Algebraically, singular points are points on the curve, in whose neighborhood the curve cannot be represented as an one-to-one and C^∞ bijective map with an open interval on the real line". Geometrically, singularities correspond to shape features that are known as cusps and self-intersections of smooth branches. *Cusps* are points on the curve where the tangent vector is the zero vector. This is a necessary and sufficient condition when the parametrization is proper [26]. Self-intersections are *multiple* points, i.e., points on C with more than one preimages.

Lemma 4.1. *The set of parameters corresponding to real cusps is*

$$\mathsf{T}_C = \left\{t \in \mathbb{R} \setminus \mathsf{T}_P^{\mathbb{R}} : (t,t) \in S\right\}.$$

The set of parameters corresponding to real multiple points is

$$\mathsf{T}_M = \{t \in \mathbb{R} \setminus \mathsf{T}_P^{\mathbb{R}} : \exists s \neq t, s \in \mathbb{R} \text{ such that } (s,t) \in S\}.$$

Notice that T_C and T_M are not necessarily disjoint, for at the same point we may have both cusps and smooth branches that intersect.

Isolated points. An isolated point on a real curve can only occur for complex values of the parameter. Then, the same point is also obtained by the conjugate of the parameter, therefore it is a multiple point. The point at infinity is not isolated because it is the limit of a sequence of real points.

Lemma 4.2. *The set of parameters generating isolated points of C is*

$$\mathsf{T}_I = \{t \in \mathbb{C} \setminus (\mathbb{R} \cup \mathsf{T}_P^{\mathbb{C}}) : (t, \bar{t}) \in S \text{ and } \nexists s \in \mathbb{R} \text{ s.t. } (s,t) \in S\}.$$

Extreme points. Consider a vector $\vec{\delta}$ and a point on C whose tangent vector is parallel to $\vec{\delta}$. If the point is not singular, then it is an extreme point of C with respect to $\vec{\delta}$. We compute the extreme points with respect to the direction of each coordinate axis. Rem. 1 leads to the following lemma:

Lemma 4.3. *The set of parameters generating extreme points is*

$$\mathsf{T}_E = \left\{t \in \mathbb{R} \setminus \mathsf{T}_P^{\mathbb{R}} : \prod_{i\in[n]} h_i(t,t) = 0 \text{ and } t \notin \mathsf{T}_C \cup \mathsf{T}_M\right\}.$$

4.1 Computation and Complexity

From Lemmata 4.1, 4.2, and 4.3, it follows that given a proper parametrization ϕ without singular points at infinity, we can easily find the poles and the set of parameters generating cusps, multiple, extreme, and isolated points. We do so, by solving an overdetermined bivariate polynomial system and univariate polynomial equations. Then, we classify the parameters that appear in the solutions, by exploiting the fact the system is symmetric.

To compute the RUR [8–10] of an overdetermined bivariate system (Thm. 4.6), we employ Lem. 4.4 and Prop. 4.5, which adapt the techniques used in [8] to our setting.

Lemma 4.4. *Let $f, g_0, g_1, \ldots, g_N \in \mathbb{Z}[X,Y]$ with degrees bounded by δ and bitsize of coefficients bounded by L. Computing a common separating element in the form $X + \alpha Y, \alpha \in \mathbb{Z}$ for the N+1 systems of bivariate polynomial equations $\{f = g_0 = 0\}$, $\{f = g_i = 0\}$, $i \in [n]$, needs $\widetilde{O}_B(N(\delta^6 + \delta^5 L))$ bit operations in the worst case, and $\widetilde{O}_B(N(\delta^5 + \delta^4 L))$ in the expected case with a Las Vegas Algorithm. Moreover, the bitsize of α does not exceed $\log(2N\delta^4)$.*

Proposition 4.5. *Let $f, g \in \mathbb{Z}[X,Y]$ with degrees bounded by δ and coefficients' bitsizes bounded by L. We can compute a rational parameterization $\{r(T), X = \frac{r_X(T)}{r_I(T)}, Y = \frac{r_Y(T)}{r_I(T)}\}$ of the system $\{f = g = 0\}$ with $r, r_I, r_X, r_Y \in \mathbb{Z}[T]$ with degrees less than δ^2 and coefficients' bitsizes in $\widetilde{O}(\delta(L + \delta))$, in $\widetilde{O}_B(\delta^5(L + \delta))$ bit operations in the worst case and $\widetilde{O}_B(\delta^4(L + \delta))$ expected bit operations with a Las Vegas Algorithm.*

Theorem 4.6. *There exists an algorithm that computes the RUR and the isolating boxes of the roots of the system $\{h_1(s,t) = \cdots = h_n(s,t) = 0\}$ with worst-case bit complexity $\widetilde{O}_B(n(d^6 + d^5\tau))$. There is also a Las Vegas variant with expected complexity $\widetilde{O}_B(d^6 + nd^5 + d^5\tau + nd^4\tau)$.*

Proof. Assume that we know a common separating linear element $\ell(s,t)$ that separates the roots of the n-1 systems of bivariate polynomial equations $\{h_1 = h_2 = 0\}$, $\{h_1 = h_i = 0\}$, for $3 \leq i \leq n$. We can compute ℓ with $\widetilde{O}_B(n(d^6 + d^5\tau))$ bit operations

in the worst case and with $\widetilde{O}_B(n(d^5+d^4\tau))$ expected bit operations with a Las Vegas algorithm (Lem. 4.4).

We denote by $\{r(T), \frac{r_s(T)}{r_I(T)}, \frac{r_t(T)}{r_I(T)}\}$ a RUR for $\{h_1 = h_2 = 0\}$ with respect to ℓ. In addition, for $3 \le i \le n$, let $\{r_i(T), \frac{r_{i,s}(T)}{r_{i,I}(T)}, \frac{r_{i,t}(T)}{r_{i,I}(T)}\}$ be the RUR of $\{h_1 = h_i = 0\}$, also with respect to ℓ. We can compute all these representations with $\widetilde{O}_B(n(d^6+d^5\tau))$ bit operations in the worst case, and with $\widetilde{O}_B(n(d^5+d^4\tau))$ in expected case with a Las Vegas algorithm (Lem. 4.5).

Then, for the system $\{h_1 = h_2 = \ldots = h_n = 0\}$ we can define a rational parameterization $\{\chi(T), \frac{r_s(T)}{r_I(T)}, \frac{r_t(T)}{r_I(T)}\}$, where

$$\begin{aligned}\chi(T) = \gcd(\quad & r(T), r_3(T), \ldots, r_n(T),\\ & r_s(T)r_{3,I}(T) - r_{3,s}(T)r_I(T), r_t(T)r_{3,I}(T) - r_{3,t}(T)r_I(T),\\ & \vdots\\ & r_s(T)r_{n,I}(T) - r_{n,s}(T)r_I(T), r_t(T)r_{n,I}(T) - r_{n,t}(T)r_I(T)).\end{aligned}$$

So, we need to compute the gcd of $3n-5$ univariate polynomials of degrees at most d^2 and coefficients of bitsizes in $\widetilde{O}(d\tau)$. This takes $\widetilde{O}_B(n(d^6+d^5\tau))$ bit operations in the worst case and $\widetilde{O}_B(n(d^4+d^3\tau))$ in the expected case (Lem. 2.2). Isolating the roots of such a parameterization requires $\widetilde{O}_B(d^6+d^5\tau)$ as in Alg. 7 from [8]. □

Remark 4 (RUR and isolating interval representation). *If we use Thm.4.6 to solve the over-determined bivariate system of the h_i polynomials of Eq. (2), then we obtain in the output a RUR for the roots, which is as follows: There is a polynomial $\chi(T) \in \mathbb{Z}[T]$ of size $(O(d^2), \widetilde{O}(d^2+d\tau))$ and a mapping*

$$\begin{aligned} V(\chi) &\to V(h_1, \ldots, h_n)\\ T &\mapsto \left(\frac{r_s(T)}{r_I(T)}, \frac{r_t(T)}{r_I(T)}\right), \end{aligned} \tag{3}$$

that defines a one-to-one correspondence between the roots of χ and those of the system. The polynomials r_s, r_t, and r_I are in $\mathbb{Z}[T]$ and have also size $(O(d^2), \widetilde{O}(d^2+d\tau))$.

Taking into account the cost to compute this parametrization of the solutions (Thm. 4.6), we can also compute at no extra cost the resultant of $\{h_1, h_2\}$ with respect to s or t. Notice that both resultants are the same polynomial, since the system is symmetric. Let $R_s(t) = \mathtt{res}_s(h_1, h_2)$. It is of size $(O(d^2), O(d^2+d\tau))$ [4, Prop. 8.46].

Under the same bit complexity, we can sufficiently refine the isolating boxes of the solutions of the bivariate system (computed in Thm. 4.6), so that every root $(\frac{r_s(\xi)}{r_I(\xi)}, \frac{r_t(\xi)}{r_I(\xi)})$, where $\chi(\xi) = 0$, has a representation as a pair of algebraic numbers in isolating interval representation:

$$((R_s, I_{1,\xi} \times I_{2,\xi}), (R_s, J_{1,\xi} \times J_{2,\xi})). \tag{4}$$

Both coordinates are roots of the same polynomial. Moreover, $I_{2,\xi}, J_{2,\xi}$ are empty sets when the corresponding algebraic number is real. Therefore, we can immediately distinguish between real and complex parameters. At the same time, we associate to each isolating box of a root of R_s the algebraic numbers $\rho = (\chi, I_\rho \times J_\rho)$ for whom it holds that $\frac{r_s(\rho)}{r_I(\rho)}$ projects inside this isolating box. We can interchange between the two of representations in constant time, and this will simplify our computations in the sequel.

Lemma 4.7. *Let C be a curve with a proper parametrization $\phi(t)$ as in (1), that has no singularities at infinity. We compute the real poles of ϕ and the parameters corresponding to singular, extreme, and isolated points of C in worst-case bit complexity*

$$\widetilde{O}_B(nd^6 + nd^5\tau + d^4(n^2+n\tau) + d^3(n^2\tau+n^3) + n^3d^2\tau),$$

and using a Las Vegas algorithm in expected bit complexity

$$\widetilde{O}_B(d^6 + d^5(n+\tau) + d^4(n^2+n\tau) + d^3(n^2\tau+n^3) + n^3d^2\tau).$$

Proof. The proof is an immediate consequence of the following:
• *We compute all $h_i \in \mathbb{Z}[s,t]$ in $\widetilde{O}_B(nd^2\tau)$*: To construct each h_i we perform d^2 multiplications of numbers of bitsize τ; the cost for this is $\widetilde{O}_B(d^2\tau)$. The bi-degree of each is at most (d,d) and $\mathcal{L}(h_i) \le 2\tau + 1 = O(\tau)$.
• *The real poles of ϕ are computed in $\widetilde{O}_B(n^2(d^4+d^3\tau))$:* To find the poles of ϕ, we isolate the real roots of each polynomial $q_i(t)$, for $i \in [n]$. This costs $\widetilde{O}_B(n(d^3+d^2\tau))$ [27]. Then we sort the roots in $\widetilde{O}_B(n\,d\,n(d^3+d^2\tau)) = \widetilde{O}_B(n^2(d^4+d^3\tau))$.
• *The parameters corresponding to cusps, multiple and isolated points of C are computed in $\widetilde{O}_B(n(d^6+d^5\tau))$:*
We solve the bivariate system (2) in $\widetilde{O}_B(n(d^6+d^5\tau))$ or in expected time $\widetilde{O}_B(d^6 + nd^5 + d^5\tau + nd^4\tau)$ (Thm. 4.6). Then we have a parametrization of the solutions of the bivariate system (2) of the form (3) and in the same time of the form (4) (see Rem. 4). Some solutions $(s,t) \in S$ may not correspond to points on the curve, since s, t can be poles of ϕ. Notice that from Rem. 3, s and t are either both poles or none of them is a pole. We compute $g_s = \gcd(R_s, Q)$, where $Q(t) = \prod_{i\in[n]} q_i(t)$, and the gcd-free part of R_s with respect to Q, say R_s^*. This is done in $\widetilde{O}_B(\max\{n,d\}\,(nd^3\tau + nd^2\tau^2))$ [10, Lem. 5]. Every root of R_s^* is an algebraic number of the form $(R_s, I_{1,\xi} \times I_{2,\xi})$, for some ξ that is root of χ. We can easily determine if it corresponds to a cusp, a multiple or an isolated point; when real (i.e., $I_{2,\xi} = \emptyset$) it corresponds to a cusp of C if and only if $((R_s, I_{1,\xi}), (R_s, I_{1,\xi}))$ is in S. Otherwise, it corresponds to a multiple point. When it is complex (i.e., $I_{2,\xi} \neq \emptyset$), it corresponds to an isolated pont of C if and only if $((R_s, I_{1,\xi} \times I_{2,\xi}), (R_s, I_{1,\xi} \times (-I_{2,\xi}))) \in S$ and there is no root in S of the form $((R_s, I_{1,\xi} \times I_{2,\xi}), (R_s, J_{1,\xi'}))$.
• *The parameters corresponding to extreme points of C are computed in $\widetilde{O}_B(d^4n\tau + d^3(n^2\tau+n^3) + d^2n^3\tau)$:*
For all $i \in [n]$, $h_i(t,t)$ is a univariate polynomial of size $(O(d), O(\tau))$. Then, $H(t) = \prod_{i\in[n]} h_i(t,t)$ is of size $(O(nd), \widetilde{O}(n\tau))$. The parameters that correspond to extreme points are among the roots of $H(t)$. To make sure that poles and parameters that give singular points are excluded, we compute $\gcd(H, Q\cdot R_s)$, where $Q(t) = \prod_{i\in[n]} q_i(t)$, and the gcd-free part of H with respect to $Q\cdot R_s$, say H^*. Since $Q\cdot R_s$ is a polynomial of size $(d^2+nd, (d+n)\tau)$, the computation of the gcd and the gcd-free part costs $\widetilde{O}_B(n(d^4\tau + nd^3\tau + n^2d^2\tau))$ [10, Lem. 5]. Then, $H = \gcd(H, Q\cdot R_s)H^*$, and the real roots of H^* give the parameters that correspond to extreme points. We isolate the real roots of H^* in $\widetilde{O}_B(n^3(d^3+d^2\tau))$, since it is a polynomial of size $(O(nd), \widetilde{O}(n(d+\tau)))$ [27]. □

5 PTOPO: TOPOLOGY AND COMPLEXITY

We present PTOPO, an algorithm to construct an abstract graph G that is isotopic [7, p.184] to C when we embed it in $\mathbb{R}^n$. We emphasize that, currently, we do not treat/compute knots in the case of space curves. The embedding consists of a graph whose

vertices are points on the curve given by their parameter values. The edges are smooth parametric arcs that we can continuously deform to branches of C without any topological changes. We need to specify a bounding box in $\mathbb{R}^n$ inside which the constructed graph results in an isotopic embedding to C. We comment at the end of the section on the case where an arbitrary box is provided at the input. We determine a bounding box in $\mathbb{R}^n$, which we call *characteristic*, that captures all the topological information of C:

Definition 2. *A characteristic box of C is a box enclosing a subset of $\mathbb{R}^n$ that intersects all components of C and contains all its singular, extreme, and isolated points.*

Let $\mathcal{B}_C$ be a characteristic box of C. If C is bounded, then $C \subset \mathcal{B}_C$. If C is unbounded, then the branches of C that extend to infinity intersect the boundary of $\mathcal{B}_C$. A branch of the curve extends to infinity if for $t \to t_0$, it holds $||\phi(t)|| > M$, for any $M > 0$, where $t_0 \in \mathbb{R} \cup \{\infty\}$. Lem. 5.1 computes a characteristic box using the degree and bitsize of the polynomials in the parametrization (1).

Lemma 5.1. *Let C be a curve with a parametrization as in (1). For $b = 15d^2(\tau + \log d) = O(d^2\tau)$, $\mathcal{B}_C = [-2^b, 2^b]^n$ is a characteristic box of C.*

Proof. We estimate the maximum and minimum values of ϕ_i, $i \in [n]$, when we evaluate it at the parameter values that correspond to special points and also at each pole that is not a root of q_i.

Let t_0 be a parameter that corresponds to a cusp or an extreme point with respect to the i-th direction. Then, it is a root of $\phi_i'(t)$. Let $N(t) = p_i'(t)q_i(t) - p_i(t)q_i'(t)$ the numerator of $\phi_i'(t)$. Then $N(t_0) = 0$. The degree of $N(t)$ is $\leq 2d - 1$ and $\mathcal{L}(N) \leq 2^{2\tau+\log d+1}$. From Lem. 2.1 we conclude that $|p_i(t_0)| \leq 2^{4d\tau+d\log(d)+(3d-1)\log(3d-1)+d-\tau}$. Analogously, it holds that $|q_i(t_0)| \geq 2^{-4d\tau-d\log(d)-(3d-1)\log(3d-1)-d+\tau}$. Therefore,

$$|\phi_i(t_0)| \leq 2^{2(4d\tau+d\log(d)+(3d-1)\log(3d-1)+d-\tau)}.$$

Now, let (t_1, t_2) be two parameters corresponding to a multiple point of C, i.e., (t_1, t_2) is a root of the bivariate system in Eq. (2). Take any $j, k \in [n]$ with $j \neq k$ and let $R(t) = \mathrm{res}_s(h_j, h_k)$. It holds that $R(t_1) = 0$. The degree of R is $\leq 2d^2$ and $\mathcal{L}(R) \leq 2d(\tau+\log(d)+\log(d+1)+1)$ [4, Prop. 8.29]. Applying Lem. 2.1, we deduce that

$$|\phi_i(t_1)| \leq 2^{4d^2(\tau+\log(d)+\log(d+1)+1)+4d^2\tau+(2d^2+d)\log(2d^2+d)}.$$

Let t_3 be a pole of ϕ with $q_j(t_3) = 0$, for some $j \neq i$. If $\phi_i(t_3)$ is defined, applying Lem. 2.1 gives

$$|\phi_i(t_3)| \leq 2^{4d\tau+4d\log 2d}.$$

To conclude, we take the maximum of the three bounds. However, to simplify notation, we slightly overestimate the latter bound. □

The vertices of the embedded graph must include the singular and the isolated points of C. Additionally, to rigorously visualize the geometry of C, we consider as vertices the extreme points of C, with respect to all coordinate directions, as well as the intersections of C with the boundary of the bounding box. We label the vertices of G using the corresponding parameter values generating these points and we connect them accordingly. Alg. 2 presents the pseudo-code of PTOPO and here we give some more details on the various steps. We construct G as follows:

First, we compute the poles and the sets $\mathsf{T}_C, \mathsf{T}_M, \mathsf{T}_E$, and T_I of parameters corresponding to "special points". Then, we compute the characteristic box of C, say $\mathcal{B}_C$. We compute the set T_B of parameters corresponding to the intersections of C with the boundary of $\mathcal{B}_C$ (if any). Lem. 5.2 describes this procedure and its complexity.

Lemma 5.2. *Let $\mathcal{B} = [l_1, r_1]\times\cdots\times[l_n, r_n]$ in $\mathbb{R}^n$ and $\mathcal{L}(l_i) = \mathcal{L}(r_i) = \sigma$, for $i \in [n]$. We can find the parameters that give the intersection points of ϕ with the boundary of $\mathcal{B}$ in $\widetilde{O}_B(n^2d^3 + n^2d^2(\tau + \sigma))$.*

Proof. For each $i \in [n]$ the polynomials $q_i(t)l_i - p_i(t) = 0$ and $q_i(t)r_i - p_i(t) = 0$ are of size $(O(d), O(\tau + \sigma))$. We compute isolating intervals for all their real solutions in $\widetilde{O}_B(d^2(\tau + \sigma))$ [27]. For any root t_0 of each of these polynomials, since ϕ is in reduced form (by assumption), we have that $t_0 \notin \mathsf{T}_P^{\mathbb{R}}$. We check if $\phi_j(t_0) \in [l_j, r_j]$, $j \in [n] \setminus i$. This requires 3 sign evaluations of univariate polynomials of size $(d, \tau + \sigma)$ at all roots of a polynomial of size $(d, \tau + \sigma)$. The bit complexity of performing these operations for all the roots is $\widetilde{O}_B(d^3 + d^2(\tau + \sigma))$ [39, Prop. 4, Prop. 6]. Since we repeat this procedure $n - 1$ times for every $i \in [n]$, the total cost is $\widetilde{O}_B(n^2d^3 + n^2d^2(\tau + \sigma))$. □

We partition $\mathsf{T}_C \cup \mathsf{T}_M \cup \mathsf{T}_E \cup \mathsf{T}_I \cup \mathsf{T}_B$ into groups of parameters that correspond to the same point on C. For each group, we add a vertex to G if and only if the corresponding point is inside the bounding box $\mathcal{B}$; for the characteristic box it is inside by construction.

Lemma 5.3. *The graph G has $\kappa = O(d^2 + nd)$ vertices, which can be computed using $O(d^2 + nd)$ arithmetic operations.*

Proof. Since $\mathsf{T}_B \cap \mathsf{T}_M = \emptyset$ and $\mathsf{T}_E \cap \mathsf{T}_M = \emptyset$, to each parameter in T_B and T_E corresponds a unique point on C. So for every $t \in \mathsf{T}_B \cup \mathsf{T}_E$ we add a vertex to G, labeled by the respective parameter. Next, we group the parameters in $\mathsf{T}_C \cup \mathsf{T}_M \cup \mathsf{T}_I$ that give the same point on C and we add for each group a vertex to G labeled by the corresponding parameter values.

Grouping of the parameters is done as follows: For every $t \in \mathsf{T}_C \cup \mathsf{T}_M$ we add a vertex to G labeled by the set $\{s \in \mathbb{R} : (s, t) \in S\} \cup \{t\}$ and for every $t \in \mathsf{T}_I$ we add a vertex to G labeled by the set $\{s \in \mathbb{C} : (s, t) \in S\} \cup \{t\}$. Notice that we took into account Rem. 3. We compute these sets simply by reading the elements of S.

It holds that $\mathsf{T}_B = O(nd)$, $\mathsf{T}_E = O(nd)$ and $|S| = O(d^2)$. Since for each vertex, we can find the parameters that give the same point in constant time, the result follows. □

We denote by $v_1, \ldots, v_\kappa$ the vertices (with distinct labels) of G and by $\lambda(v_1), \ldots, \lambda(v_\kappa)$ their label sets (i.e., the parameters that correspond to each vertex). Let T be the *sorted* list of parameters in $\mathsf{T}_C \cup \mathsf{T}_M \cup \mathsf{T}_E \cup \mathsf{T}_B$[3]. If for two consecutive elements $t_1 < t_2$ in T there exists a pole $s \in \mathsf{T}_P^{\mathbb{R}}$ such that $t_1 < s < t_2$, then we split T into two lists: T_1 containing the elements $\leq t_1$ and T_2 containing the elements $\geq t_2$. We continue recursively for T_1 and T_2, until there are no poles between any two elements of the resulting list. This procedure partitions T into $\mathsf{T}_1, \ldots, \mathsf{T}_\ell$.

To add edges to G, we consider each T_i with more than one element, where $i \in [\ell]$, independently. For any consecutive elements $t_1 < t_2$ in T_i, with $t_1 \in \lambda(v_{i,1})$ and $t_2 \in \lambda(v_{i,2})$, we

[3] Notice that we exclude the parameters of the isolated points.

add the edge $\{v_{i,1}, v_{i,2}\}$[4]. If $\mathbf{p}_\infty$ exists, we add an edge to the graph connecting the vertices corresponding to the last element of T_ℓ and the first element of the T_1.

Algorithm 2: PTOPO(ϕ) (Inside the characteristic box)

Input: A proper parametrization $\phi \in \mathbb{Z}(t)^n$ without singular points at infinity.
Output: Abstract graph G

Compute real poles $\mathsf{T}_P^{\mathbb{R}}$.
Compute parameters of 'special points' $\mathsf{T}_C, \mathsf{T}_M, \mathsf{T}_E, \mathsf{T}_I$.
/* Characteristic box */
$b \leftarrow 15d^2(\tau + \log d)$, $\mathcal{B}_C \leftarrow [-2^b, 2^b]^n$
$\mathsf{T}_B \leftarrow$ parameters that give to intersections of C with $\mathcal{B}_C$
Construct the set of vertices of G using Lem.5.3
Sort the list of all the parameters $\mathsf{T} = [\mathsf{T}_C, \mathsf{T}_M, \mathsf{T}_E, \mathsf{T}_B]$.
Let $T_1, \ldots, T_\ell$ the sublists of T when split at parameters in $\mathsf{T}_P^{\mathbb{R}}$
for *every list* $\mathsf{T}_i = [t_{i,1}, \ldots, t_{i,k_i}]$ **do**
  for $j = 1, \ldots, k_i - 1$ **do**
    Add the edge $\{t_{i,j}, t_{i,j+1}\}$ to the graph
if $\mathbf{p}_\infty$ *exists* **then**
  Add the edge $\{t_{1,1}, t_{\ell,k_\ell}\}$ to the graph

Theorem 5.4 (PTOPO inside the characteristic box). *Consider a proper parametrization ϕ of curve C involving polynomials of degree d and bitsize τ, as (1). Alg. 2 outputs a graph G that, if embedded in $\mathbb{R}^n$, is isotopic to C, within the characteristic box $\mathcal{B}_C$. It has worst case complexity*

$$\widetilde{O}_B(d^6(n+\tau) + nd^5\tau + n^2d^4\tau + d^3(n^2\tau + n^3) + n^3d^2\tau),$$

while its expected complexity is

$$\widetilde{O}_B(d^6\tau + nd^5\tau + n^2d^4\tau + d^3(n^2\tau + n^3) + n^3d^2\tau).$$

If $n = O(1)$, then bounds become $\widetilde{O}_B(N^7)$, where $N = \max\{d, \tau\}$.

Proof. We count on the fact that ϕ is continuous in $\mathbb{R} \setminus \mathsf{T}_P^{\mathbb{R}}$. Thus, for each real interval $[s,t]$ with $[s,t] \cap \mathsf{T}_P^{\mathbb{R}} = \emptyset$, there is a parametric arc connecting the points $\phi(s)$ and $\phi(t)$. Since for any (sorted) list T_i, for $i \in [\ell]$, the interval defined by the minimum and maximum value of its elements has empty intersection with $\mathsf{T}_P^{\mathbb{R}}$, then for any $s, t \in \mathsf{T}_i$ there exists a parametric arc connecting $\phi(s)$ and $\phi(t)$ and it is entirely contained in $\mathcal{B}_C$. If $\mathbf{p}_\infty$ exists, then $\mathbf{p}_\infty$ is inside $\mathcal{B}_C$. Let $t_{1,1}, t_{\ell,k_\ell}$ be the first element of the first list and the last element of the last list. There is a parametric arc connecting $\phi(t_{1,1})$ with $\mathbf{p}_\infty$ and $\mathbf{p}_\infty$ with $\phi(t_{\ell,k_\ell})$. So we add the edge $\{t_{1,1}, t_{\ell,\kappa_\ell}\}$ to G. Then, every edge of G is embedded to a unique smooth parametric arc and the embedding of G can be trivially continuously deformed to C.

For the complexity analysis, we know from Lem.4.7 that steps 1-2 can be performed in wost-case bit complexity

$$\widetilde{O}_B(nd^6 + nd^5\tau + d^4(n^2 + n\tau) + d^3(n^2\tau + n^3) + n^3d^2\tau),$$

and in expected bit complexity

$$\widetilde{O}_B(d^6 + d^5(n+\tau) + d^4(n^2 + n\tau) + d^3(n^2\tau + n^3) + n^3d^2\tau),$$

using a Las Vegas algorithm. From Lemmata 5.1, 5.2, and 5.3 steps 4-5 cost $\widetilde{O}_B(n^2(d^3\tau))$.

To perform steps 6-7 we must sort all the parameters in $\mathsf{T} \cup \mathsf{T}_P^{\mathbb{R}}$, i.e., we sort $O(d^2 + nd)$ algebraic numbers: The parameters that correspond to cusps and extreme points can be expressed as roots of $\prod_{i\in[n]} h_i(t,t)$, which is of size $(nd, n\tau)$. The poles are roots of $\prod_{i\in[n]} q_i(t)$, which has size $(nd, n\tau)$. The parameters that correspond to multiple points are roots of R_s which has size $(d^2, d\tau)$. At last, parameters in T_B are roots of a polynomial of size $(d, d^2\tau)$.

We can consider all these algebraic numbers together as roots of a single univariate polynomial (the product of all the corresponding polynomials). It has degree $O(d^2 + nd)$ and bitsize $\widetilde{O}(d^2\tau + n\tau)$. Hence, its separation bound is $\widetilde{O}(d^4\tau + nd^3\tau + nd^2\tau + n^2d\tau)$. To sort the list of all the algebraic numbers we have to perform $O(d^2 + nd)$ comparisons and each costs $\widetilde{O}(d^4\tau + nd^3\tau + nd^2\tau + n^2d\tau)$. Thus, the overall cost for sorting is $\widetilde{O}_B(d^6\tau + nd^5\tau + n^2d^4\tau + n^2d^3\tau + n^3d^2\tau)$. The overall bit complexities in the worst and expected case follow by summing the previous bounds. □

Following the proof of Thm. 5.4 we notice that the term $d^6\tau$ in the worst case bound is due to the introduction of the intersection points of C with $\mathcal{B}_C$. For visualizing the curve within $\mathcal{B}_C$, these points are essential and we cannot avoid them. However, if we are interested only in the topology of C, i.e., the abstract graph G, these points are not important any more. We sketch a procedure to avoid them and gain a factor of d in the complexity bound:

Assume that we have not computed the points on $C \cap \mathcal{B}_C$. We split again the sorted list $\mathsf{T} = [\mathsf{T}_C, \mathsf{T}_M, \mathsf{T}_E]$ at the real poles, and we add an artificial parameter at the beginning and at the end of each sublist. The rest of the procedure remains unaltered.

To verify the correctness of this approach, it suffices to prove that the graph that we obtain by this procedure, is isomorphic to the graph G at the output of Alg. 2. It is immediate to see that the latter holds, possibly up to the dissolution of the vertices corresponding to the first and last artificial vertices. Adding these artificial parameters does not affect the overall complexity, since we do not perform any algebraic operations. Therefore, the bit complexity of the algorithm is determined by the complexity of computing the parameters of the special points (Lem. 4.7), and so, we have the following theorem:

Theorem 5.5 (PTOPO and an abstract graph). *Consider a proper parametrization ϕ of curve C involving polynomials of degree d and bitsize τ, as (1). Alg. 2 outputs a graph G that, if embedded in $\mathbb{R}^n$, is isotopic to C. It has worst case complexity*

$$\widetilde{O}_B(nd^6 + nd^5\tau + d^4(n^2 + n\tau) + d^3(n^2\tau + n^3) + n^3d^2\tau),$$

while its expected complexity is

$$\widetilde{O}_B(d^6 + d^5(n+\tau) + d^4(n^2 + n\tau) + d^3(n^2\tau + n^3) + n^3d^2\tau),$$

If $n = O(1)$, then bounds become $\widetilde{O}_B(N^6)$, where $N = \max\{d, \tau\}$.

Remark 5. *If we are given a box $\mathcal{B} \subset \mathbb{R}^n$ at the input, we slightly modify* PTOPO, *as follows: We discard the parameter values in $\mathsf{T}_C \cup \mathsf{T}_M \cup \mathsf{T}_E \cup \mathsf{T}_I$ that correspond to points not contained in $\mathcal{B}$. Then, the set of G's vertices is constructed in the same way. To connect the vertices, we follow the previously used method with a minor modification: For any consecutive elements $t_1 < t_2$ in a list T_i with more than two elements, such that $t_1 \in \lambda(v_{i,1})$ and*

[4] To avoid multiple edges, we make the convention that we add an edge between $v_{i,j}$, $j = 1, 2$, and an (artificial) intermediate point corresponding to a parameter in (t_1, t_2).

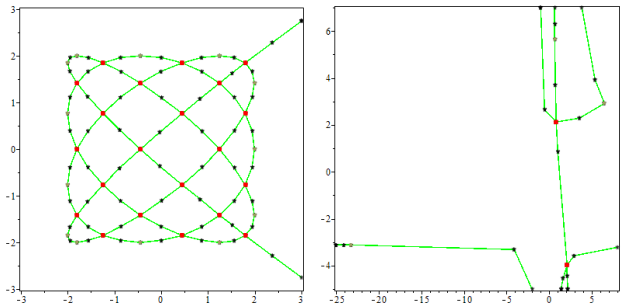


Figure 1: *The left figure is the output of* PTOPO *for the parametric curve* $(t^8 - 8t^6 + 20t^4 - 16t^2 + 2, t^7 - 7t^5 + 14t^3 - 7t)$ *while the right figure is the output for the curve* $(\frac{37t^3-23t^2+87t+44}{29t^3+98t^2-23t+10}, \frac{-61t^3-8t^2-29t+95}{11t^3-49t^2-47t+40})$.

$t_2 \in \lambda(v_{i,2})$, *we add the edge* $\{v_{i,1}, v_{i,2}\}$ *if and only if* $\phi(t_1), \phi(t_2)$ *are not both on the boundary of* $\mathcal{B}$*; or in other words* t_1 *and* t_2 *are not both in* T_B.

6 IMPLEMENTATION AND EXAMPLES

We have implemented `PTOPO` in MAPLE[5]. We build upon the real root isolation routines of MAPLE's `RootFinding` library and the SLV package [15], in order to use a certified implementation of general purpose exact computations with one and two real algebraic numbers, like comparison and sign evaluations. `PTOPO` computes the topology and visualizes parametric curves (currently planar).

To demonstrate its capabilities, we present in Fig. 1 the topology of two planar curves from [3]. For a given parametric representation of a curve, `PTOPO` computes the special points on the curve, the characteristic box, the corresponding graph, and then it visualizes the curve (inside the box). The computation, in all cases, takes less than a second in a MacBook laptop, running MAPLE 2019.

Acknowledgments FR, ET and ZZ are partially supported by Fondation Mathématique Jacques Hadamard PGMO grand ALMA, Agence Nationale de la Recherche ANR-17-CE40-0009, PHC GRAPE, and by the projects 118F321 under the program 2509, 118C240 under the program 2232, and 117F100 under the program 3501 of the Scientific and Technological Research Council of Turkey.

REFERENCES

[1] S. S. Abhyankar and C. J. Bajaj. Automatic parameterization of rational curves and surfaces IV: Algebraic space curves. *ACM Trans. Graph.*, 8(4):325–334, 1989.
[2] L. Alberti, B. Mourrain, and J. Wintz. Topology and arrangement computation of semi-algebraic planar curves. *CAGD*, 25(8):631 – 651, 2008.
[3] J. G. Alcázar and G. M. Díaz-Toca. Topology of 2D and 3D rational curves. *CAGD*, 27(7):483 – 502, 2010.
[4] S. Basu, R. Pollack, and M-F.Roy. *Algorithms in Real Algebraic Geometry*, volume 10 of *Algorithms and Computation in Mathematics*. Springer-Verlag, 2003.
[5] A. Bernardi, A. Gimigliano, and M. Idà. Singularities of plane rational curves via projections. *J. Symb. Comput.*, 09 2016.
[6] A. Blasco and S. Pérez-Díaz. An in depth analysis, via resultants, of the singularities of a parametric curve. *CAGD*, 68:22–47, 2019.
[7] J.-D. Boissonnat and M. Teillaud, editors. *Effective Computational Geometry for Curves and Surfaces*. Springer-Verlag, Mathematics and Visualization, 2006.

[5]https://webusers.imj-prg.fr/~christina.katsamaki/ptopo/

This project has received funding from the European Union's Horizon 2020 research and innovation programme under the Marie Skłodowska-Curie grant agreement No 754362.

[8] Y. Bouzidi, S. Lazard, G. Moroz, M. Pouget, F. Rouillier, and M. Sagraloff. Solving bivariate systems using Rational Univariate Representations. *J. Complexity*, 37:34–75, 2016.
[9] Y. Bouzidi, S. Lazard, M. Pouget, and F. Rouillier. Rational univariate representations of bivariate systems and applications. In *Proc. 38th Int'l Symp. on Symbolic and Algebraic Computation*, ISSAC '13, pages 109–116, NY, USA, 2013. ACM.
[10] Y. Bouzidi, S. Lazard, M. Pouget, and F. Rouillier. Separating linear forms and rational univariate representations of bivariate systems. *J. Symb. Comput.*, pages 84–119, 2015.
[11] L. Busé, C. Laroche, and F. Yıldırım. Implicitizing rational curves by the method of moving quadrics. *Computer-Aided Design*, 114:101–111, 2019.
[12] J. Caravantes, M. Fioravanti, L. Gonzalez-Vega, and I. Necula. Computing the topology of an arrangement of implicit and parametric curves given by values. In V. P. Gerdt, W. Koepf, W. M. Seiler, and E. V. Vorozhtsov, editors, *Computer Algebra in Scientific Computing*, pages 59–73, Cham, 2014. Springer.
[13] D. Cox, A. Kustin, C. Polini, and B. Ulrich. A study of singularities on rational curves via syzygies. *Memoirs of the American Mathematical Society*, 222, 02 2011.
[14] D. N. Diatta, S. Diatta, F. Rouillier, M.-F. Roy, and M. Sagraloff. Bounds for polynomials on algebraic numbers and application to curve topology. *arXiv preprint arXiv:1807.10622*, 2018.
[15] D. I. Diochnos, I. Z. Emiris, and E. P. Tsigaridas. On the asymptotic and practical complexity of solving bivariate systems over the reals. *J. Symb. Comput.*, 44(7):818–835, 2009. (Special issue on ISSAC 2007).
[16] R. T. Farouki, C. Giannelli, and A. Sestini. Geometric design using space curves with rational rotation-minimizing frames. In M. Dæhlen, M. Floater, T. Lyche, J.-L. Merrien, K. Mørken, and L. L. Schumaker, editors, *Mathematical Methods for Curves and Surfaces*, pages 194–208. Springer, 2010.
[17] W. Fulton. *Algebraic Curves. An Introduction to Algebraic Geometry*. Addison Wesley, 1969.
[18] X.-S. Gao and S.-C. Chou. Implicitization of rational parametric equations. *J. Symb. Comput.*, 14(5):459 – 470, 1992.
[19] J. Gutierrez, R. Rubio, and D. Sevilla. On multivariate rational function decomposition. *J. Symb. Comput.*, 33(5):545 – 562, 2002.
[20] J. Gutierrez, R. Rubio, and J.-T. Yu. D-resultant for rational functions. *Proc. American Mathematical Society*, 130, 08 2002.
[21] X. Jia, X. Shi, and F. Chen. Survey on the theory and applications of μ-bases for rational curves and surfaces. *J. Comput. Appl. Math.*, 329:2–23, 2018.
[22] A. Kobel and M. Sagraloff. On the complexity of computing with planar algebraic curves. *J. Complexity*, 31, 08 2014.
[23] Y.-M. Li and R. J. Cripps. Identification of inflection points and cusps on rational curves. *CAGD*, 14(5):491 – 497, 1997.
[24] T. Lickteig and M.-F. Roy. Sylvester–Habicht sequences and fast Cauchy index computation. *J. Symb. Comput.*, 31(3):315–341, Mar. 2001.
[25] K. Mahler. On some inequalities for polynomials in several variables. *J. London Mathematical Society*, 1(1):341–344, 1962.
[26] D. Manocha and J. F. Canny. Detecting cusps and inflection points in curves. *CAGD*, 9(1):1 – 24, 1992.
[27] V. Pan and E. Tsigaridas. Accelerated approximation of the complex roots and factors of a univariate polynomial. *Theor. Computer Science*, 681:138 – 145, 2017.
[28] H. Park. Effective computation of singularities of parametric affine curves. *J. Pure and Applied Algebra*, 173:49–58, 08 2002.
[29] S. Pérez-Díaz. On the problem of proper reparametrization for rational curves and surfaces. *CAGD*, 23(4):307–323, 2006.
[30] S. Pérez-Díaz. Computation of the singularities of parametric plane curves. *J. Symb. Comput.*, 42(8):835 – 857, 2007.
[31] C. A. T. Recio. Plotting missing points and branches of real parametric curves. *Applicable Algebra in Engineering, Communication and Computing*, 18, 02 2007.
[32] R. Rubio, J. Serradilla, and M. Vélez. Detecting real singularities of a space curve from a real rational parametrization. *J. Symb. Comput.*, 44(5):490 – 498, 2009.
[33] A. Schönhage. Probabilistic computation of integer polynomial gcds. *J. Algorithms*, 9(3):365 – 371, 1988.
[34] T. W. Sederberg. Improperly parametrized rational curves. *CAGD*, 3(1):67–75, May 1986.
[35] T. W. Sederberg and F. Chen. Implicitization using moving curves and surfaces. In *Proc. of the 22nd Annual Conference on Computer Graphics and Interactive Techniques*, SIGGRAPH '95, pages 301–308, NY, USA, 1995.
[36] J. R. Sendra. Normal parametrizations of algebraic plane curves. *J. Symb. Comput.*, 33:863–885, 2002.
[37] J. R. Sendra and F. Winkler. Algorithms for rational real algebraic curves. *Fundam. Inf.*, 39(1,2):211–228, Apr. 1999.
[38] J. R. Sendra, F. Winkler, and S. Pérez-Díaz. Rational algebraic curves. *Algorithms and Computation in Mathematics*, 22, 2008.
[39] A. Strzebonski and E. Tsigaridas. Univariate real root isolation in an extension field and applications. *J. Symb. Comput.*, 92:31 – 51, 2019.
[40] J. von zur Gathen and J. Gerhard. *Modern computer algebra*. Cambridge University Press, 3rd edition, 2013.
[41] R. J. Walker. *Algebraic curves*. Springer-Verlag, 1978.

On the Skolem Problem and Prime Powers

George Kenison
george.kenison@cs.ox.ac.uk
University of Oxford
Oxford, UK

Richard Lipton
richard.lipton@cc.gatech.edu
Georgia Institute of Technology
Atlanta, USA

Joël Ouaknine*
joel@mpi-sws.org
Max Planck Institute for Software Systems
Saarbrücken, Germany

James Worrell†
jbw@cs.ox.ac.uk
University of Oxford
Oxford, UK

ABSTRACT

The Skolem Problem asks, given a linear recurrence sequence (u_n), whether there exists $n \in \mathbb{N}$ such that $u_n = 0$. In this paper we consider the following specialisation of the problem: given in addition $c \in \mathbb{N}$, determine whether there exists $n \in \mathbb{N}$ of the form $n = lp^k$, with $k, l \leq c$ and p any prime number, such that $u_n = 0$.

CCS CONCEPTS

• **Mathematics of computing → Discrete mathematics**.

KEYWORDS

Skolem Problem, Algebraic number theory, Recurrence sequences, Decidability

ACM Reference Format:
George Kenison, Richard Lipton, Joël Ouaknine, and James Worrell. 2020. On the Skolem Problem and Prime Powers. In *International Symposium on Symbolic and Algebraic Computation (ISSAC '20), July 20–23, 2020, Kalamata, Greece.* ACM, New York, NY, USA, 8 pages. https://doi.org/10.1145/3373207.3404036

1 INTRODUCTION

A sequence $(u_n)_{n=0}^{\infty}$ of real algebraic numbers is called a *linear recurrence sequence* if its terms satisfy a recurrence relation $u_n = a_1u_{n-1} + a_2u_{n-2} + \cdots + a_\ell u_{n-\ell}$, with fixed real algebraic constants $a_1, \ldots, a_\ell$ such that $a_\ell \neq 0$. Such a recurrence is said to have order ℓ and a sequence (u_n) satisfying the recurrence is wholly determined by the initial values $u_0, \ldots, u_{\ell-1}$. The study of linear recurrence sequences is motivated by a wide range of phenomena, in areas such as analysis of algorithms, and biological and economic modelling. Natural decision problems for linear recurrence sequences include: whether all the terms in a sequence are positive, whether the terms of the sequence are eventually positive, and whether the sequence contains a zero. The latter, commonly known as the Skolem Problem [6, 7], is the main object of study in the current paper.

Let (u_n) be a linear recurrence sequence. A remarkable result of Skolem, Mahler, and Lech states that the set $\{n \in \mathbb{N} : u_n = 0\}$ is the union of a finite set together with a finite number of (infinite) arithmetic progressions. The original result, proved by Skolem [14] for the field of rational numbers, was subsequently extended to the field of algebraic numbers by Mahler [9, 10], and then further extended to any field of characteristic 0 by Lech [8]. All known proofs of the Skolem-Mahler-Lech Theorem (as it is now known) employ techniques from p-adic analysis. These proofs are non-constructive and the decidability of the Skolem Problem remains open. Berstel and Mignotte, however, gave an effective method to obtain all of the arithmetic progressions in the statement of the theorem [2].

For fields of positive characteristic, the conclusion of the Skolem-Mahler-Lech Theorem does not hold. Indeed, Lech [8] gave the following illustrative example. Let $K = \mathbb{F}_p(t)$ and consider the sequence with terms $u_n = (1+t)^n - t^n - 1$. Then (u_n) satisfies a linear recurrence over K, but $u_n = 0$ if, and only if, $n = p^k$. Nevertheless, Derksen [5] established an analogue of the Skolem-Mahler-Lech Theorem for fields of positive characteristic, namely he proved that the set of zeroes in a field of characteristic p is a p-automatic set. The proof of Derksen was moreover effective, allowing to construct for a given sequence the automaton representing the set of its zeros.

Returning to the characteristic-zero setting, progress on the decidability of the Skolem Problem has been made by restricting the problem to linear recurrence sequences of low order. Decidability of the Skolem Problem for sequences of order at most 2 is straightforward and the results are considered folklore. Breakthrough work by Mignotte, Shorey, and Tijdeman [11], and, independently, Vereshchagin [15], showed decidability of the Skolem Problem for linear recurrence sequences of order 3 and 4. Techniques from p-adic analysis and algebraic number theory are employed in both [11] and [15]. Both papers moreover make critical use of Baker's theorem for linear forms in logarithms of algebraic numbers. The approach via Baker's Theorem taken in the above papers does not appear to extend easily to recurrences of higher order. In particular, decidability of Skolem's Problem remains open for recurrences of order 5. However, the recent resurgence of research activity concerning the decidability of various sub-cases of the Skolem Problem and related questions (see the survey [13]) gives an indication of its fundamental importance to the field.

*Also affiliated with the Department of Computer Science, Oxford University, UK.
Supported by ERC grant AVS-ISS (648701) and DFG grant 389792660 as part of TRR 248 (see https://perspicuous-computing.science).
†Supported by EPSRC Fellowship EP/N008197/1.

ISSAC '20, July 20–23, 2020, Kalamata, Greece

ACM ISBN 978-1-4503-7100-1/20/07.
https://doi.org/10.1145/3373207.3404036

In this paper we pursue an alternative approach to restricting the order of the recurrence as a means of obtaining decidable specialisations of Skolem's Problem. We consider general recurrences, but ask to decide the existence of zeros of certain prescribed forms. For example, we ask whether one can show decidability of the Skolem Problem when we consider only those $n \in \mathbb{N}$ that are prime powers. Our first basic result—which we will generalise in various ways in the rest of the paper—is the following, which applies to a class of *simple* linear recurrence sequences (i.e., those sequences without repeated characteristic roots):

Theorem 1.1. *Suppose that each term in a linear recurrence sequence (u_n) can be written as an algebraic exponential polynomial $u_n = A_1\lambda_1^n + \cdots + A_m\lambda_m^n$ with $A_1, \ldots, A_m \in \mathbb{Z}$ and $\lambda_1, \ldots, \lambda_m$ distinct algebraic integers. Fix $c \in \mathbb{N}$. Then one can decide whether there exists $n \in \{p^k : p \text{ prime}, k \leq c\}$ such that $u_n = 0$.*

In general, a simple linear recurrence sequence (u_n) has the property that each of its terms is given by an algebraic exponential polynomial $u_n = A_1\lambda_1^n + \cdots + A_m\lambda_m^n$ with $A_1, \ldots, A_m \in \mathfrak{O}$ algebraic integers in a number field K. In Theorem 1.1 we assumed that $A_1, \ldots, A_m \in \mathbb{Z}$. More generally, a linear recurrence sequence (u_n) can always be written in the form $u_n = A_1(n)\lambda_1^n + \cdots + A_m(n)\lambda_m^n$, where the A_i are univariate polynomials and the λ_i are characteristic roots of the recurrence relation. We establish decidability results for linear recurrence sequences (u_n) in this general setting. We consider the case of rational polynomial coefficients in Section 3; that is, $A_1, \ldots, A_m \in \mathbb{Z}[x]$ and, more generally, algebraic polynomial coefficients in Section 5. We outline two generalisations of Theorem 1.1 below.

First, assume that the linear recurrence sequence (u_n) satisfies $u_n = A_1(n)\lambda_1^n + \cdots + A_m(n)\lambda_m^n$ such that $A_1, \ldots, A_m \in \mathbb{Z}[x]$. The next result follows as a corollary to Theorem 3.3. In the proof of Theorem 3.3 we introduce and analyse an associated simple linear recurrence (v_n) with terms $v_n = A_1(0)\lambda_1^n + \cdots + A_m(0)\lambda_m^n$.

Theorem 1.2. *Let (u_n) be a recurrence sequence with rational polynomial coefficients and (v_n) the associated simple recurrence. Fix $c \in \mathbb{N}$. If $v_1 \neq 0$ then one can decide whether there exists $n \in \{p^k : p \text{ prime}, k \leq c\}$ such that $u_n = 0$.*

Now suppose that the terms of (u_n) are given by $u_n = A_1(n)\lambda_1^n + \cdots + A_m(n)\lambda_m^n$ where the coefficients $A_1, \ldots, A_m \in \mathfrak{O}[x]$ are univariate polynomial with $\mathfrak{O}$ the ring of integers of a finite Galois extension K over $\mathbb{Q}$. As before, let (v_n) be the associated simple recurrence. To each rational prime p we associate a constant $f(p)$ (the *inertial degree* of $p\mathbb{Z}$ in K). The next result follows as a corollary to Theorem 4.1.

Theorem 1.3. *Suppose that (u_n) is a recurrence sequence with algebraic polynomial coefficients and (v_n) the associated linear recurrence as above. Fix $c \in \mathbb{N}$. If $v_1 \neq 0$ then one can decide whether there exists $n \in \{p^{kf(p)} : p \text{ prime}, k \leq c\}$ such that $u_n = 0$.*

We motivate our decidability results with a discussion of the decidability of the Skolem Problem for linear recurrence sequences of order 5. The authors of [7] claim to prove that the Skolem Problem is decidable for integer linear recurrence sequences of order 5; however, as pointed out in [12], there is a gap in the argument. The critical case for which the decidability of the Skolem Problem is open is that of a recurrence sequence of order 5 whose characteristic polynomial has five distinct roots: four distinct roots $\lambda_1, \overline{\lambda_1}, \lambda_2, \overline{\lambda_2} \in \mathbb{C}$ such that $|\lambda_1| = |\lambda_2|$, and a fifth root $\rho \in \mathbb{R}$ of strictly smaller magnitude. In this case the terms of such a recurrence sequence (u_n) are given by $u_n = a\left(\lambda_1^n + \overline{\lambda_1^n}\right) + b\left(\lambda_2^n + \overline{\lambda_2^n}\right) + c\rho^n$. Here $a, b, c \in \mathbb{R}$ are algebraic numbers. If $|a|$ and $|b|$ are not equal then there is no known general procedure to determine $\{n \in \mathbb{N} : u_n = 0\}$.

Next we consider an example of a linear recurrence sequence from the aforementioned critical case. We motivate the results herein and also illustrate the techniques used in this paper by demonstrating that the sequence does not vanishes at any prime index.

Example 1.4. For this example set $\lambda_1 = 39 + 52\mathrm{i}$, $\lambda_2 = -60 + 25\mathrm{i}$ and $\rho = 1$. (Our choices of Pythagorean triples $(39, 52, 65)$ and $(25, 60, 65)$ ensure that $|\lambda_1| = |\lambda_2| = 65$.) Let (v_n) be the linear recurrence sequence whose terms satisfy

$$v_n = \lambda_1^n + \overline{\lambda_1^n} + 3\left(\lambda_2^n + \overline{\lambda_2^n}\right) + \rho^n.$$

There are no rational primes $p \in \mathbb{N}$ for which $v_p = 0$.

We omit many technical definitions and details in the following presentation (for such details we refer the reader to the preliminariy material in the next section).

Proof of Example 1.4. Let K be the *splitting field* of the minimal polynomial (over $\mathbb{Q}$) associated to (v_n). We find that $K = \mathbb{Q}(\lambda_1, \overline{\lambda_1}, \lambda_2, \overline{\lambda_2}, 1) \cong \mathbb{Q}(i)$. The *dimension d* of the field K as a vector space over $\mathbb{Q}$ is 2. There is a computable constant $N \in \mathbb{N}$ depending only on v_1 and the field K introduced in the preliminaries—the norm of the principal ideal generated by v_1—with the following property. Suppose that $p \in \mathbb{N}$ is a rational prime. Then, by Corollary 3.1 and Lemma 3.2, $v_p = 0$ only if $p \mid N$.

Assume that $v_p = 0$ for some prime $p \in \mathbb{N}$. We calculate $v_1 = -281$, which we use to determine N. Here $N = |v_1|^d = 281^2$. Thus $p \mid N = 281^2$ from our assumption. By happy coincidence, 281 is a rational prime and so it is sufficient to check whether $v_p = 0$ for the only possible candidate $p = 281$. Using Mathematica we compute $v_{281} \approx 3.7 \times 10^{509}$ (to two significant figures). We conclude that there does not exist a rational prime $p \in \mathbb{N}$ such that $v_p = 0$. □

This paper is organised as follows. In Section 2, we recall preliminary terminology and background material from algebraic number theory and recurrence sequences. In Section 3, we prove decidability results locating zeroes of recurrence sequences of the form $u_n = A_1(n)\lambda_1^n + \cdots + A_m(n)\lambda_m^n$ with polynomial coefficients $A_1, \ldots, A_m \in \mathbb{Z}[x]$ having integer coefficients. The main result in Section 3 is Theorem 3.3. In Section 4 we prove decidability results for linear recurrence sequences with polynomial coefficients $A_1, \ldots, A_m \in \mathfrak{O}[x]$, where $\mathfrak{O}$ is the ring of integers of a Galois number field. The main result in Section 4 is Theorem 4.1. In Section 5 we show that the problem of deciding whether a given linear recurrence sequence has a prime zero is NP-hard. This matches the best known lower bound for the general Skolem Problem.

2 ALGEBRAIC NUMBER THEORY AND LINEAR RECURRENCE SEQUENCES

In this section we recall some basic notions concerning algebraic numbers and linear recurrences that will be used in the sequel.

A complex number α is *algebraic* if there exists a polynomial $P \in \mathbb{Q}[x]$ such that $P(\alpha) = 0$. The *minimal polynomial* of $\alpha \in \mathbb{A}$ is the unique monic polynomial $\mu_\alpha \in \mathbb{Q}[x]$ of least degree such that $\mu(\alpha) = 0$. The *degree* of α, written $\deg(\alpha)$, is the degree of its minimal polynomial. An *algebraic integer* α is an algebraic number whose minimal polynomial has integer coefficients. The collection of all algebraic integers forms a ring $\mathbb{B}$.

A *number field* K is a field extension of $\mathbb{Q}$ whose dimension as a vector space over $\mathbb{Q}$ is finite. We call the dimension of this vector space the *degree* of the number field and use the notation $[K\colon \mathbb{Q}]$ for the degree of K. Call a number field K *Galois* if it is the splitting field of some separable polynomial over $\mathbb{Q}$. Let $\mathfrak{O} = \mathbb{B} \cap K$ be the ring of algebraic integers in K. Because $\mathbb{B} \cap \mathbb{Q} = \mathbb{Z}$, we refer to the elements of $\mathbb{Z}$ as *rational integers*. For each $\alpha \in K$ there exists a non-zero $q \in \mathbb{Z}$ such that $q\alpha \in \mathfrak{O}$.

Given a number field K of degree d over $\mathbb{Q}$, there are exactly d distinct monomorphisms $\sigma_i\colon K \to \mathbb{C}$. We define the *norm* $N_K(\alpha)$ of $\alpha \in K$ by

$$N_K(\alpha) = \prod_{\ell=1}^{d} \sigma_\ell(\alpha).$$

Then $N_K(\alpha) \in \mathbb{Q}$ and furthermore $N_K(\alpha) \in \mathbb{Z}$ if $\alpha \in \mathfrak{O}$.

Suppose that $P \in \mathbb{Z}[x]$ is a polynomial with integer coefficients. The *height* of P is the maximum of the absolute values of its coefficients and write $\|P\|$ for the bit length of the list of its coefficients encoded in binary. It is clear that the degree of P is at most $\|P\|$, and the height of P is at most $2^{\|P\|}$.

There is a standard representation of an algebraic number α as a tuple $(\mu_\alpha, a, b, \varepsilon)$ where μ_α is the minimal polynomial of α and $a, b, \varepsilon \in \mathbb{Q}$ with $\varepsilon > 0$ sufficiently small so that α is the unique root of μ_α inside the ball of radius ε centred at $a + b\mathrm{i} \in \mathbb{C}$. Given a polynomial $P \in \mathbb{Z}[x]$, we can compute a standard representation for each of its roots in time polynomial in $\|P\|$.

We recall some standard terminology and basic results about ideals in $\mathfrak{O}$. The ideal $\mathfrak{a} = a\mathfrak{O}$ generated by a single element $a \in \mathfrak{O}$ is called *principal*. For two ideals $\mathfrak{a}$ and $\mathfrak{b}$ of $\mathfrak{O}$, define the sum and product by

$$\mathfrak{a} + \mathfrak{b} := \{a + b : a \in \mathfrak{a}, b \in \mathfrak{b}\}, \quad \text{and}$$

$$\mathfrak{a}\mathfrak{b} := \left\{ \sum_{j=1}^{k} a_j b_j : a_j \in \mathfrak{a}, b_j \in \mathfrak{b} \right\}.$$

Two ideals $\mathfrak{a}$ and $\mathfrak{b}$ are said to be *coprime* if $\mathfrak{a} + \mathfrak{b} = \mathfrak{O}$. In this case we have $\mathfrak{a}\mathfrak{b} = \mathfrak{a} \cap \mathfrak{b}$.

For ideals $\mathfrak{a}, \mathfrak{b}$ of $\mathfrak{O}$ we say $\mathfrak{a}$ *divides* $\mathfrak{b}$, and write $\mathfrak{a} \mid \mathfrak{b}$, if there exists an ideal $\mathfrak{c}$ such that $\mathfrak{b} = \mathfrak{a}\mathfrak{c}$. In addition, $\mathfrak{a} \mid \mathfrak{b}$ if, and only if, $\mathfrak{b} \subseteq \mathfrak{a}$. An ideal $\mathfrak{p}$ of $\mathfrak{O}$ is called *prime* if $\mathfrak{p} \mid \mathfrak{a}\mathfrak{b}$ implies $\mathfrak{p} \mid \mathfrak{a}$ or $\mathfrak{p} \mid \mathfrak{b}$. Recall that the ring of integers $\mathfrak{O}$ of a number field does not necessarily have unique factorisation. However every non-zero ideal of $\mathfrak{O}$ can be written as a product of prime ideals and, in addition, this factorisation is unique up to the order of the factors.

Let $\mathfrak{a}$ be a non-zero ideal of $\mathfrak{O}$ then the quotient ring $\mathfrak{O}/\mathfrak{a}$ is finite, which leads us to define the *norm* of $\mathfrak{a}$ by $N(\mathfrak{a}) = |\mathfrak{O}/\mathfrak{a}|$. This norm has a multiplicative property: $N(\mathfrak{a}\mathfrak{b}) = N(\mathfrak{a})N(\mathfrak{b})$ for every pair of non-zero ideals $\mathfrak{a}, \mathfrak{b}$ of $\mathfrak{O}$. We can connect norms of elements and ideals as follows. Suppose that $a \in \mathfrak{O}$ is non-zero then $N(a\mathfrak{O}) = |N_K(a)|$ and, in addition, if $a \in \mathbb{Q}$ then $N(a\mathfrak{O}) = |a^d|$ where $d = [K\colon \mathbb{Q}]$.

Suppose that $\mathfrak{p}$ is a prime ideal. Since the quotient ring $\mathfrak{O}/\mathfrak{p}$ is a finite field and, by definition, $N(\mathfrak{p}) = |\mathfrak{O}/\mathfrak{p}|$, we conclude that $N(\mathfrak{p}) = p^f$ where $f \leq [K\colon \mathbb{Q}]$ and p is a rational prime. Indeed, $p \in \mathfrak{p}$ and, further, it is the only rational prime in $\mathfrak{p}$. Thus, we say that the prime ideal $\mathfrak{p}$ *lies above* the prime ideal $p\mathbb{Z}$. We will frequently use the following version of Fermat's Little Theorem:

Theorem 2.1. *For any prime ideal $\mathfrak{p}$ and algebraic integer $\lambda \in \mathfrak{O}$, $\lambda^{N(\mathfrak{p})} - \lambda \in \mathfrak{p}$.*

We now recall some of the terminology connecting linear recurrence sequences and exponential polynomials. For further details on this correspondence we refer the reader to [6].

We call a sequence of algebraic numbers $(u_n)_{n=0}^{\infty}$ satisfying a recurrence relation $u_n = a_1u_{n-1} + a_2u_{n-2} + \cdots + a_\ell u_{n-\ell}$ with fixed real algebraic constants $a_1, \ldots, a_\ell$ such that $a_\ell \neq 0$ a *linear recurrence sequence*. Together with the recurrence relation, the sequence is wholly determined by the initial values $u_0, \ldots, u_{\ell-1}$. The polynomial $f(x) = x^\ell - a_1x^{\ell-1} - \cdots - a_{\ell-1}x - a_\ell$ is called the *characteristic polynomial* associated to the relation. Associated to each linear recurrence sequence (u_n) is a recurrence relation of minimal length. We call the characteristic polynomial of this minimal length relation the *minimal polynomial* of the sequence. Moreover, given a recurrence relation the minimal polynomial divides any characteristic polynomial. The *order* of a linear recurrence sequence is the degree of its minimal polynomial.

Let μ be the minimal polynomial of a linear recurrence sequence (u_n) and K the splitting field of μ. Over K the polynomial factorises as a product of powers of distinct linear factors $\mu(x) = \prod_{i=1}^{m}(x - \lambda_i)^{n_i}$. Here the constants $\lambda_1, \ldots, \lambda_m \in K$ are the *characteristic roots* of (u_n) with multiplicities $n_1, \ldots, n_m$. The terms of a linear recurrence sequence can be realised as an *exponential polynomial* such that $u_n = \sum_{i=1}^{m} A_i(n)\lambda_i^n$. Here the λ_i are the distinct characteristic roots of the recurrence (u_n) alongside polynomial coefficients $A_k \in K[x]$. If the characteristic polynomial of a sequence has no repeated roots, the terms in the sequence are each given by an exponential polynomial $u_n = \sum_{i=1}^{m} A_i(0)\lambda_i^n$ with constant coefficients. A linear recurrence sequence that satisfies this condition is called *simple*.

Suppose that $(u_n)_{n=0}^{\infty}$ is a linear recurrence sequence with characteristic roots $\lambda_1, \ldots, \lambda_m \in K$. For each $i \in \{1, \ldots, m\}$ there exist non-zero $q_i \in \mathbb{Z}$ such that $q_i\lambda_i \in \mathfrak{O}$. Consider the linear recurrence sequence $(w_n)_{n=0}^{\infty}$ with terms given by $w_n = q_1^n \cdots q_m^n u_n$. By construction, $w_n = 0$ if and only if $u_n = 0$ and, further, the characteristic roots of (w_n) are algebraic integers in $\mathfrak{O}$. Thus, without loss of generality, we assume that each $\lambda_i \in \mathfrak{O}$ and, in addition, that $A_1, \ldots, A_m \in \mathfrak{O}[x]$.

Let (u_n) be a linear recurrence sequence with terms

$$u_n = A_1(n)\lambda_1^n + \cdots + A_m(n)\lambda_m^n$$

where $\lambda_1, \ldots, \lambda_m \in \mathfrak{O}$ and $A_1, \ldots, A_m \in \mathfrak{O}[x]$. We associate to (u_n) a simple linear recurrence (v_n) given by an exponential polynomial $v_n = A_1(0)\lambda_1^n + \cdots + A_m(0)\lambda_m^n$.

We are interested in determining whether $u_n = 0$ for $n = \ell p^k$ with $k, \ell \in \mathbb{N}$ bounded and p any rational prime. In particular, our method is limited to those coefficients $\ell \in \{0, 1, \ldots, c\}$ for which $v_\ell \neq 0$. We introduce the set $\mathcal{L}_c = \{\ell \in \mathbb{N} : \ell \leq c,\ v_\ell \neq 0\}$ consisting of such coefficients. In the case that $(u_n)_{n=0}^{\infty}$ is simple we have that $u_n = v_n$ for each $n \in \mathbb{N}$, and so we need only consider the $\ell \leq c$ such that $u_\ell \neq 0$. In the case that $(u_n)_{n=0}^{\infty}$ is not simple it is possible that (v_n) is identically zero; for example, $u_n = n\lambda^n$. If $v_0 \neq 0$ then (v_n) is not identically zero. Otherwise $v_0 = u_0 = 0$ and we have identified a zero term at an index of the desired form.

3 COEFFICIENTS IN $\mathbb{Z}[x]$

3.1 Decidability results

Given a positive rational integer n, recall the multinomial expansion with exponent n is given by the identity

$$(A_1 x_1 + \cdots + A_m x_m)^n = \sum_{b_1 + \cdots + b_m = n} \binom{n}{b_1, b_2, b_3, \ldots, b_m} \prod_{t=1}^{m} A_t^{b_t} x_t^{b_t}$$

with the combinatorial coefficient representing the quotient

$$\binom{n}{b_1, b_2, b_3, \ldots, b_m} = \frac{n!}{b_1! b_2! \cdots b_m!}.$$

We shall make use of the following result, commonly called the *freshman's dream.*

Corollary 3.1. *Suppose that $A_1, \ldots, A_m \in \mathbb{Z}$ and $\lambda_1, \ldots, \lambda_m$ lie in the ring $\mathfrak{O}$ of integers of some number field k. Then for any prime p and $k \in \mathbb{N}$ we have the following congruence:*

$$(A_1\lambda_1 + \cdots + A_m\lambda_m)^{p^k} \equiv A_1\lambda_1^{p^k} + \cdots + A_m\lambda_m^{p^k} \pmod{p\mathfrak{O}}.$$

Proof. Let us expand the left-hand side using the aforementioned multinomial identity. Now consider each of the combinatorial coefficients in this expansion. If exactly one of the choices $b_1, \ldots, b_t$ is equal to p^k then the corresponding coefficient is equal to 1, and otherwise it is an integer multiple of p. Hence

$$(A_1\lambda_1 + \cdots + A_m\lambda_m)^{p^k} \equiv A_1^{p^k}\lambda_1^{p^k} + \cdots + A_m^{p^k}\lambda_m^{p^k} \pmod{p\mathfrak{O}}.$$

The result follows by repeated application of Fermat's Little Theorem, $A_i^{p^k} \equiv A_i \pmod{p\mathbb{Z}}$. □

In combination with Corollary 3.1, we use the following technical lemma in the proof of Theorem 1.1.

Lemma 3.2. *Suppose that $b \in \mathfrak{O}$ is non-zero. There are only finitely many rational primes p such that $p\mathfrak{O} \mid b\mathfrak{O}$ and, in addition, $N(b\mathfrak{O})$ is an effective bound on such primes.*

Proof. Since the ideal norm is multiplicative we have $p^d = N(p\mathfrak{O}) \mid N(b\mathfrak{O})$ where $d = [K : \mathbb{Q}]$. We can calculate $N(b\mathfrak{O}) \in \mathbb{Z}$ and so obtain an effective bound on any rational prime p such that $p\mathfrak{O} \mid b\mathfrak{O}$. □

Proof of Theorem 1.1. Let us assume that the algebraic integers $\lambda_1, \ldots, \lambda_m$ all lie in a given number field K, and let us denote by $\mathfrak{O}$ the ring of algebraic integers in K. We note that it is decidable whether $u_{p^0} = u_1 = A_1 + \cdots + A_m = 0$. Thus we can assume, without loss of generality, that $u_1 \neq 0$. We shall prove the case $k = 1$. The proof for higher powers follows with only minor changes to the argument below.

By Corollary 3.1, the following congruence holds modulo $p\mathfrak{O}$,

$$u_1^p = (A_1\lambda_1 + \cdots + A_m\lambda_m)^p \equiv A_1\lambda_1^p + \cdots + A_m\lambda_m^p = u_p.$$

Thus u_1^p and u_p lie in the same coset of $p\mathfrak{O}$. It follows that $u_p = 0$ only if $u_1^p \in p\mathfrak{O}$. Since $p\mathfrak{O} \mid u_1^p\mathfrak{O}$ and $u_1 \neq 0$ (by assumption), we can apply Lemma 3.2. As $N(u_1^p\mathfrak{O})$ has only finitely many prime divisors, we obtain an effective bound on the rational primes p such that $u_p = 0$. We have the desired result: given $c \in \mathbb{N}$, it is decidable whether there exists an $n \in \{p : p \text{ prime}\}$ such that $u_n = 0$. □

We now turn our attention to decidability results for linear recurrence sequences whose terms are given by an exponential polynomial with polynomial coefficients in $\mathbb{Z}[x]$.

Let (u_n) be a linear recurrence sequence whose terms are given by $u_n = A_1(n)\lambda_1^n + \cdots + A_m(n)\lambda_m^n$ with $A_1, \ldots, A_m \in \mathbb{Z}[x]$ and $\lambda_1, \ldots, \lambda_m \in \mathfrak{O}$ for some ring of integers in a number field K. We associate a simple sequence (v_n) with terms given by $v_n = A_1(0)\lambda_1^n + \cdots + A_m(0)\lambda_m^n$ to each such sequence (u_n). Given $c \in \mathbb{N}$, we define the set $\mathcal{N}_c \subset \mathbb{N}$ as follows:

$$\mathcal{N}_c := \bigcup_{\ell \in \mathcal{L}_c} \{\ell p^k : p \text{ prime},\ k \leq c\}.$$

We recall the set $\mathcal{L}_c = \{\ell \in \mathbb{N} : \ell \leq c,\ v_\ell \neq 0\}$ defined in the previous section. Hence $\mathcal{N}_c$ implicitly depends on the sequence (u_n). If $u_0 = 0$ then we have identified a zero term at a desired index. Otherwise $u_0 \neq 0$ and so, for c sufficiently large, $\mathcal{N}_c$ is infinite. The goal of this section is to prove the following theorem.

Theorem 3.3. *Let (u_n) be a linear recurrence sequence whose terms are given by an exponential polynomial with rational polynomial coefficients as above. Fix $c \in \mathbb{N}$. Then one can decide whether there is an $n \in \mathcal{N}_c$ such that $u_n = 0$.*

Lemma 3.4 below is a generalisation of Corollary 3.1 in two senses: the lemma considers sequences that are not necessarily simple and indices of the form $\ell p^k \in \mathbb{N}$.

Lemma 3.4. *Let (u_n) be a recurrence sequence as above and (v_n) the associated simple recurrence sequence. Let $p \in \mathbb{N}$ be prime and $k, \ell \in \mathbb{N}$. Then $v_\ell^{p^k} - u_{\ell p^k} \in p\mathfrak{O}$.*

Proof. We prove the case when $k = 1$. The general case, dealing with higher powers p^k, follows with only minor changes.

First, we have the congruence $v_\ell^p \equiv v_{\ell p} \pmod{p\mathfrak{O}}$ by Corollary 3.1 since

$$\left(A_1(0)\lambda_1^\ell + \cdots + A_m(0)\lambda_m^\ell\right)^p \equiv A_1(0)\lambda_1^{\ell p} + \cdots + A_m(0)\lambda_m^{\ell p}.$$

Recall that for $A \in \mathbb{Z}[x]$ we have $(x - y) \mid (A(x) - A(y))$. By induction, one can show that $p \mid (A(lp) - A(0))$ and so $A(0) \equiv A(\ell p) \pmod{p\mathbb{Z}}$ for each $A \in \mathbb{Z}[x]$. This is sufficient to deduce a second congruence

$$v_{\ell p} \equiv A_1(\ell p)\lambda_1^{\ell p} + \cdots + A_m(\ell p)\lambda_m^{\ell p} = u_{\ell p} \pmod{p\mathfrak{O}}.$$

Together these two congruences give $v_\ell^p - u_{\ell p} \in p\mathfrak{O}$, the desired result. □

Proof of Theorem 3.3. Let us consider the case that $k = 1$. As previously noted, we can assume there is an $\ell \leq c$ and $v_\ell \neq 0$ (otherwise $u_0 = 0$). Suppose that $u_{\ell p} = 0$. Then, by Lemma 3.4, $v_\ell^p \in p\mathfrak{O}$ and so $p\mathfrak{O} \mid v_\ell^p \mathfrak{O}$. Thus $p \mid N(v_\ell^p \mathfrak{O})$. Since $\mathfrak{O}$ is a commutative ring and the ideal norm is multiplicative, we have that $p \mid N(v_\ell \mathfrak{O})$. By Lemma 3.2, we obtain an effective bound on the divisors of $v_\ell \mathfrak{O}$ of the form $p\mathfrak{O}$ and hence a bound on the rational primes for which $u_{\ell p} = 0$ is possible. Mutatis mutandis the proof holds for prime powers p^k with $k > 1$. Clearly the case $k = 0$ is decided by determining whether $u_\ell = 0$. □

3.2 Complexity upper bound

Given a simple linear recurrence sequence (u_n), we establish a quantitative bound on the magnitude of any prime p such that $u_p = 0$. The bound is in terms of the size of the problem instance. In the case that (u_n) is a simple linear recurrence sequence, we know that $u_n = A_1\lambda_1^n + \cdots + A_m\lambda_m^n$ and so the size of the problem instance is the bit length $S = \|\langle \lambda_1, \lambda_2, \ldots, \lambda_m, A_1, A_2, \ldots, A_m \rangle\|$.

We give the following rudimentary bounds in terms of S. First, we bound $\log_2 |A_i|+1$, bit length of the integer A_i, from above by 2^S. Second, $|\lambda_i|$ is bounded from above by $H(\lambda_i) \leq 2^S$ where the height $H(\lambda_i)$ is the maximum absolute value of the coefficients in μ_{λ_i}. Finally, we have $\deg(\lambda_i) \leq S$, from which it follows that $[K\colon \mathbb{Q}] = [\mathbb{Q}(\lambda_1, \ldots, \lambda_m)\colon \mathbb{Q}] \leq m^S \leq S^S$. Because $u_1 = A_1\lambda_1 + \cdots A_m\lambda_m$ we have the following elementary bound

$$N(u_1\mathfrak{O}) \leq \prod_{\ell=1}^{[K\colon \mathbb{Q}]} \sum_{k=1}^{m} |\sigma_\ell(A_k)\sigma_\ell(\lambda_k)| \leq \prod_{\ell=1}^{[K\colon \mathbb{Q}]} S2^{3S} \leq \left(S2^{3S}\right)^{S^S}.$$

From the above calculations it follows that if $u_p = 0$ for some prime p then p is at most $(S2^{3S})^{S^S}$, i.e., double exponential in S, the size of the problem instance.

4 COEFFICIENTS IN $\mathfrak{O}[x]$

Let us first recall some background material on the decomposition of prime ideals in the ring of integers $\mathfrak{O}$ of a Galois number field K. Such decompositions (as products of powers of prime ideals) are particularly well-behaved in this setting— a comprehensive presentation of this material can be found in [4]. Let $p \in \mathbb{N}$ be prime. Then $p\mathfrak{O} = \prod_{i=1}^{g} \mathfrak{p}_i^e$ where the $\mathfrak{p}_i$ are the prime ideals lying above $p\mathbb{Z}$. Here the integer $e(p) \geq 1$ is the *ramification index* of p. The degree of the field extension $f(p) = [\mathfrak{O}/\mathfrak{p}_i : \mathbb{Z}/p\mathbb{Z}]$, the *inertial degree* of $\mathfrak{p}_i$ over $p\mathbb{Z}$, is independent of the prime ideal $\mathfrak{p}_i$. Suppose that $\mathfrak{p}$ lies above $p\mathbb{Z}$. We have $N(\mathfrak{p}) = N(p\mathbb{Z})^{f(p)} = p^{f(p)}$. A prime $p\mathbb{Z}$ is *ramified* in $\mathfrak{O}$ if $e > 1$ and *unramified* otherwise. In particular, only finitely many primes ramify in $\mathfrak{O}$ since $p\mathbb{Z}$ ramifies in $\mathfrak{O}$ if, and only if, p divides the discriminant of K (see e.g. [4]).

Suppose that K is Galois over $\mathbb{Q}$ and let $\mathfrak{O}$ be the algebraic integers in K. In this section we shall prove decidability results locating the zeroes of sequences (u_n) whose terms are given by an exponential polynomial of the form $u_n = A_1(n)\lambda_1^n + \cdots + A_m(n)\lambda_m^n$ with coefficients $A_1, \ldots, A_m \in \mathfrak{O}[x]$ and $\lambda_1, \ldots, \lambda_m \in \mathfrak{O}$. For such a sequence, fix $c \in \mathbb{N}$ and let $\mathcal{L}_c = \{\ell \in \mathbb{N} : \ell \leq c,\ v_\ell \neq 0\}$ where (v_n) is the simple recurrence sequence with terms given by $v_n = A_1(0)\lambda_1^n + \cdots + A_m(0)\lambda_m^n$. Let $f(p)$ be the inertial degree of $p\mathbb{Z}$ in $\mathfrak{O}$. Then define the set $\mathcal{N}_c(K)$ as the union

$$\mathcal{N}_c(K) = \bigcup_{\ell \in \mathcal{L}_c} \{\ell p^{kf(p)} : p \text{ prime},\ k \leq c\}.$$

Here our choice of notation is meant to draw comparison with our previous definition for the set $\mathcal{N}_c$. Without loss of generality we assume that given $c \in \mathbb{N}$ there is an $l \leq c$ such that $v_\ell \neq 0$ for otherwise the sequence (u_n) vanishes at $u_0 = v_0 = 0$. We denote by $\mathcal{Q}_c(K)$ the subset

$$\mathcal{Q}_c(K) = \bigcup_{\ell \in \mathcal{L}_c} \{\ell p^{kf(p)} : p\mathbb{Z} \text{ unramified},\ k \leq c\}.$$

Similarly, let $\mathcal{R}_c(K) \subset \mathcal{N}_c(K)$ be the corresponding set of elements where $p\mathbb{Z}$ is ramified in $\mathfrak{O}$. Since there are only finitely many prime ideals $p\mathbb{Z}$ that are ramified in $\mathfrak{O}$, the cardinality of the set $\mathcal{R}_c(K)$ is finite. By definition, $\mathcal{N}_c(K) = \mathcal{Q}_c(K) \cup \mathcal{R}_c(K)$.

Our main result is the following theorem.

Theorem 4.1. *Fix $c \in \mathbb{N}$. Given (u_n) as above, one can decide whether there is an $n \in \mathcal{N}_c(K)$ such that $u_n = 0$.*

Since the set $\mathcal{R}_c(K)$ is finite, locating zero terms $u_n = 0$ for $n \in \mathcal{R}_c(K)$ is clearly decidable. So to prove Theorem 4.1 it is sufficient to prove the next theorem.

Theorem 4.2. *Fix $c \in \mathbb{N}$. Given (u_n) as above, one can decide whether there is an $n \in \mathcal{Q}_c(K)$ such that $u_n = 0$.*

In order to prove Theorem 4.2, we first prove two technical results. The first, Lemma 4.3, concerns elements of cosets of $p\mathfrak{O}$ in $\mathfrak{O}$. The second, Lemma 4.4, plays an analogous rôle to that of Lemma 3.4 in Section 3.

Lemma 4.3. *Suppose that $\varphi \in \mathfrak{O}$ and $p\mathbb{Z}$ is non-zero prime ideal. If $p\mathbb{Z}$ is unramified with inertial degree $f(p)$ then $\varphi^{p^{f(p)}} - \varphi \in p\mathfrak{O}$.*

Proof. Write $p\mathfrak{O} = \mathfrak{p}_1 \cdots \mathfrak{p}_g$ for the unique factorisation of $p\mathfrak{O}$ as a product of the distinct prime ideals $\mathfrak{p}_i$ lying above $p\mathbb{Z}$. Here the ramification index is unity because $p\mathbb{Z}$ is unramified. By Theorem 2.1, for each $i \in \{1, \ldots, g\}$ and $\varphi \in \mathfrak{O}$ we have $\varphi^{N(\mathfrak{p}_i)} - \varphi \in \mathfrak{p}_i$. Since each of the exponents satisfy $N(\mathfrak{p}_i) = p^{f(p)}$, we deduce that $\varphi^{p^{f(p)}} - \varphi \in \cap_i \mathfrak{p}_i$. Because the distinct prime ideals $\mathfrak{p}_i$ are pairwise co-prime, we have $\cap_i \mathfrak{p}_i = \mathfrak{p}_1 \cdots \mathfrak{p}_g = p\mathfrak{O}$ and hence we have the desired result. □

Lemma 4.4. *Let (u_n) be a recurrence sequence and (v_n) the associated simple recurrence sequence as above. Let $p \in \mathbb{N}$ be a rational prime and $k, \ell \in \mathbb{N}$. If $p\mathbb{Z} \subset \mathfrak{O}$ is unramified with inertial degree $f(p)$ then $v_\ell - u_{\ell p^{kf(p)}} \in p\mathfrak{O}$.*

Proof. The result is a consequence of the next congruences

$$v_\ell \equiv v_{\ell p^{kf(p)}} \equiv u_{\ell p^{kf(p)}} \pmod{p\mathfrak{O}}.$$

The congruences hold trivially when $k = 0$. We shall prove the case $k = 1$ below and omit the case $k > 1$ as it follows similarly. The first congruence is a simple application of Lemma 4.3:

$$v_\ell = \sum_{j=1}^{m} A_j(0)\lambda_j^\ell \equiv \sum_{j=1}^{m} A_j(0)\lambda_j^{\ell p^{f(p)}} = v_{\ell p^{f(p)}} \pmod{p\mathfrak{O}}.$$

Recall that for $A \in \mathfrak{O}[x]$ we have $(x - y) \mid (A(x) - A(y))$. The second congruence holds since $p\mathfrak{O} \ni \ell p^{f(p)} \mid (A(\ell p^{f(p)}) - A(0))$

or equivalently $A(0) \equiv A(\ell p^{f(p)}) \pmod{p\mathfrak{O}}$ for each $A \in \mathfrak{O}[x]$. Thus

$$v_{\ell p^{f(p)}} \equiv \sum_{j=1}^{m} A_j\left(\ell p^{f(p)}\right)\lambda_j^{\ell p^{f(p)}} = u_{\ell p^{f(p)}} \pmod{p\mathfrak{O}}.$$

Hence $v_\ell - u_{\ell p^{f(p)}} \in p\mathfrak{O}$ as desired. □

Proof of Theorem 4.2. Fix $c \in \mathbb{N}$ and assume that $n \in \mathcal{Q}_c(K)$ such that $u_n = 0$. Then n is of the form $\ell p^{kf(p)}$ where p is a prime and $p\mathbb{Z} \subset \mathfrak{O}$ is unramified. By Lemma 4.4, $v_\ell - u_{\ell p^{kf(p)}} \in p\mathfrak{O}$. Thus $v_\ell \in p\mathfrak{O}$ and therefore $p\mathfrak{O} \mid v_\ell\mathfrak{O}$. We then apply Lemma 3.2 to give an effective bound on the primes by a divisibility argument for $N(v_\ell\mathfrak{O})$. Hence the result. □

Our approach in the proof of Theorem 4.1 extends in the following way: we can decide whether there exists there is an $n = \sum_{j=1}^{t} l_j p^{k_j f(p)}$ such that $u_n = 0$. Here the constants $k_j, l_j \in \mathbb{N}$ are bounded independently of the rational prime p, and $f(p)$ is the inertial degree of $p\mathbb{Z} \subset \mathfrak{O}$. For $l_1, \ldots, l_t, k_1, \ldots, k_t \in \mathbb{N}$, we define

$$S_m = S_m(l_j; k_j) := \begin{cases} \sum_{j=1}^{t} l_j m^{k_j f(m)} & \text{if } m \text{ is prime,} \\ \sum_{j=1}^{t} l_j & \text{if } m = 1. \end{cases}$$

Fix $c \in \mathbb{N}$ and, as before, let $\mathcal{L}_c = \{\ell \in \mathbb{N} \colon \ell \le c,\ v_\ell \neq 0\}$. Define the set $\mathcal{N}'_c(K)$ as follows

$$\mathcal{N}'_c(K) = \bigcup_{S_1 \in \mathcal{L}_c} \left\{S_p(l_j; k_j) \colon p \text{ prime},\ k_j \le c\right\}.$$

We define the sets $\mathcal{Q}'_c(K)$, for unramified $p\mathbb{Z}$ in K, and $\mathcal{R}'_c(K)$, for ramified $p\mathbb{Z}$ in K, in an analogous manner to the sets $\mathcal{Q}_c(K)$ and $\mathcal{R}_c(K)$ associated to $\mathcal{N}_c(K)$. Then, like before, $\mathcal{N}'_c(K) = \mathcal{Q}'_c(K) \cup \mathcal{R}'_c(K)$ and $\mathcal{R}'_c(K)$ has finite cardinality.

We have the next decidability result.

Theorem 4.5. *Fix $c \in \mathbb{N}$. Then, given (u_n) as above, one can decide whether there is an $n \in \mathcal{N}'_c(K)$ such that $u_n = 0$.*

The proof of Theorem 4.5 follows the approach in the proof of Theorem 4.1. Since the cardinality of $\mathcal{R}'_c(K)$ is finite, we need only prove the next theorem in order to prove Theorem 4.5.

Theorem 4.6. *Fix $c \in \mathbb{N}$. Then, given (u_n) as above, one can decide whether there is an $n \in \mathcal{Q}'_c(K)$ such that $u_n = 0$.*

Given its similarities to the proof of Theorem 4.2, we omit a formal proof of Theorem 4.6; instead, we outline the key steps in the proof. We require the following technical lemma; Lemma 4.7 generalises the result in Lemma 4.4.

Lemma 4.7. *Let (u_n) be a recurrence sequence and (v_n) the associated simple recurrence sequence as above. Let $p \in \mathbb{N}$ be a rational prime and $S_p(l_j; k_j)$ be defined as above. If $p\mathbb{Z} \subset \mathfrak{O}$ is unramified then $u_{S_p} - v_{S_1} \in p\mathfrak{O}$.*

Proof. We avoid repeating the proof of Lemma 4.4 by limiting our presentation to the next two observations. First, for each polynomial $A \in \mathfrak{O}[x]$ we have $A(S_p) - A(0) \in p\mathfrak{O}$ since $p\mathfrak{O} \ni S_p$ divides $A(S_p) - A(0)$. Second, by repeated application of Lemma 4.3, we have $\lambda^{S_p} - \lambda^{S_1} \in p\mathfrak{O}$ for $\lambda \in \mathfrak{O}$. From these observations, one can obtain the congruences $v_{S_1} \equiv v_{S_p} \equiv u_{S_p} \pmod{p\mathfrak{O}}$ and hence the desired result. □

We sketch the key steps in the proof of Theorem 4.6.

Proof of Theorem 4.6. Fix $c \in \mathbb{N}$. Assume that $u_{S_p} = 0$ for some $S_p(l_j; k_j) \in \mathcal{N}'_c(K)$ where $p\mathbb{Z} \subset \mathfrak{O}$ is an unramified prime. Note that $v_{S_1} \neq 0$ since $S_p(l_j; k_j) \in \mathcal{N}'_c(K)$. Then, by Lemma 4.7, $v_{S_1} \in p\mathfrak{O}$ and so $p\mathfrak{O} \mid v_{S_1}\mathfrak{O}$. By Lemma 3.2, p necessarily divides $N(v_{S_1}\mathfrak{O})$. Since $N(v_{S_1}\mathfrak{O})$ is computable, one can derive an effective bound on the rational primes p such that $u_{S_p} = 0$. □

5 HARDNESS RESULT

In [3], Blondel and Portier proved that the Skolem Problem is NP-hard (see also [1]). In this section we show that the prime variant of the Skolem Problem is likewise NP-hard. Following [1], our proof is by reduction from the *Subset Sum Problem*: given a finite set of integer $A = \{a_1, \ldots, a_m\}$ and $b \in \mathbb{Z}$ a target, written in binary, decide whether there is a subset $S \subseteq \{1, \ldots, m\}$ such that $\sum_{k \in S} a_k = b$.

Let us state two well-known theorems in number theory in order to derive a simple corollary that is fundamental to our proof of Theorem 5.6.

Theorem 5.1 (Chinese remainder theorem). *Let $n_1, \ldots, n_m$ be positive integers that are pairwise co-prime. Then the system of m equations $r \equiv a_k \pmod{n_k}$ with each $a_k \in \mathbb{Z}$ has a unique solution modulo N where $N = n_1 n_2 \cdots n_m$.*

Dirichlet proved the following theorem on primes in arithmetic progressions. We use the notation (m, n) to indicate the greatest common divisor of $m, n \in \mathbb{Z}$.

Theorem 5.2. *Suppose that q and r are co-prime positive integers. Then there are infinitely many primes of the form $\ell q + r$ with $\ell \in \mathbb{N}$.*

The next corollary is immediate.

Corollary 5.3. *Let $p_1, \ldots, p_m$ be a finite set of distinct primes. Then the system of m equations $r \equiv a_k \pmod{p_k}$ with each $a_k \in \mathbb{Z}$ has a unique solution $r \in \{0, 1, \ldots, P-1\}$ where $P = p_1 p_2 \cdots p_m$. Additionally, if $(r, P) = 1$ then there are infinitely many $\ell \in \mathbb{N}$ for which $\ell P + r$ is prime.*

Recall that the *nth cyclotomic polynomial* given by

$$\Phi_n(x) = \prod_{\substack{k \in \{1,\ldots,n\} \\ (k,n)=1}} \left(x - \mathrm{e}^{2\pi \mathrm{i} k/n}\right)$$

is the minimal polynomial over $\mathbb{Q}$ of a primitive nth root of unity.

We call an integer linear recurrence sequence *cyclotomic* if its characteristic roots are all roots of unity. The next theorem, concerning Skolem's Problem in the restricted setting of cyclotomic sequences, follows from work in [1]. We reproduce the proof as a lead into our original work on the Skolem Problem restricted to prime numbers.

Theorem 5.4. *The cyclotomic Skolem Problem is* NP*-hard.*

The proof of Theorem 5.4 is by reduction from the Subset Sum Problem and follows directly from the technical lemma, Lemma 5.5, below. Before we present the proof, we introduce some notation.

Let $\{p_1, \ldots, p_m\}$ be the set of the first m prime numbers. We define the linear recurrence sequence $(s_k(n))_{n=0}^{\infty}$ with $k \in \{1, \ldots, m\}$ as follows. Let $s_k(n) = s_k(n - p_k)$ for $n \ge p_k$ with initial conditions

$s_k(0) = 1, s_k(1) = \cdots = s_k(p_k - 1) = 0$. Then each sequence $(s_k(n))$ is periodic with period p_k. The characteristic polynomial associated to $(s_k(n))$ is given by

$$x^{p_k} - 1 = \prod_{\ell=0}^{p_k-1} \left(x - \mathrm{e}^{2\pi \mathrm{i}\ell/p_k}\right).$$

Thus $(s_k(n))$ is a cyclotomic sequence.

In order to reduce the Subset Sum Problem to the cyclotomic Skolem Problem, we consider the inhomogeneous linear recurrence sequence $(t(n))_{n=0}^{\infty}$ with terms given by $t(n) = b - \sum_{k=1}^{m} a_k s_k(n)$. The characteristic polynomial associated to $(t(n))$ is given by the least common multiple of

$$(x^{p_1} - 1)(x - 1), x^{p_2} - 1, \ldots, x^{p_m} - 1$$

(see [6]), from which it follows that each of the characteristic roots of $(t(n))$ are themselves roots of unity, i.e., $(t(n))$ is a cyclotomic sequence.

Lemma 5.5. *For $(t(n))$ given as above, there exists $N \in \mathbb{N}$ such that $t(N) = 0$ if and only if the Subset Sum Problem with inputs $\{a_1, \ldots, a_m; b\}$ has a solution.*

Proof. Suppose that there exists an $N \in \mathbb{N}$ such that $t(N) = 0$, then the Subset Sum Problem has a solution because the selectors $s_k(n)$ are $\{0, 1\}$-valued. Conversely, suppose that there is a subset $S \subseteq \{1, \ldots, m\}$ such that $\sum_{k \in S} a_k = b$ and define $N = \prod_{k \in S} p_k$. We have $s_k(N) = 1$ for each $k \in S$ since $p_k \mid N$, and $s_k(N) = 0$ otherwise. Thus

$$t(N) = b - \sum_{k=1}^{m} a_k s_k(N) = b - \sum_{k \in S} a_k = 0,$$

as required. □

We prove the following complexity result for the Skolem Problem for primes.

Theorem 5.6. *Suppose that (u_n) is a cyclotomic integer linear recurrence sequence. The problem of deciding whether there is a prime $p \in \mathbb{N}$ such that $u_p = 0$ is* NP*-hard.*

The proof of Theorem 5.6 involves an analysis of the NP-hardness proof for Skolem's Problem. Technically we will derive the result from Lemma 5.7, below.

Let $p_1, \ldots, p_m$ be the first m odd primes. We define selector sequences $(\sigma_k(n))$ with $k \in \{1, \ldots, m\}$ as follows. Let $\sigma_k(n) = \sigma_k(n - p_k)$ for $n \geq p_k$ with initial conditions $\sigma_k(1) = 1$, $\sigma_k(0) = \sigma_k(2) = \cdots = \sigma_k(p_k - 1) = 0$. Then each sequence $(\sigma_k(n))$ is periodic with period p_k. Let $\tau(n) = b - \sum_{k=1}^{m} a_k \sigma_k(n)$. It is easily shown that $(\sigma_k(n))$ and $(\tau(n))$ are cyclotomic recurrence sequences.

Lemma 5.7. *There exists an odd prime $p \in \mathbb{N}$ such that $\tau(p) = 0$ if and only if there exists a subset $S \subseteq \{1, \ldots, m\}$ that is a solution to the Subset Sum Problem with inputs $\{a_1, \ldots, a_m; b\}$.*

Proof. Suppose that there is an odd prime $p \in \mathbb{N}$ such that $\tau(p) = 0$. Then there is a solution to the Subset Sum Problem as $\sigma_k(p) \in \{0, 1\}$ for each k.

Conversely, suppose that there a subset $S \subseteq \{1, \ldots, m\}$ such that $\sum_{k \in S} a_k = b$. Consider the set $Q(S) \subseteq \mathbb{Z}$ of integer solutions to the set of m equations

$$\begin{cases} r \equiv 1 \pmod{p_k} & \text{if } k \in S, \quad \text{and} \\ r \equiv 2 \pmod{p_k} & \text{if } k \in \{1, \ldots, m\} \setminus S. \end{cases}$$

The choice of residue ensures that r is not divisible by any of the primes $p_1, p_2, \ldots, p_m$. By the Chinese Remainder Theorem, $Q(S)$ is an infinite arithmetic progression. Suppose that $q \in Q(S)$. Then, by definition of the selector sequences, $\sigma_k(q) = 1$ if and only if $q \equiv 1 \pmod{p_k}$ if and only if $k \in S$. Then

$$\tau(q) = b - \sum_{k=1}^{m} a_k \sigma_k(q) = b - \sum_{k \in S} a_k = 0.$$

It remains to show that there is a prime number in $Q(S)$. This result follows easily from Corollary 5.3, which completes the proof. □

6 SUMMARY

In this paper we have given decision procedures for finding zeroes of certain prescribed linear recurrence sequences. Our main result shows how to decide the existence of a prime p such that $u_p = 0$ for a simple linear recurrence sequence (u_n). We have noted that this decision problem is NP-hard and, implicitly, that the magnitude of the smallest prime p such that $u_p = 0$ is at least exponential in the size of the problem instance. On the other hand, our decision procedure yields a double exponential bound on the magnitude of the prime p. Closing this exponential gap would be an interesting direction for further work. Another direction for research would be to locate zeroes $u_n = 0$ where the index $n \in \mathbb{N}$ has two prime factors.

REFERENCES

[1] S. Akshay, Nikhil Balaji, and Nikhil Vyas. 2017. Complexity of Restricted Variants of Skolem and Related Problems. In *42nd International Symposium on Mathematical Foundations of Computer Science (MFCS 2017) (Leibniz International Proceedings in Informatics (LIPIcs))*, K. Larsen, H. Bodlaender, and J-F. Raskin (Eds.), Vol. 83. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik, Dagstuhl, Germany, 78:1–78:14. https://doi.org/10.4230/LIPIcs.MFCS.2017.78

[2] Jean Berstel and Maurice Mignotte. 1976. Deux propriétés décidables des suites récurrentes linéaires. *Bulletin de la Société Mathématique de France* 104, 2 (1976), 175–184.

[3] Vincent D. Blondel and Natacha Portier. 2002. The presence of a zero in an integer linear recurrent sequence is NP-hard to decide. *Linear Algebra Appl.* 351/352 (2002), 91–98. https://doi.org/10.1016/S0024-3795(01)00466-9 Fourth special issue on linear systems and control.

[4] Henri Cohen. 1993. *A course in computational algebraic number theory*. Graduate Texts in Mathematics, Vol. 138. Springer-Verlag, Berlin. xii+534 pages.

[5] Harm Derksen. 2007. A Skolem-Mahler-Lech theorem in positive characteristic and finite automata. *Inventiones Mathematicae* 168, 1 (2007), 175–224.

[6] Graham Everest, Alf van der Poorten, Igor Shparlinski, and Thomas Ward. 2003. *Recurrence sequences*. Mathematical Surveys and Monographs, Vol. 104. Amer. Math. Soc., Providence, RI. xiv+318 pages.

[7] Vesa Halava, Tero Harju, Mika Hirvensalo, and Juhani Karhumäki. 2005. *Skolem's problem–on the border between decidability and undecidability.* Technical Report. Turku Centre for Computer Science.

[8] Christer Lech. 1953. A note on recurring series. *Arkiv för Matematik* 2 (1953), 417–421.

[9] K. Mahler. 1935. Eine arithmetische Eigenschaft der Taylor-koeffizienten rationaler Funktionen. *Proc. Akad. Wet. Amst.* 38 (1935), 50–69.

[10] K. Mahler and J. Cassels. 1956. On the Taylor coefficients of rational functions. *Mathematical Proceedings of the Cambridge Philosophical Society* 52, 1 (1956), 39–48.

[11] Maurice Mignotte, Tarlok Shorey, and Robert Tijdeman. 1984. The distance between terms of an algebraic recurrence sequence. *Journal für die Reine und Angewandte Mathematik* (1984), 63–76.
[12] Joël Ouaknine and James Worrell. 2012. Decision problems for linear recurrence sequences. In *Reachability problems*. Lecture Notes in Computer Science, Vol. 7550. Springer, Heidelberg, 21–28.
[13] Joël Ouaknine and James Worrell. 2015. On Linear Recurrence Sequences and Loop Termination. *ACM SIGLOG News* 2, 2 (April 2015), 4–13.
[14] Thoralf Skolem. 1934. Ein Verfahren zur Behandlung gewisser exponentialer Gleichungen und diophantischer Gleichungen. *8de Skand. Mat. Kongress, Stockholm (1934)* (1934), 163–188.
[15] Nikolai Vereshchagin. 1985. Occurrence of zero in a linear recursive sequence. *Mathematical notes of the Academy of Sciences of the USSR* 38, 2 (01 Aug 1985), 609–615.

Computing the Real Isolated Points of an Algebraic Hypersurface

Huu Phuoc Le
Sorbonne Université, CNRS,
Laboratoire d'Informatique de Paris 6,
LIP6, Équipe PolSys
F-75252, Paris Cedex 05, France
huu-phuoc.le@lip6.fr

Mohab Safey El Din
Sorbonne Université, CNRS,
Laboratoire d'Informatique de Paris 6,
LIP6, Équipe PolSys
F-75252, Paris Cedex 05, France
mohab.safey@lip6.fr

Timo de Wolff
Technische Universität Braunschweig,
Institut für Analysis und Algebra, AG Algebra
38106 Braunschweig, Germany
t.de-wolff@tu-braunschweig.de

ABSTRACT

Let $\boldsymbol{R}$ be the field of real numbers. We consider the problem of computing the real isolated points of a real algebraic set in $\boldsymbol{R}^n$ given as the vanishing set of a polynomial system. This problem plays an important role for studying rigidity properties of mechanism in material designs. In this paper, we design an algorithm which solves this problem. It is based on the computations of critical points as well as roadmaps for answering connectivity queries in real algebraic sets. This leads to a probabilistic algorithm of complexity $(nd)^{O(n\log(n))}$ for computing the real isolated points of real algebraic hypersurfaces of degree d. It allows us to solve in practice instances which are out of reach of the state-of-the-art.

CCS CONCEPTS

• **Theory of computation → Computational geometry**; • **Computing methodologies → Symbolic and algebraic algorithms**.

KEYWORDS

Semi-algebraic sets; Critical point method; Real algebraic geometry; Auxetics; Rigidity

ACM Reference Format:
Huu Phuoc Le, Mohab Safey El Din, and Timo de Wolff. 2020. Computing the Real Isolated Points of an Algebraic Hypersurface. In *International Symposium on Symbolic and Algebraic Computation (ISSAC '20), July 20–23, 2020, Kalamata, Greece.* ACM, New York, NY, USA, 8 pages. https://doi.org/10.1145/3373207.3404049

1 INTRODUCTION

Let $\boldsymbol{Q}$, $\boldsymbol{R}$ and $\boldsymbol{C}$ be respectively the fields of rational, real and complex numbers. For $\boldsymbol{x} \in \boldsymbol{R}^n$ and $r \in \boldsymbol{R}$, we denote by $B(\boldsymbol{x}, r) \subset \boldsymbol{R}^n$ the open ball centered at $\boldsymbol{x}$ of radius r.

Mohab Safey El Din and Huu Phuoc Le are supported by the ANR grants ANR-18-CE33-0011 Sesame, and ANR-19-CE40-0018 De Rerum Natura, the joint ANR-FWF ANR-19-CE48-0015 ECARP project, the PGMO grant CAMiSAdo and the European Union's Horizon 2020 research and innovative training network programme under the Marie Skłodowska-Curie grant agreement N° 813211 (POEMA). Timo de Wolff is supported by the DFG grant WO 2206/1-1.

ISSAC '20, July 20–23, 2020, Kalamata, Greece

ACM ISBN 978-1-4503-7100-1/20/07…$15.00
https://doi.org/10.1145/3373207.3404049

Let $f \in \boldsymbol{Q}[x_1, \ldots, x_n]$ and $\mathcal{H} \subset \boldsymbol{C}^n$ be the hypersurface defined by $f = 0$. We aim at computing the *isolated points* of $\mathcal{H} \cap \boldsymbol{R}^n$, i.e. the set of points $\boldsymbol{x} \in \mathcal{H} \cap \boldsymbol{R}^n$ s.t. for some positive r, $B(\boldsymbol{x}, r) \cap \mathcal{H} = \{\boldsymbol{x}\}$. We shall denote this set of isolated real points by $\mathscr{Z}(\mathcal{H})$.

Motivation. We consider here a particular instance of the more general problem of computing the isolated points of a *semi-algebraic* set. Such problems arise naturally and frequently in the design of rigid mechanism in material design. Those are modeled canonically with semi-algebraic constraints, and isolated points to the semi-algebraic set under consideration are related to mobility/rigidity properties of the mechanism. A particular example is the study of *auxetic* materials, i.e., materials that shrink in all directions under compression. These materials appear in nature (first discovered in [20]) e.g., in foams, bones or propylene; see e.g. [32], and have various potential applications. They are an active field of research, not only on the practical side, e.g., [11, 16], but also with respect to mathematical foundations; see e.g. [5, 6]. On the constructive side, these materials are closely related to *tensegrity frameworks*, e.g., [8, 21], which can possess various sorts of rigidity properties.

Hence, we aim to provide a practical algorithm for computing these real isolated points in the particular case of real traces of complex hypersurfaces first. This simplification allows us to significantly improve the state-of-the-art complexity for this problem and to establish a new algorithmic framework for such computations.

State-of-the-art. As far as we know, there is no established algorithm dedicated to the problem under consideration here. However, effective real algebraic geometry provides subroutines from which such a computation could be done. Let $\mathcal{H}$ be a hypersurface defined by $f = 0$ with $f \in \boldsymbol{Q}[x_1, \ldots, x_n]$ of degree d.

A first approach would be to compute a cylindrical algebraic decomposition adapted to $\mathcal{H} \cap \boldsymbol{R}^n$ [7]. It partitions $\mathcal{H} \cap \boldsymbol{R}^d$ into connected *cells*, i.e. subsets which are homeomorphic to $]0, 1[^i$ for some $1 \leq i \leq n$. Next, one needs to identify cells which correspond to isolated points using adjacency information (see e.g. [1]). Such a procedure is at least doubly exponential in n and polynomial in d.

A better alternative is to encode real isolated points with quantified formula over the reals. Using e.g. [2, Algorithm 14.21], one can compute isolated points of $\mathcal{H} \cap \boldsymbol{R}^n$ in time $d^{O(n^2)}$. Note also that [31] allows to compute isolated points in time $d^{O(n^3)}$.

A third alternative (suggested by the reviewers) is to use [2, Algorithm 12.16] to compute sample points in each connected component of $\mathcal{H} \cap \boldsymbol{R}^n$ and then decide whether spheres, centered at these points, of infinitesimal radius, meet $\mathcal{H} \cap \boldsymbol{R}^n$. Note that these

points are encoded with parametrizations of degree $d^{O(n)}$ (their coordinates are evaluations of polynomials at the roots of a univariate polynomial with infinitesimal coefficients). Applying [2, Alg. 12.16] on this last real root decision problem would lead to a complexity $d^{O(n^2)}$ since the input polynomials would have degree $d^{O(n)}$. Another approach would be to run [2, Alg. 12.16] modulo the algebraic extension used to define the sample points. That would lead to a complexity $d^{O(n)}$ but this research direction requires modifications of [2, Alg. 12.16] since it assumes the input coefficients to lie in an *integral domain*, which is not satisfied in our case. Besides, we report on practical experiments showing that using [2, Alg. 12.16] to compute *only* sample points in $\mathcal{H} \cap \boldsymbol{R}^n$ does not allow us to solve instances of moderate size.

The topological nature of our problem is related to connectedness. Computing isolated points of $\mathcal{H} \cap \boldsymbol{R}^n$ is equivalent to computing those connected components of $\mathcal{H} \cap \boldsymbol{R}^n$ which are reduced to a single point (see Lemma 1). Hence, one considers computing *roadmaps*: these are algebraic curves contained in $\mathcal{H}$ which have a non-empty and connected intersection with all connected components of the real set under study. Once such a roadmap is computed, it suffices to compute the isolated points of a semi-algebraic curve in $\boldsymbol{R}^n$. This latter step is not trivial; as many of the algorithms computing roadmaps output either curve segments (see e.g., [4]) or algebraic curves (see e.g., [28]). Such curves are encoded through *rational parametrizations*, i.e., as the Zariski closure of the projection of the $(x_1, \ldots, x_n)$-space of the solution set to

$$w(t, s) = 0, x_i = v_i(t, s)/\frac{\partial w}{\partial t}(t, s), \quad 1 \leq i \leq n$$

where $w \in \boldsymbol{Q}[t, s]$ is square-free and monic in t and s and the v_i's lie in $\boldsymbol{Q}[t, s]$ (see e.g., [28]). As far as we know, there is no published algorithm for computing isolated points from such an encoding.

Computing roadmaps started with Canny's (probabilistic) algorithm running in time $d^{O(n^2)}$ on real algebraic sets. Later on, [27] introduced new types of connectivity results enabling more freedom in the design of roadmap algorithms. This led to [4, 27] for computing roadmaps in time $(nd)^{O(n^{1.5})}$. More recently, [3], still using these new types of connectivity results, provide a roadmap algorithm running in time $d^{O(n \log^2 n)} n^{O(n \log^3 n)}$ for general real algebraic sets (at the cost of introducing a number of infinitesimals). This is improved in [28], for smooth bounded real algebraic sets, with a probabilistic algorithm running in time $O((nd)^{12n \log_2 n})$. These results makes plausible to obtain a full algorithm running in time $(nd)^{O(n \log n)}$ to compute the isolated points of $\mathcal{H} \cap \boldsymbol{R}^n$.

Main result. We provide a probabilistic algorithm which takes as input f and computes the set of real isolated points $\mathscr{Z}(\mathcal{H})$ of $\mathcal{H} \cap \boldsymbol{R}^n$. A few remarks on the output data-structure are in order. Any finite algebraic set $Z \subset \boldsymbol{C}^n$ defined over $\boldsymbol{Q}$ can be represented as the projection on the $(x_1, \ldots, x_n)$-space of the solution set to

$$w(t) = 0, x_i = v_i(t), \quad 1 \leq i \leq n$$

where $w \in \boldsymbol{Q}[t]$ is square-free and the v_i's lie in $\boldsymbol{Q}[t]$. The sequence of polynomials $(w, v_1, \ldots, v_n)$ is called a *zero-dimensional parametrization*; such a representation goes back to Kronecker [19]. Such representations (and their variants with denominators) are widely used in computer algebra (see e.g. [12–14]). For a zero-dimensional parametrization $\mathfrak{Q}$, $Z(\mathfrak{Q}) \subset \boldsymbol{C}^n$ denotes the finite set represented by $\mathfrak{Q}$. Observe that considering additionally isolating boxes, one can encode $Z(\mathfrak{Q}) \cap \boldsymbol{R}^n$. Our main result is as follows.

Theorem 1. *Let $f \in \boldsymbol{Q}[x_1, \ldots, x_n]$ of degree d and $\mathcal{H} \subset \boldsymbol{C}^n$ be the algebraic set defined by $f = 0$. There exists a probabilistic algorithm which, on an input f of degree d, computes a zero-dimensional parametrization $\mathfrak{P}$ and isolating boxes which encode $\mathscr{Z}(\mathcal{H})$ using $(nd)^{O(n \log(n))}$ arithmetic operations in $\boldsymbol{Q}$.*

In Section 5, we report on practical experiments showing that it already allows us to solve non-trivial problems which are actually out of reach of [2, Alg. 12.16] to compute sample points in $\mathcal{H} \cap \boldsymbol{R}^n$ only. We sketch now the geometric ingredients which allow us to obtain such an algorithm. Assume that f is non-negative over $\boldsymbol{R}^n$ (if this is not the case, just replace it by its square) and let $\boldsymbol{x} \in \mathscr{Z}(\mathcal{H})$. Since $\boldsymbol{x}$ is isolated and f is non-negative over $\boldsymbol{R}^n$, the intuition is that for $e > 0$ and small enough, the real solution set to $f = e$ looks like a ball around $\boldsymbol{x}$, hence a bounded and closed connected component $C_{\boldsymbol{x}}$. Then the restriction of every projection on the x_i-axis to the algebraic set $\mathcal{H}_e \subset \boldsymbol{C}^n$ defined by $f = e$ intersects $C_{\boldsymbol{x}}$. When e tends to 0, these critical points in $C_{\boldsymbol{x}}$ "tend to $\boldsymbol{x}$". This first process allows us to compute a superset of candidate points in $\mathcal{H} \cap \boldsymbol{R}^n$ containing $\mathscr{Z}(\mathcal{H})$. Of course, one would like that this superset is finite and this will be the case up to some generic linear change of coordinates, using e.g. [25].

All in all, at this stage we have "candidate points" that may lie in $\mathscr{Z}(\mathcal{H})$. Writing a quantified formula to decide if there exists a ball around these points which does not meet $\mathcal{H} \cap \boldsymbol{R}^n$ raises complexity issues (those points are encoded by zero-dimensional parametrizations of degree $d^{O(n)}$, given as input to a decision procedure).

Hence we need new ingredients. Note that our "candidate points" lie on "curves of critical points" which are obtained by letting e vary in the polynomial systems defining the aforementioned critical points. Assume now that $\mathcal{H} \cap \boldsymbol{R}^n$ is bounded, hence contained in a ball B. Then, for e' small enough, the real algebraic set defined by $f = 0$ is "approximated" by the union of the connected components of the real set defined by $f = e'$ which are contained in B. Besides, these "curves of critical points", that we just mentioned, hit these connected components when one fixes e'. We actually prove that two distinct points of our set of "candidate points" are connected through these "curves of critical points" and those connected components defined by $f = e'$ in B if and only if they do not lie in $\mathscr{Z}(\mathcal{H})$. Hence, we use computations of roadmaps of the real set defined by $f = e'$ to answer those connectivity queries. Then, advanced algorithms for roadmaps and polynomial system solving allows us to achieve the announced complexity bound.

Many details are hidden in this description. In particular, we use infinitesimal deformations and techniques of semi-algebraic geometry. While infinitesimals are needed for proofs, they may be difficult to use in practice. On the algorithmic side, we go further exploiting the geometry of the problem to avoid using infinitesimals.

Structure of the paper. In Section 2, we study the geometry of our problem and prove a series of auxiliary results (in particular Proposition 7, which coins the theoretical ingredient we need). Section 3 is devoted to describe the algorithm. Section 4 is devoted to the complexity analysis and Section 5 reports on the practical performances of our algorithm.

Acknowledgments. We thank the reviewers for their helpful comments.

2 THE GEOMETRY OF THE PROBLEM

2.1 Candidates for isolated points

As above, let $f \in \mathbf{Q}[x_1, \ldots, x_n]$ and $\mathcal{H} \subset \mathbf{C}^n$ be the hypersurface defined by $f = 0$. Let $\mathbf{f}$ be a subset of $\mathbf{C}[x_1, \ldots, x_n]$, we denote by $V(\mathbf{f})$ the simultaneous vanishing locus in $\mathbf{C}^n$ of $\mathbf{f}$.

Lemma 1. *The set $\mathscr{Z}(\mathcal{H})$ is the (finite) union of the semi-algebraically connected components of $\mathcal{H} \cap \mathbf{R}^n$ which are a singleton.*

Proof. Recall that real algebraic sets have a finite numner of semi-algebraically connected components [2, Theorem 5.21]. Let C be a semi-algebraically connected component of $\mathcal{H} \cap \mathbf{R}^n$.

Assume that C is not a singleton and take $\boldsymbol{x}$ and $\boldsymbol{y}$ in C with $\boldsymbol{x} \neq \boldsymbol{y}$. Then, there exists a semi-algebraic continuous map $\gamma : [0, 1] \to C$ s.t. $\gamma(0) = \boldsymbol{x}$ and $\gamma(1) = \boldsymbol{y}$; besides, since $\boldsymbol{x} \neq \boldsymbol{y}$, there exist $t \in (0, 1)$ such that $\gamma(t) \neq \boldsymbol{x}$. By continuity of γ and the norm function, any ball B centered at $\boldsymbol{x}$ contains a point $\gamma(t) \neq \boldsymbol{x}$.

Now assume that $C = \{\boldsymbol{x}\}$. Observe that $\mathcal{H} \cap \mathbf{R}^n - \{\boldsymbol{x}\}$ is closed (since semi-algebraically connected components of real algebraic sets are closed). Since $\mathcal{H} \cap \mathbf{R}^n$ is bounded, we deduce that $\mathcal{H} \cap \mathbf{R}^n - \{\boldsymbol{x}\}$ is closed and bounded. Then, the map $\boldsymbol{y} \to \|\boldsymbol{y} - \boldsymbol{x}\|^2$ reaches a minimum over $\mathcal{H} \cap \mathbf{R}^n - \{\boldsymbol{x}\}$. Let e be this minimum value. We deduce that any ball centered at $\boldsymbol{x}$ of radius less than e does not meet $\mathcal{H} \cap \mathbf{R}^n - \{\boldsymbol{x}\}$. □

To compute those connected components of $\mathcal{H} \cap \mathbf{R}^n$ which are singletons, we use classical objects of optimization and Morse theory which are mainly *polar varieties*. Let $\boldsymbol{K}$ be an algebraically closed field, let $\phi \in \boldsymbol{K}[x_1, \ldots, x_n]$ which defines the polynomial mapping $(x_1, \ldots, x_n) \mapsto \phi(x_1, \ldots, x_n)$ and $V \subset \boldsymbol{K}^n$ be a smooth equidimensional algebraic set. We denote by $W(\phi, V)$ the set of critical points of the restriction of ϕ to V. If c is the co-dimension of V and $(g_1, \ldots, g_s)$ generates the vanishing ideal associated to V, then $W(\phi, V)$ is the subset of V at which the Jacobian matrix associated to $(g_1, \ldots, g_s, \phi)$ has rank less than or equal to c (see e.g., [28, Subsection 3.1]).

In particular, the case where ϕ is replaced by the canonical projection on the i-th coordinate

$$\pi_i : (x_1, \ldots, x_n) \mapsto x_i,$$

is excessively used throughout our paper.

In our context, we do not assume that $\mathcal{H}$ is smooth. Hence, to exploit strong topological properties of polar varieties, we retrieve a smooth situation using deformation techniques. We consider an infinitesimal ε, i.e., a transcendental element over $\boldsymbol{R}$ such that $0 < \varepsilon < r$ for any positive element $r \in \boldsymbol{R}$, and the field of Puiseux series over $\boldsymbol{R}$, denoted by

$$\boldsymbol{R}\langle\varepsilon\rangle = \left\{ \textstyle\sum_{i \geq i_0} a_i \varepsilon^{i/q} \mid i \in \mathbb{N}, i_0 \in \mathbb{Z}, q \in \mathbb{N} - \{0\}, a_i \in \boldsymbol{R} \right\}.$$

Recall that $\boldsymbol{R}\langle\varepsilon\rangle$ is a real closed field [2, Theorem 2.91]. One defines $\boldsymbol{C}\langle\varepsilon\rangle$ as for $\boldsymbol{R}\langle\varepsilon\rangle$ but taking the coefficients of the series in $\boldsymbol{C}$. Recall that $\boldsymbol{C}\langle\varepsilon\rangle$ is an algebraic closure of $\boldsymbol{R}\langle\varepsilon\rangle$ [2, Theorem 2.17]. Consider $\sigma = \sum_{i \geq i_0} a_i \varepsilon^{i/q} \in \boldsymbol{R}\langle\varepsilon\rangle$ with $a_{i_0} \neq 0$. Then, a_{i_0} is called the *valuation* of σ. When $i_0 \geq 0$, σ is said to be *bounded over* $\boldsymbol{R}$ and the set of bounded elements of $\boldsymbol{R}\langle\varepsilon\rangle$ is denoted by $\boldsymbol{R}\langle\varepsilon\rangle_b$. One defines the function $\lim_\varepsilon : \boldsymbol{R}\langle\varepsilon\rangle_b \to \boldsymbol{R}$ that maps σ to a_0 (which is 0 when $i_0 > 0$) and writes $\lim_\varepsilon \sigma = a_0$; note that $\lim_\varepsilon$ is a ring homomorphism from $\boldsymbol{R}\langle\varepsilon\rangle_b$ to $\boldsymbol{R}$. All these definitions extend to $\boldsymbol{R}\langle\varepsilon\rangle^n$ componentwise. For a semi-algebraic set $\mathcal{S} \subset \boldsymbol{R}\langle\varepsilon\rangle^n$, we naturally define the limit of $\mathcal{S}$ as $\lim_\varepsilon \mathcal{S} = \{\lim_\varepsilon \boldsymbol{x} \mid \boldsymbol{x} \in \mathcal{S} \text{ and } \boldsymbol{x} \text{ is bounded over } \boldsymbol{R}\}$.

Let $\mathcal{S} \subset \boldsymbol{R}^n$ be a semi-algebraic set defined by a semi-algebraic formula Φ. We denote by $\mathrm{ext}(\mathcal{S}, \boldsymbol{R}\langle\varepsilon\rangle)$ the semi-algebraic set of points which are solutions of Φ in $\boldsymbol{R}\langle\varepsilon\rangle^n$. We refer to [2, Chap. 2] for more details on infinitesimals and real Puiseux series.

By e.g., [22, Lemma 3.5], $\mathcal{H}_\varepsilon$ and $\mathcal{H}_{-\varepsilon}$ respectively defined by $f = \varepsilon$ and $f = -\varepsilon$ are two disjoint smooth algebraic sets in $\boldsymbol{C}\langle\varepsilon\rangle^n$.

Lemma 2. *For any $\boldsymbol{x}$ lying in a bounded connected component of $\mathcal{H} \cap \mathbf{R}^n$, there exists a point $\boldsymbol{x}_\varepsilon \in (\mathcal{H}_\varepsilon \cup \mathcal{H}_{-\varepsilon}) \cap \boldsymbol{R}\langle\varepsilon\rangle_b^n$ such that $\lim_\varepsilon \boldsymbol{x}_\varepsilon = \boldsymbol{x}$. For such a point $\boldsymbol{x}_\varepsilon$, let C_ε be the connected component of $(\mathcal{H}_\varepsilon \cup \mathcal{H}_{-\varepsilon}) \cap \boldsymbol{R}\langle\varepsilon\rangle^n$ containing $\boldsymbol{x}_\varepsilon$. Then, C_ε is bounded over $\boldsymbol{R}$.*

Proof. See [22, Lemma 3.6] for the first claim. The second part can be deduced following the proof of [2, Proposition 12.51]. □

Proposition 3. *Assume that $\mathscr{Z}(\mathcal{H})$ is not empty and let $\boldsymbol{x} \in \mathscr{Z}(\mathcal{H})$. There exists a semi-algebraically connected component C_ε that is bounded over $\boldsymbol{R}$ of $(\mathcal{H}_\varepsilon \cup \mathcal{H}_{-\varepsilon}) \cap \boldsymbol{R}\langle\varepsilon\rangle^n$ such that $\lim_\varepsilon C_\varepsilon = \{\boldsymbol{x}\}$.*

Consequently, for $1 \leq i \leq n$, there exists an $\boldsymbol{x}_\varepsilon \in (W(\pi_i, \mathcal{H}_\varepsilon) \cup W(\pi_i, \mathcal{H}_{-\varepsilon})) \cap C_\varepsilon$ such that $\lim_\varepsilon \boldsymbol{x}_\varepsilon = \boldsymbol{x}$. Hence we have that

$$\mathscr{Z}(\mathcal{H}) \subset \cap_{i=1}^n \lim_\varepsilon ((W(\pi_i, \mathcal{H}_\varepsilon) \cup W(\pi_i, \mathcal{H}_{-\varepsilon})) \cap \boldsymbol{R}\langle\varepsilon\rangle_b^n).$$

Proof. By Lemma 2, there exists $\boldsymbol{x}_\varepsilon \in (\mathcal{H}_\varepsilon \cup \mathcal{H}_{-\varepsilon}) \cap \boldsymbol{R}\langle\varepsilon\rangle^n$ such that $\lim_\varepsilon \boldsymbol{x}_\varepsilon = \boldsymbol{x}$. Assume that $\boldsymbol{x}_\varepsilon \in \mathcal{H}_\varepsilon$ and let C_ε be the connected component of $\mathcal{H}_\varepsilon \cap \boldsymbol{R}\langle\varepsilon\rangle^n$ containing $\boldsymbol{x}_\varepsilon$. Again, by Lemma 2, C_ε is bounded over $\boldsymbol{R}$. We prove that $\lim_\varepsilon C_\varepsilon = \{\boldsymbol{x}\}$ by contradiction. The case $\boldsymbol{x}_\varepsilon \in \mathcal{H}_{-\varepsilon}$ is done similarly.

Assume that there exists a point $\boldsymbol{y}_\varepsilon \in C_\varepsilon$ such that $\lim_\varepsilon \boldsymbol{y}_\varepsilon = \boldsymbol{y}$ and $\boldsymbol{y} \neq \boldsymbol{x}$. Since C_ε is semi-algebraically connected, there exists a semi-algebraically continuous function $\gamma : \mathrm{ext}([0, 1], \boldsymbol{R}\langle\varepsilon\rangle) \to C_\varepsilon$ such that $\gamma(0) = \boldsymbol{x}_\varepsilon$ and $\gamma(1) = \boldsymbol{y}_\varepsilon$. By [2, Proposition 12.49], $\lim_\varepsilon \mathrm{Im}(\gamma)$ is connected and contains $\boldsymbol{x}$ and $\boldsymbol{y}$. As $\lim_\varepsilon$ is a ring homomorphism, $f(\lim_\varepsilon \gamma(t)) = \lim_\varepsilon f(\gamma(t)) = 0$, so $\lim_\varepsilon \mathrm{Im}(\gamma)$ is contained in $\mathcal{H} \cap \mathbf{R}^n$. This contradicts the isolatedness of $\boldsymbol{x}$, then we conclude that $\lim_\varepsilon C_\varepsilon = \{\boldsymbol{x}\}$.

Since C_ε is a semi-algebraically connected component of the real algebraic set $\mathcal{H}_\varepsilon \cap \boldsymbol{R}\langle\varepsilon\rangle^n$, it is closed. Also, C_ε is bounded over $\boldsymbol{R}$. Hence, for any $1 \leq i \leq n$, the projection π_i reaches its extrema over C_ε [2, Proposition 7.6], which implies that $C_\varepsilon \cap W(\pi_i, \mathcal{H}_\varepsilon)$ is non-empty. Take $\boldsymbol{x}_\varepsilon \in W(\pi_i, \mathcal{H}_\varepsilon) \cap C_\varepsilon$, then $\boldsymbol{x}_\varepsilon$ is bounded over $\boldsymbol{R}$ and its limit is $\boldsymbol{x}$. Thus, $\mathscr{Z}(\mathcal{H}) \subset \lim_\varepsilon (W(\pi_i, \mathcal{H}_\varepsilon) \cap \boldsymbol{R}\langle\varepsilon\rangle_b^n)$ for any $1 \leq i \leq n$, which implies $\mathscr{Z}(\mathcal{H}) \subset \cap_{i=1}^n \lim_\varepsilon (W(\pi_i, \mathcal{H}_\varepsilon) \cap \boldsymbol{R}\langle\varepsilon\rangle_b^n)$. □

2.2 Simplification

We introduce in this subsection a method to reduce our problem to the case where $\mathcal{H} \cap \mathbf{R}^n$ is bounded for all $\boldsymbol{x} \in \mathbf{R}^n$. Such assumptions are required to prove the results in Subsection 2.3. Our technique is inspired by [2, Section 12.6]. The idea is to associate to the possibly unbounded algebraic set $\mathcal{H} \cap \mathbf{R}^n$ a bounded real algebraic set whose isolated points are strongly related to $\mathscr{Z}(\mathcal{H})$. The construction of such an algebraic set is as follows.

Let x_{n+1} be a new variable and $0 < \rho \in \boldsymbol{R}$ such that ρ is greater than the Euclidean norm $\|\cdot\|$ of every isolated point of $\mathcal{H} \cap \mathbf{R}^n$. Note

that such a ρ can be obtained from a finite set of points containing the the isolated points of $\mathcal{H} \cap \boldsymbol{R}^n$. We explain in Subsection 3.2 how to compute such a finite set.

We consider the algebraic set $\mathcal{V}$ defined by the system

$$f = 0,\ x_1^2 + \ldots + x_n^2 + x_{n+1}^2 - \rho^2 = 0.$$

Let $\pi_{\boldsymbol{x}}$ be the projection $(x_1, \ldots, x_n, x_{n+1}) \mapsto (x_1, \ldots, x_n)$.

The real counterpart of $\mathcal{V}$ is the intersection of $\mathcal{H}$ lifted to $\boldsymbol{R}^{n+1}$ with the sphere of center $\mathbf{0}$ and radius ρ. Therefore, $\mathcal{V}$ is a bounded real algebraic set in $\boldsymbol{R}^{n+1}$. Moreover, the restriction of $\pi_{\boldsymbol{x}}$ to $\mathcal{V} \cap \boldsymbol{R}^{n+1}$ is exactly $\mathcal{H} \cap B(\mathbf{0}, \rho)$. By the definition of ρ, this image contains all the real isolated points of $\mathcal{H}$. Lemma 4 below relates $\mathscr{Z}(\mathcal{H})$ to the isolated points of $\mathcal{V} \cap \boldsymbol{R}^{n+1}$.

Lemma 4. *Let $\mathcal{V}$ and $\pi_{\boldsymbol{x}}$ as above. We denote by $\mathscr{Z}(\mathcal{V}) \subset \boldsymbol{R}^{n+1}$ the set of real isolated points of $\mathcal{V}$ with non-zero x_{n+1} coordinate. Then, $\pi_{\boldsymbol{x}}(\mathscr{Z}(\mathcal{V})) = \mathscr{Z}(\mathcal{H})$.*

Proof. Note that $\pi_{\boldsymbol{x}}(\mathcal{V} \cap \boldsymbol{R}^{n+1}) = (\mathcal{H} \cap \boldsymbol{R}^n) \cap B(\mathbf{0}, \rho)$. We consider a real isolated point $\boldsymbol{x'} = (\alpha_1, \ldots, \alpha_n, \alpha_{n+1})$ of $\mathcal{V}$ with $\alpha_{n+1} \neq 0$ and $\boldsymbol{x} = \pi_{\boldsymbol{x}}(\boldsymbol{x'}) = (\alpha_1, \ldots, \alpha_n)$. Assume by contradiction that $\boldsymbol{x} \notin \mathscr{Z}(\mathcal{H})$, we will prove that $\boldsymbol{x'} \notin \mathscr{Z}(\mathcal{V})$, i.e., for any $r > 0$, there exists $\boldsymbol{y'} = (\beta_1, \ldots, \beta_n, \beta_{n+1}) \in \mathcal{V} \cap \boldsymbol{R}^{n+1}$ such that $\|\boldsymbol{y'} - \boldsymbol{x'}\| < r$. Since $\boldsymbol{x}$ is not isolated, there exists a point $\boldsymbol{y} \neq \boldsymbol{x}$ such that $\|\boldsymbol{y} - \boldsymbol{x}\| < \frac{r}{1+2\rho/|\alpha_{n+1}|}$. Let $\boldsymbol{y'} \in \pi_{\boldsymbol{x}}^{-1}(\boldsymbol{y})$ such that $\alpha_{n+1}\beta_{n+1} \geq 0$. We have that $\|\boldsymbol{x}\|^2 + \alpha_{n+1}^2 = \|\boldsymbol{y}\|^2 + \beta_{n+1}^2 = \rho^2$. Now we estimate

$$|\|\boldsymbol{y}\|^2 - \|\boldsymbol{x}\|^2| = (\|\boldsymbol{x}\| + \|\boldsymbol{y}\|) \cdot |\|\boldsymbol{y}\| - \|\boldsymbol{x}\|| \leq 2\rho \cdot \|\boldsymbol{y} - \boldsymbol{x}\|,$$

$$|\alpha_{n+1} - \beta_{n+1}| \leq \frac{|\alpha_{n+1}^2 - \beta_{n+1}^2|}{|\alpha_{n+1}|} = \frac{|\|\boldsymbol{y}\|^2 - \|\boldsymbol{x}\|^2|}{|\alpha_{n+1}|} \leq \frac{2\rho \cdot \|\boldsymbol{y} - \boldsymbol{x}\|}{|\alpha_{n+1}|}.$$

Finally,

$$\|\boldsymbol{y'} - \boldsymbol{x'}\| \leq \|\boldsymbol{y} - \boldsymbol{x}\| + |\alpha_{n+1} - \alpha_{n+1}| \leq \left(1 + \frac{2\rho}{|\alpha_{n+1}|}\right) \|\boldsymbol{y} - \boldsymbol{x}\| < r.$$

So, $\boldsymbol{x'}$ is not isolated in $\mathcal{V} \cap \boldsymbol{R}^{n+1}$. This contradiction implies that $\pi_{\boldsymbol{x}}(\mathscr{Z}(\mathcal{V})) \subset \mathscr{Z}(\mathcal{H})$.

It remains to prove that $\mathscr{Z}(\mathcal{H}) \subset \pi_{\boldsymbol{x}}(\mathscr{Z}(\mathcal{V}))$. For any real isolated point $\boldsymbol{x} \in \mathscr{Z}(\mathcal{H})$, we consider a ball $B(\boldsymbol{x}, r') \subset B(\mathbf{0}, \rho) \subset \boldsymbol{R}^n$ such that $B(\boldsymbol{x}, r') \cap \mathcal{H} = \{\boldsymbol{x}\}$. We have that $\pi_{\boldsymbol{x}}^{-1}(B(\boldsymbol{x}, r')) \cap \mathcal{V} \cap \boldsymbol{R}^{n+1}$ is equal to $\pi_{\boldsymbol{x}}^{-1}(\boldsymbol{x}) \cap \mathcal{V} \cap \boldsymbol{R}^{n+1}$, which is finite. So, all the points in $\pi_{\boldsymbol{x}}^{-1}(B(\boldsymbol{x}, r')) \cap \mathcal{V} \cap \boldsymbol{R}^{n+1}$ are isolated. Since $\mathscr{Z}(\mathcal{H}) \subset B(\mathbf{0}, \rho)$, we deduce that $\mathscr{Z}(\mathcal{H})$ is contained in $\pi_{\boldsymbol{x}}(\mathscr{Z}(\mathcal{V}))$.

Thus, we conclude that $\pi_{\boldsymbol{x}}(\mathscr{Z}(\mathcal{V})) = \mathscr{Z}(\mathcal{H})$. □

Note that the condition $x_{n+1} \neq 0$ is crucial. For a connected component C of $\mathcal{H} \cap \boldsymbol{R}^n$ that is not a singleton, its intersection with the closed ball $\overline{B(\mathbf{0}, \rho)}$ can have an isolated point on the boundary of the ball, which corresponds to an isolated point of $\mathcal{V} \cap \boldsymbol{R}^{n+1}$. This situation depends on the choice of ρ and can be easily detected by checking the vanishing of the coordinate x_{n+1}.

2.3 Identification of isolated points

By Proposition 3, the real points of $\cap_{i=1}^n \lim_\varepsilon W(\pi_i, \mathcal{H}_\varepsilon)$ are potential isolated points of $\mathcal{H} \cap \boldsymbol{R}^n$. We study now how to identify, among those candidates, which points are truely isolated.

We use the same $g = x_1^2 + \ldots + x_{n+1}^2 - \rho^2$ and $\mathcal{V} = V(f, g) \subset \boldsymbol{C}^{n+1}$ as in Subsection 2.2. Let $\mathcal{V}_\varepsilon = V(f - \varepsilon, g)$ and $\mathcal{V}_{-\varepsilon} = V(f + \varepsilon, g)$, note that they are both algebraic subsets of $C\langle\varepsilon\rangle^{n+1}$.

Lemma 5. *Let $\boldsymbol{x} \in \mathcal{V} \cap \boldsymbol{R}^{n+1}$ such that its x_{n+1} coordinate is non-zero. Then, $\boldsymbol{x}$ is not an isolated point of $\mathcal{V} \cap \boldsymbol{R}^{n+1}$ if and only if there exists a semi-algebraically connected component C_ε of $(\mathcal{V}_\varepsilon \cup \mathcal{V}_{-\varepsilon}) \cap \boldsymbol{R}\langle\varepsilon\rangle^{n+1}$, bounded over $\boldsymbol{R}$, such that $\{\boldsymbol{x}\} \subsetneq \lim_\varepsilon C_\varepsilon$.*

Proof. Let $\boldsymbol{x} = (\alpha_1, \ldots, \alpha_{n+1}) \in \mathcal{V} \cap \boldsymbol{R}^{n+1}$ such that $\alpha_{n+1} \neq 0$. As $f(\alpha_1, \ldots, \alpha_n) = 0$, by Lemma 2, there exists a point $\boldsymbol{x}_\varepsilon = (\beta_1, \ldots, \beta_{n+1}) \in \boldsymbol{R}\langle\varepsilon\rangle^{n+1}$ such that $(\beta_1, \ldots, \beta_n) \in (\mathcal{H}_\varepsilon \cup \mathcal{H}_{-\varepsilon}) \cap \boldsymbol{R}\langle\varepsilon\rangle^n$ and $\lim_\varepsilon(\beta_1, \ldots, \beta_n) = (\alpha_1, \ldots, \alpha_n)$. Since $\alpha_{n+1} \neq 0$, we can choose β_{n+1} such that $g(\boldsymbol{x}_\varepsilon) = 0$. Therefore, for any $\boldsymbol{x}$ as above, there exists $\boldsymbol{x}_\varepsilon \in (\mathcal{V}_\varepsilon \cup \mathcal{V}_{-\varepsilon}) \cap \boldsymbol{R}\langle\varepsilon\rangle^{n+1}$ such that $\lim_\varepsilon \boldsymbol{x}_\varepsilon = \boldsymbol{x}$.

Since $(\mathcal{V}_\varepsilon \cup \mathcal{V}_{-\varepsilon}) \cap \boldsymbol{R}\langle\varepsilon\rangle^{n+1}$ lies on the sphere (in $\boldsymbol{R}\langle\varepsilon\rangle^{n+1}$) defined by $g = 0$, every connected component of $(\mathcal{V}_\varepsilon \cup \mathcal{V}_{-\varepsilon}) \cap \boldsymbol{R}\langle\varepsilon\rangle^{n+1}$ is bounded over $\boldsymbol{R}$. Hence, the points of $\mathcal{V} \cap \boldsymbol{R}^{n+1}$ whose x_{n+1} coordinates are not zero are contained in $\lim_\varepsilon(\mathcal{V}_\varepsilon \cup \mathcal{V}_{-\varepsilon}) \cap \boldsymbol{R}\langle\varepsilon\rangle^{n+1}$.

Let $\boldsymbol{x}$ be a non-isolated point of $\mathcal{V} \cap \boldsymbol{R}^{n+1}$ whose x_{n+1}-coordinate is not zero. We assume by contradiction that for any semi-algebraically connected component C_ε of $(\mathcal{V}_\varepsilon \cup \mathcal{V}_{-\varepsilon}) \cap \boldsymbol{R}\langle\varepsilon\rangle^{n+1}$ (which is bounded over $\boldsymbol{R}$ by above), then it happens that either $\lim_\varepsilon C_\varepsilon = \{\boldsymbol{x}\}$ or $\boldsymbol{x} \notin \lim_\varepsilon C_\varepsilon$.

Since $(\mathcal{V}_\varepsilon \cup \mathcal{V}_{-\varepsilon}) \cap \boldsymbol{R}\langle\varepsilon\rangle^{n+1}$ has finitely many connected components, the number of connected components of the second type is also finite. Since $\mathcal{V} \cap \boldsymbol{R}^{n+1}$ is not a singleton (by the existence of $\boldsymbol{x}$), the connected components of the second type exist. So, we enumerate them as $C_1, \ldots, C_k$ and $\boldsymbol{x} \notin \lim_\varepsilon C_j$ for $1 \leq j \leq k$.

As $\boldsymbol{x}$ is not isolated in $\mathcal{V} \cap \boldsymbol{R}^{n+1}$ with non-zero x_{n+1} coordinate by assumption, there exists a sequence of points $(\boldsymbol{x}_i)_{i \geq 0}$ in $\mathcal{V} \cap \boldsymbol{R}^{n+1}$ of non-zero x_{n+1} coordinates that converges to $\boldsymbol{x}$. Since there are finitely many C_i, there exists an index j such that $\lim_\varepsilon C_j$ contains a sub-sequence of $(\boldsymbol{x}_i)_{i \geq 0}$. By Proposition 12.49 [BPR], the limit of the semi-algebraically connected component C_j (which is bounded over $\boldsymbol{R}$) is a closed and connected semi-algebraic set. It follows that $\boldsymbol{x} \in \lim_\varepsilon C_j$, which is a contradiction. Therefore, there exists a semi-algebraically connected component of $(\mathcal{V}_\varepsilon \cup \mathcal{V}_{-\varepsilon}) \cap \boldsymbol{R}\langle\varepsilon\rangle^{n+1}$, bounded over $\boldsymbol{R}$, such that $\{\boldsymbol{x}\} \subsetneq \lim_\varepsilon C_\varepsilon$.

It remains to prove the reverse implication. Assume that $\{\boldsymbol{x}\} \subsetneq \lim_\varepsilon C_\varepsilon$ for some semi-algebraically connected component C_ε of $(\mathcal{V}_\varepsilon \cup \mathcal{V}_{-\varepsilon}) \cap \boldsymbol{R}\langle\varepsilon\rangle^{n+1}$ that is bounded over $\boldsymbol{R}$. As $\lim_\varepsilon C_\varepsilon$ is connected, we finish the proof. □

Lemma 6. *Let $\boldsymbol{x} \in \mathcal{V} \cap \boldsymbol{R}^{n+1}$ whose x_{n+1} coordinate is non-zero. Assume that $\boldsymbol{x}$ is not an isolated point of $\mathcal{V} \cap \boldsymbol{R}^{n+1}$. For any semi-algebraically connected component C_ε of $(\mathcal{V}_\varepsilon \cup \mathcal{V}_{-\varepsilon}) \cap \boldsymbol{R}^{n+1}$, bounded over $\boldsymbol{R}$, such that $\{\boldsymbol{x}\} \subsetneq \lim_\varepsilon C_\varepsilon$, there exists $1 \leq i \leq n$ such that $C_\varepsilon \cap (W(\pi_i, \mathcal{V}_\varepsilon) \cup W(\pi_i, \mathcal{V}_{-\varepsilon}))$ contains a point $\boldsymbol{x}'_\varepsilon$ which satisfies $\lim_\varepsilon \boldsymbol{x}'_\varepsilon \neq \boldsymbol{x}$.*

Proof. Let C_ε be semi-algebraically connected component of $(\mathcal{V}_\varepsilon \cup \mathcal{V}_{-\varepsilon}) \cap \boldsymbol{R}\langle\varepsilon\rangle^{n+1}$, bounded over $\boldsymbol{R}$, such that $\{\boldsymbol{x}\} \subsetneq \lim_\varepsilon C_\varepsilon$. Lemma 5 ensures the existence of such a connected component C_ε.

Now let $\boldsymbol{x}_\varepsilon$ and $\boldsymbol{y}_\varepsilon$ be two points contained in C_ε such that $\lim_\varepsilon \boldsymbol{x}_\varepsilon = \boldsymbol{x}$, $\lim_\varepsilon \boldsymbol{y}_\varepsilon = \boldsymbol{y}$ and $\boldsymbol{x} \neq \boldsymbol{y}$. Let $\boldsymbol{x} = (\alpha_1, \ldots, \alpha_{n+1})$ and $\boldsymbol{y} = (\beta_1, \ldots, \beta_{n+1})$. Since $\boldsymbol{x} \neq \boldsymbol{y}$, there exists $1 \leq i \leq n+1$ such that $\alpha_i \neq \beta_i$. Note that if $(\alpha_1, \ldots, \alpha_n) = (\beta_1, \ldots, \beta_n)$ for any $\boldsymbol{y} \in \lim_\varepsilon C_\varepsilon$, then $\lim_\varepsilon C_\varepsilon$ contains at most two points (by the constraint $g = 0$). However, since $\lim_\varepsilon C_\varepsilon$ is connected and contains at least two points, it must be an infinite set. So, we can choose $\boldsymbol{y}$ such that have that $1 \leq i \leq n$.

As C_ε is closed in $\boldsymbol{R}\langle\varepsilon\rangle^{n+1}$ (as a connected component of an algebraic set) and bounded over $\boldsymbol{R}$ by definition, its projection on the x_i coordinate is a closed interval $[a,b] \subset \boldsymbol{R}\langle\varepsilon\rangle$ (see [2, Theorem 3.23]), which is bounded over $\boldsymbol{R}$ (because C_ε is). Also, since $[a,b]$ is closed, there exist $\boldsymbol{x}'_a$ and $\boldsymbol{x}'_b$ in $\boldsymbol{R}\langle\varepsilon\rangle^{n+1}$ such that $\boldsymbol{x}'_a \in \pi_i^{-1}(a) \cap C_\varepsilon \cap (W(\pi_i, \mathcal{V}_\varepsilon) \cup W(\pi_i, \mathcal{V}_{-\varepsilon}))$ and $\boldsymbol{x}'_b \in \pi_i^{-1}(b) \cap C_\varepsilon \cap (W(\pi_i, \mathcal{V}_\varepsilon) \cup W(\pi_i, \mathcal{V}_{-\varepsilon}))$. Since $\alpha_i \neq \beta_i$ both lying in $\boldsymbol{R}$, $\{\alpha_i, \beta_i\} \subset [\lim_\varepsilon a, \lim_\varepsilon b]$ implies that $\lim_\varepsilon a \neq \lim_\varepsilon b$. It follows that $\lim_\varepsilon \boldsymbol{x}'_a \neq \lim_\varepsilon \boldsymbol{x}'_b$. Thus, at least one point among $\lim_\varepsilon \boldsymbol{x}'_a$ and $\lim_\varepsilon \boldsymbol{x}'_b$ does not coincide with $\boldsymbol{x}$. Hence, there exists a point $\boldsymbol{x}'_\varepsilon$ in $C_\varepsilon \cap (W(\pi_i, \mathcal{V}_\varepsilon) \cup W(\pi_i, \mathcal{V}_{-\varepsilon}))$ such that $\lim_\varepsilon \boldsymbol{x}'_\varepsilon \neq \boldsymbol{x}$. □

We can easily deduce from Lemma 5 and Lemma 6 the following proposition, which is the main ingredient of our algorithm.

Proposition 7. *Let $\boldsymbol{x} \in \cap_{i=1}^n \lim_\varepsilon W(\pi_i, \mathcal{V}_\varepsilon) \cup W(\pi_i, \mathcal{V}_{-\varepsilon})$ whose x_{n+1} coordinate is non-zero. Then, $\boldsymbol{x}$ is not an isolated point of $\mathcal{V} \cap \boldsymbol{R}^{n+1}$ if and only if there exist $1 \leq i \leq n$ and a connected component C_ε of $\mathcal{V}_\varepsilon \cap \boldsymbol{R}\langle\varepsilon\rangle^{n+1}$, which is bounded over $\boldsymbol{R}$, such that $C_\varepsilon \cap W(\pi_i, \mathcal{H}_\varepsilon)$ contains $\boldsymbol{x}_\varepsilon, \boldsymbol{x}'_\varepsilon$ satisfying $\boldsymbol{x} = \lim_\varepsilon \boldsymbol{x}_\varepsilon \neq \lim_\varepsilon \boldsymbol{x}'_\varepsilon$.*

3 ALGORITHM

3.1 General description

The algorithm takes as input a polynomial $f \in \boldsymbol{R}[x_1, \ldots, x_n]$.

The first step consists in computing a parametrization $\mathfrak{P}$ encoding a finite set of points which contains $\mathscr{Z}(\mathcal{H})$. Let $\mathcal{H}_\varepsilon$ and $\mathcal{H}_{-\varepsilon}$ be the algebraic subsets of $C\langle\varepsilon\rangle^n$ respectively defined by $f = \varepsilon$ and $f = -\varepsilon$. By Proposition 3, the set $\cap_{i=1}^n \lim_\varepsilon W(\pi_i, \mathcal{H}_\varepsilon) \cup W(\pi_i, \mathcal{H}_{-\varepsilon})$ contains the real isolated points of $\mathcal{H}$. To ensure that this set is finite, we use *generically chosen* linear change of coordinates.

Given a matrix $\boldsymbol{A} \in GL_n(\boldsymbol{Q})$, a polynomial $p \in \boldsymbol{Q}[x_1, \ldots, x_n]$ and an algebraic set $\mathcal{S} \subset \boldsymbol{C}^n$, we denote by $p^{\boldsymbol{A}}$ the polynomial $p(\boldsymbol{A} \cdot \boldsymbol{x})$ obtained by applying the change of variables $\boldsymbol{A}$ to p and $\mathcal{S}^{\boldsymbol{A}} = \{\boldsymbol{A}^{-1} \cdot \boldsymbol{x} \mid \boldsymbol{x} \in \mathcal{S}\}$. Then, we have that $V(p)^{\boldsymbol{A}} = V(p^{\boldsymbol{A}})$.

In [25], it is proved that, with $\boldsymbol{A}$ outside a prescribed proper Zariski closed subset of $GL_n(\boldsymbol{Q})$, $W(\pi_i, \mathcal{H}_\varepsilon^{\boldsymbol{A}}) \cup W(\pi_i, \mathcal{H}_{-\varepsilon}^{\boldsymbol{A}})$ is finite for $1 \leq i \leq n$. Additionally, since $\boldsymbol{A}$ is assumed to be generically chosen, [24] shows that the ideal $\left\langle \ell \cdot \frac{\partial f^{\boldsymbol{A}}}{\partial x_i} - 1, \frac{\partial f^{\boldsymbol{A}}}{\partial x_j} \text{ for all } j \neq i \right\rangle$ defines either an empty set or a one-equidimensional algebraic set, where ℓ is a new variable. Those extra assumptions are required in our subroutine Candidates (see the next subsection). Note that, for any matrix $\boldsymbol{A}$, the real isolated points of $\mathcal{H}^{\boldsymbol{A}}$ is the image of $\mathscr{Z}(\mathcal{H})$ by the linear mapping associated to $\boldsymbol{A}^{-1}$. Thus, in practice, we will choose randomly a $\boldsymbol{A} \in GL_{n+1}(\boldsymbol{Q})$, compute the real isolated points of $\mathcal{H}^{\boldsymbol{A}}$, and then go back to $\mathscr{Z}(\mathcal{H})$ by applying the change of coordinates induced by $\boldsymbol{A}^{-1}$. This random choice of $\boldsymbol{A}$ makes our algorithm probabilistic.

The next step consists of identifying those of the candidates which are isolated in $\mathcal{H}^{\boldsymbol{A}} \cap \boldsymbol{R}^n$; this step relies on Proposition 7. To reduce our problem to the context where Proposition 7 can be applied, we use Lemma 4. One needs to compute $\rho \in \boldsymbol{R}$, such that ρ is larger than the maximum norm of the real isolated points we want to compute. This value of ρ can be easily obtained by isolating the real roots of the zero-dimensional parametrization encoding the candidates. Further, we call GetNormBound a subroutine which takes as input $\mathfrak{P}$ and returns ρ as we just sketched. We let $g = x_1^2 + \ldots + x_n^2 + x_{n+1}^2 - \rho^2$. By Lemma 4, $\mathscr{Z}(\mathcal{H})$ is the projection of the set of real isolated points of the algebraic set $\mathcal{V}$ defined by $f = g = 0$ at which $x_{n+1} \neq 0$. Let $\mathcal{X}$ be the set of points of $\mathcal{V}$ projecting to the candidates encoded by $\mathfrak{P}$.

Proposition 7 would lead us to compute $W(\pi_i, \mathcal{V}_\varepsilon^{\boldsymbol{A}}) \cup W(\pi_i, \mathcal{V}_{-\varepsilon}^{\boldsymbol{A}})$ as well as a roadmap of $\mathcal{V}_\varepsilon^{\boldsymbol{A}} \cup \mathcal{V}_{-\varepsilon}^{\boldsymbol{A}}$. As explained in the introduction, this induces computations over the ground field $\boldsymbol{R}\langle\varepsilon\rangle$ which we want to avoid. We bypass this computational difficulty as follows. We compute a roadmap $\mathcal{R}_e$ for $\mathcal{V}_e^{\boldsymbol{A}} \cup \mathcal{V}_{-e}^{\boldsymbol{A}} \cap \boldsymbol{R}^{n+1}$ (defined by $\{f^{\boldsymbol{A}} = e, g = 0\}$ and $\{f^{\boldsymbol{A}} = -e, g = 0\}$ respectively) for e small enough (see Subsection 3.3) and define a semi-algebraic curve $\mathcal{K}$ containing $\mathcal{X}$ such that $\boldsymbol{x} \in \mathcal{X}$ is isolated in $\mathcal{V}^{\boldsymbol{A}} \cap \boldsymbol{R}^{n+1}$ if and only if it is not connected to any other $\boldsymbol{x}' \in \mathcal{X}$ by $\mathcal{K}$. We call IsIsolated the subroutine that takes as input $\mathfrak{P}$, $f^{\boldsymbol{A}}$ and g and returns $\mathfrak{P}$ with isolating boxes B of the real points of defined by $\mathfrak{P}$ which are isolated in $\mathcal{V}^{\boldsymbol{A}} \cap \boldsymbol{R}^{n+1}$.

Once the real isolated points of $\mathcal{V}^{\boldsymbol{A}}$ is computed, we remove the boxes corresponding to points at which $x_{n+1} = 0$ and project the remaining points on the $(x_1, \ldots, x_n)$-space to obtain the isolated points of $\mathcal{H}^{\boldsymbol{A}}$. This whole step uses a subroutine which we call Remove (see [28, Appendix J]). Finally, we reverse the change of variable by applying $\boldsymbol{A}^{-1}$ to get $\mathscr{Z}(\mathcal{H})$.

We summarize our discussion in Algorithm 1 below.

Algorithm 1: IsolatedPoints

Input: A polynomial $f \in \boldsymbol{Q}[x_1, \ldots, x_n]$
Output: A zero-dimensional parametrization $\mathfrak{P}$ such that $\mathscr{Z}(\mathcal{H}) \subset Z(\mathfrak{P})$ and a set of boxes isolating $\mathscr{Z}(\mathcal{H})$

$\boldsymbol{A}$ chosen randomly in $GL_{n+1}(\boldsymbol{Q})$
$\mathfrak{P} \leftarrow$ Candidates($f^{\boldsymbol{A}}$)
$\rho \leftarrow$ GetNormBound($\mathfrak{P}$)
$g \leftarrow x_1^2 + \ldots + x_n^2 + x_{n+1}^2 - \rho^2$
$\mathfrak{P}, B \leftarrow$ IsIsolated($\mathfrak{P}, f^{\boldsymbol{A}}, g$)
$\mathfrak{P}, B \leftarrow$ Removes($\mathfrak{P}, B, x_{n+1}$)
$\mathfrak{P}, B \leftarrow \mathfrak{P}^{\boldsymbol{A}^{-1}}, B^{\boldsymbol{A}^{-1}}$
return $(\mathfrak{P}, B)$

3.2 Computation of candidates

Further, we let $\mathcal{H}_\varepsilon^{\boldsymbol{A}}$ (resp. $\mathcal{H}_{-\varepsilon}^{\boldsymbol{A}}$) be the algebraic set associated to $f^{\boldsymbol{A}} = \varepsilon$ (resp. $f^{\boldsymbol{A}} = -\varepsilon$). To avoid to overload notation, we omit the change of variables $\boldsymbol{A}$ as upper script. Let ℓ be a new variable. For $1 \leq i \leq n$, I_i denotes the ideal of $\boldsymbol{Q}[\ell, x_1, \ldots, x_n]$ generated by the set of polynomials $\left\{\ell \cdot \frac{\partial f}{\partial x_i} - 1, \frac{\partial f}{\partial x_j} \quad \text{for all } j \neq i\right\}$.

Following the discussion in Subsection 3.1, the algebraic set associated to I_i is either empty or one-equidimensional and $W(\pi_i, \mathcal{H}_\varepsilon) \cup W(\pi_i, \mathcal{H}_{-\varepsilon})$ is finite. Hence, [24, Theorem 1] shows that the algebraic set associated to the ideal $\langle f \rangle + (I_i \cap \boldsymbol{Q}[x_1, \ldots, x_n])$ is zero-dimensional and contains $\lim_\varepsilon W(\pi_i, \mathcal{H}_\varepsilon) \cup W(\pi_i, \mathcal{H}_{-\varepsilon})$.

In our problem, the intersection of $\lim_\varepsilon W(\pi_i, \mathcal{H}_\varepsilon) \cup W(\pi_i, \mathcal{H}_{-\varepsilon})$ is needed rather than each limit itself. Hence, we use the inclusion

$$\cap_{i=1}^n \lim_\varepsilon W(\pi_i, \mathcal{H}_\varepsilon) \cup W(\pi_i, \mathcal{H}_{-\varepsilon}) \subset V\left(\langle f \rangle + \textstyle\sum_{i=1}^n I_i \cap \boldsymbol{Q}[x_1, \ldots, x_n]\right).$$

We can compute the algebraic set on the right-hand side as follows:

(1) For each $1 \leq i \leq n$, compute a set G_i of generators of the ideal $I_i \cap \boldsymbol{Q}[x_1, \ldots, x_n]$.

(2) Compute a zero-dimensional parametrization $\mathfrak{P}$ of the set of polynomials $\{f\} \cup G_1 \cup \ldots \cup G_n$.

Such computations mimic those in [24]. The complexity of this algorithm of course depends on the algebraic elimination procedure we use. For the complexity analysis in Section 4, we employ the geometric resolution [14]. It basically consists in computing a one-dimensional parametrization of the curve defined by I_i and next computes a zero-dimensional parametrization of the finite set obtained by intersecting this curve with the hypersurface defined by $f = 0$. We call ParametricCurve a subroutine that, taking the polynomial f and $1 \leq i \leq n$, computes a one-dimensional parametrization $\mathfrak{G}_i$ of the curve defined above. Also, let IntersectCurve be a subroutine that, given a one-dimensional rational parametrization $\mathfrak{G}_i$ and f, outputs a zero-dimensional parametrization $\mathfrak{P}_i$ of their intersection. Finally, we use a subroutine Intersection that, from the parametrizations $\mathfrak{P}_i$'s, computes a zero-dimensional parametrization of $\cap_{i=1}^n Z(\mathfrak{P}_i)$.

Algorithm 2: Algorithm Candidates

Input: The polynomial $f \in \boldsymbol{Q}[x_1, \ldots, x_n]$
Output: A zero-dimensional parametrization $\mathfrak{P}$

```
for 1 ≤ i ≤ n do
    G_i ← ParametricCurve(g, i)
    P_i ← IntersectCurve(G_i, g)
end
P ← Intersection(P_1, ..., P_n)
return P
```

3.3 Description of IsIsolated

This subsection is devoted to the subroutine IsIsolated that identifies isolated points of $\mathcal{H}^{\boldsymbol{A}} \cap \boldsymbol{R}^n$ among the candidates $Z(\mathfrak{P}) \cap \boldsymbol{R}^n$ computed in the previous subsection. We keep using f to address $f^{\boldsymbol{A}}$. Let $\mathcal{P}$ be the set $\{\boldsymbol{x} = (x_1, \ldots, x_{n+1}) \in \boldsymbol{R}^{n+1} \mid (x_1, \ldots, x_n) \in Z(\mathfrak{P}), g(\boldsymbol{x}) = 0, x_{n+1} \neq 0\}$.

For $e \in \boldsymbol{R}$, $\mathcal{V}_e \in \boldsymbol{C}^{n+1}$ denotes the algebraic set defined by $f = e$ and $g = 0$. We follow the idea mentioned in the end of Subsection 3.1, that is to replace the infinitesimal ε by a sufficiently small $e \in \boldsymbol{R}$ then adapt the results of Subsection 2.3 to $\mathcal{V}_e$.

By definition, $\mathcal{V}_e \cap \boldsymbol{R}^{n+1}$ is bounded for any $e \in \boldsymbol{R}$. Let t be a new variable, $\pi_{\boldsymbol{x}} : (\boldsymbol{x}, t) \mapsto \boldsymbol{x}$ and $\pi_t : (\boldsymbol{x}, t) \mapsto t$. For a semi-algebraic set $\mathcal{S} \subset \boldsymbol{R}^{n+1} \times \boldsymbol{R}$ in the coordinate $(\boldsymbol{x}, t)$ and a subset $\mathcal{I}$ of $\boldsymbol{R}$, the notation $\mathcal{S}_{\mathcal{I}}$ stands for the fiber $\pi_t^{-1}(\mathcal{I}) \cap \mathcal{S}$. Let $\mathcal{V}_t = \{(\boldsymbol{x}, t) \in \boldsymbol{R}^{n+1} \times \boldsymbol{R} \mid f(\boldsymbol{x}) = t, g(\boldsymbol{x}) = 0\}$. Note that $\mathcal{V}_t$ is smooth. Recall that the set of critical values of the restriction of π_t to $\mathcal{V}_t$ is finite by the algebraic Sard's theorem (see e.g., [28, Proposition B.2]).

Since for $e \in \boldsymbol{R}$, the set $\mathcal{V}_e \cap \boldsymbol{R}^{n+1}$ is compact, the restriction of π_t to $\mathcal{V}_t$ is proper. Then, by Thom's isotopy lemma [9], π_t realizes a locally trivial fibration over any open connected subset of $\boldsymbol{R}$ which does not intersect the set of critical values of the restriction of π_t to $\mathcal{V}_t$. Let $\eta \in \boldsymbol{R}$ such that the open set $]-\eta, 0[\cup]0, \eta[$ does not contain any critical value of the restriction of π_t to the algebraic set $\mathcal{V}_t$. Hence, $\mathcal{V}_e$ is nonsingular for $e \in]-\eta, 0[\cup]0, \eta[$, $(\mathcal{V}_e \cap \boldsymbol{R}^{n+1}) \times (]-\eta, 0[\cup]0, \eta[)$ is diffeomorphic to $\mathcal{V}_{t,]-\eta,0[\cup]0,\eta[}$.

We need to mention that $W(\pi_i, \mathcal{H}_e)$ corresponds to the critical points of π_i restricted to $\mathcal{V}_e$ with non-zero x_{n+1} coordinate. Further, we use $W(\pi_i, \mathcal{V}_e)$ to address those latter critical points.

Now, for $1 \leq i \leq n$, we define $\mathcal{W}_i$ as the closure of

$$\left\{(\boldsymbol{x}, t) \in \boldsymbol{R}^{n+2} \mid \frac{\partial f}{\partial x_i}(\boldsymbol{x}) \neq 0, \frac{\partial f}{\partial x_j}(\boldsymbol{x}) = 0 \text{ for } j \neq i, x_{n+1} \neq 0\right\} \cap \mathcal{V}_t.$$

Since $\boldsymbol{A}$ is assumed to be generically chosen, $\mathcal{W}_i$ is either empty or one-equidimensional (because $\langle \ell \cdot \frac{\partial f}{\partial x_i} - 1, \frac{\partial f}{\partial x_j} \forall j \neq i \rangle$ either defines an empty set or a one-equidimensional algebraic set by [24]). This implies that the set of singular points of $\mathcal{W}_i$ is finite.

By [18], the set of non-properness of the restriction of π_t to $\mathcal{W}_i$ is finite (this is the set of points y such that for any closed interval U containing y, $\pi_t^{-1}(U) \cap \mathcal{W}_i$ is not bounded). Using again [18], the restriction of π_t to $\mathcal{W}_i$ realizes a locally trivial fibration over any connected open subset which does not meet the union of the images by π_t of the singular points of $\mathcal{W}_i$, the set of non-properness, and the set of critical values of the restriction of π_t to $\mathcal{W}_i$. We let η_i' be the minimum of the absolute values of the points in this union.

We choose now $0 < e_0 < \min(\eta, \eta_1', \ldots, \eta_n')$. We call SpecializationValue a subroutine that takes as input f and g and returns such a rational number e_0. Note that SpecializationValue is easily obtained from elimination algorithms solving polynomial systems (from which we can compute critical values) and from [26] to compute the set of non-properness of some map.

With e_0 as above, we denote $\mathcal{I} =]-e_0, 0[\cup]0, e_0[$. Let $\mathcal{W}_{i,\mathcal{I}}$ is semi-algebraically diffeomorphic to $\mathcal{W}_{i,e} \times \mathcal{I}$ for every $e \in \mathcal{I}$. As $\mathcal{V}_e$ is nonsingular, the critical locus $W(\pi_i, \mathcal{V}_e)$ is guaranteed to be finite by the genericity of the change of variables $\boldsymbol{A}$ (hence $\mathcal{W}_{i,e}$ is) and that $W(\pi_i, \mathcal{V}_e) \cap \boldsymbol{R}^{n+1}$ coincides with $\pi_{\boldsymbol{x}}(\mathcal{W}_{i,e})$. Thus, the above diffeomorphism implies that, for any connected component C of $\mathcal{W}_{i,\mathcal{I}}$, C is diffeomorphic to an open interval in $\boldsymbol{R}$. Moreover, if C is bounded, then $\overline{C} \setminus C$ contains exactly two points which satisfy respectively $f = 0$ and $f^2 = e_0^2$. We now consider

$$\mathcal{L}_i = \left\{\boldsymbol{x} \in \boldsymbol{R}^{n+1} \mid 0 < f < e_0, g = 0, \frac{\partial f}{\partial x_j} = 0 \text{ for} j \neq i, x_{n+1} \neq 0\right\}.$$

It is the intersection of the Zariski closure $\mathcal{K}_i$ of the solution set to $\left\{\frac{\partial f}{\partial x_i} \neq 0, \frac{\partial f}{\partial x_j} = 0 \text{ for } j \neq i, x_{n+1} \neq 0\right\}$ with the semi-algebraic set defined by $0 < f < e_0$. Note that $\mathcal{K}_i$ is either empty or one-equidimensional. As $\mathcal{V}_e$ is nonsingular for $e \in \mathcal{I}$, $\mathcal{L}_i$ and $\mathcal{L}_j$ are disjoint for $i \neq j$. Since the restriction of $\pi_{\boldsymbol{x}}$ to $\mathcal{V}_t$ is an isomorphism between the algebraic sets $\mathcal{V}_t$ and $\boldsymbol{R}^{n+1}$ with the inverse map $\boldsymbol{x} \mapsto (\boldsymbol{x}, f(\boldsymbol{x}))$, the properties of $\mathcal{W}_{\mathcal{I}}$ mentioned above are transferred to its image $\mathcal{L}_i$ by the projection $\pi_{\boldsymbol{x}}$.

Further, we consider a subroutine ParametricCurve which takes as input f and $i \in [1, n]$ and returns a rational parametrization $\mathfrak{K}_i$ of $\mathcal{K}_i$. Also, let Union be a subroutine that takes a family of rational parametrizations $\mathfrak{K}_1, \ldots, \mathfrak{K}_n$ to compute a rational parametrization encoding the union of the algebraic curves defined by the $\mathfrak{K}_i$'s. We denote by $\mathfrak{K}$ the output of Union ; it encodes $\mathcal{K} = \cup_{i=1}^n \mathcal{K}_i$. We refer to [28, Appendix J.2] for these two subroutines.

Lemma 8 below establishes a *well-defined* notion of limit for a point $\boldsymbol{x}_e \in W(\pi_i, \mathcal{V}_e) \cup W(\pi_i, \mathcal{V}_{-e})$ when e tends to 0.

Lemma 8. *Let e_0 and $\mathcal{L}_i$ be as above. For $e \in]0, e_0[$ and $\boldsymbol{x}_e \in (W(\pi_i, \mathcal{V}_e) \cup W(\pi_i, \mathcal{V}_{-e})) \cap \boldsymbol{R}^{n+1}$, there exists a (unique) connected component C of $\mathcal{L}_i$ containing $\boldsymbol{x}_e$. If C is bounded, let $\boldsymbol{x}$ be the only point in $\overline{C}$ satisfying $f(\boldsymbol{x}) = 0$, then $\boldsymbol{x} \in \lim_\varepsilon(W(\pi_i, \mathcal{V}_\varepsilon) \cup W(\pi_i, \mathcal{V}_{-\varepsilon})) \cap \boldsymbol{R}\langle\varepsilon\rangle^{n+1}$. Thus, we set $\lim_0 \boldsymbol{x}_e = \boldsymbol{x}$.*

Moreover, the extension $\mathrm{ext}(C, \boldsymbol{R}\langle\varepsilon\rangle)$ *contains exactly one point* $\boldsymbol{x}_\varepsilon$ *such that* $f(\boldsymbol{x}_\varepsilon)^2 = \varepsilon^2$ *and* $\lim_\varepsilon \boldsymbol{x}_\varepsilon = \boldsymbol{x}$.

Proof. Since $\boldsymbol{x}_e \in (W(\pi_i, \mathcal{V}_e) \cup W(\pi_i, \mathcal{V}_{-e})) \cap \boldsymbol{R}^{n+1}$ and $0 < e < e_0$, we have $\boldsymbol{x}_e \in \mathcal{L}_i$, the existence of C follows naturally. Let $\boldsymbol{x}$ be the unique point of $\overline{C}$ satisfying $f = 0$. Then, the notion $\lim_0$ is well-defined. From the proof of [2, Theorem 12.43], we have that

$$\lim_\varepsilon (W(\pi_i, \mathcal{V}_\varepsilon) \cup W(\pi_i, \mathcal{V}_{-\varepsilon})) \cap \boldsymbol{R}\langle\varepsilon\rangle^{n+1} = \pi_{\boldsymbol{x}}\left(\overline{\mathcal{W}_{(0,+\infty)}} \cap V(t)\right).$$

As $\pi_{\boldsymbol{x}}\left(\overline{\mathcal{W}_{(0,+\infty)}} \cap V(t)\right)$ is the set of points corresponding to $f = 0$ of $\mathcal{L}_i$, we deduce that $\boldsymbol{x} \in \lim_\varepsilon (W(\pi_i, \mathcal{V}_\varepsilon) \cup W(\pi_i, \mathcal{V}_{-\varepsilon})) \cap \boldsymbol{R}\langle\varepsilon\rangle^{n+1}$.

Since the extension $\mathrm{ext}(C, \boldsymbol{R}\langle\varepsilon\rangle)$ is a connected component of $\mathrm{ext}(\mathcal{L}_i, \boldsymbol{R}\langle\varepsilon\rangle)$ and homeomorphic to an open interval in $\boldsymbol{R}\langle\varepsilon\rangle$, there exists $\boldsymbol{x}_\varepsilon \in \mathrm{ext}(C, \boldsymbol{R}\langle\varepsilon\rangle)$ such that $f(\boldsymbol{x}_\varepsilon)^2 = \varepsilon^2$. Moreover, since $0 = \lim_\varepsilon f(\boldsymbol{x}_\varepsilon)^2 = f(\lim_\varepsilon \boldsymbol{x}_\varepsilon)^2$ and $\boldsymbol{x}$ is the only point in $\overline{C}$ satisfying $f = 0$, we conclude that $\lim_\varepsilon \boldsymbol{x}_\varepsilon = \boldsymbol{x}$. □

Now, let $\mathcal{R}_e$ be a roadmap associated to the algebraic set $\mathcal{V}_e \cup \mathcal{V}_{-e}$, i.e. $\mathcal{R}_e$ is contained in $(\mathcal{V}_e \cup \mathcal{V}_{-e}) \cap \boldsymbol{R}^{n+1}$, of at most dimension one and has non-empty intersection with every connected component of $(\mathcal{V}_e \cup \mathcal{V}_{-e}) \cap \boldsymbol{R}^{n+1}$. We also require that $\mathcal{R}_e$ contains $\cup_{i=1}^n (W(\pi_i, \mathcal{V}_e) \cup W(\pi_i, \mathcal{V}_{-e})) \cap \boldsymbol{R}^{n+1}$. The proposition below is the key to describe IsIsolated.

Proposition 9. *Given* $e \in]0, e_0[$ *and* $\mathcal{I} =]-e_0, 0[\cup]0, e_0[$ *as above. Let* $\mathcal{L} = \cup_{i=1}^n \mathcal{L}_i$ *and* $\boldsymbol{x} \in \mathcal{P}$. *Then* $\boldsymbol{x}$ *is not isolated in* $\mathcal{V} \cap \boldsymbol{R}^{n+1}$ *if and only if there exists* $\boldsymbol{x}' \in \mathcal{P}$ *such that* $\boldsymbol{x}$ *and* $\boldsymbol{x}'$ *are connected in* $\mathcal{P} \cup \mathcal{L} \cup \mathcal{R}_e$.

Proof. Assume first that $\boldsymbol{x}$ is not isolated. By Proposition 7, there exists $1 \le i \le n$ and a connected component C_ε of $(\mathcal{V}_\varepsilon \cup \mathcal{V}_{-\varepsilon}) \cap \boldsymbol{R}\langle\varepsilon\rangle^{n+1}$, which is bounded over $\boldsymbol{R}$, such that $C_\varepsilon \cap (W(\pi_i, \mathcal{V}_\varepsilon) \cup W(\pi_i, \mathcal{V}_{-\varepsilon}))$ contains $\boldsymbol{x}_\varepsilon$ and $\boldsymbol{x}'_\varepsilon$ satisfying $\boldsymbol{x} = \lim_\varepsilon \boldsymbol{x}_\varepsilon \neq \lim_\varepsilon \boldsymbol{x}'_\varepsilon$. By the choice of e_0, there exist a diffeomorphism $\theta : \mathcal{V}_{t,\mathcal{I}} \to \mathcal{V}_e \times \mathcal{I}$ such that $\theta(\mathcal{W}_{i,\mathcal{I}}) = \theta(\mathcal{W}_{i,e}) \times \mathcal{I}$. Using [2, Exercise 3.2], $\mathrm{ext}(\theta, \boldsymbol{R}\langle\varepsilon\rangle)$ is a diffeomorphism between:

$$\mathrm{ext}(\mathcal{V}_{t,\mathcal{I}}, \boldsymbol{R}\langle\varepsilon\rangle) \cong \mathrm{ext}(\mathcal{V}_e, \boldsymbol{R}\langle\varepsilon\rangle) \times \mathrm{ext}(\mathcal{I}, \boldsymbol{R}\langle\varepsilon\rangle),$$
$$\mathrm{ext}(\mathcal{W}_{i,\mathcal{I}}, \boldsymbol{R}\langle\varepsilon\rangle) \cong \mathrm{ext}(\mathcal{W}_{i,e}, \boldsymbol{R}\langle\varepsilon\rangle) \times \mathrm{ext}(\mathcal{I}, \boldsymbol{R}\langle\varepsilon\rangle).$$

As $\pi_{\boldsymbol{x}}$ is an isomorphism from $\mathcal{V}_t$ to $\boldsymbol{R}^{n+1}$, there exists a (unique) bounded connected component C_e of $\mathcal{V}_e \cap \boldsymbol{R}^{n+1}$ s.t. C_ε is diffeomorphic to $\mathrm{ext}(C_e, \boldsymbol{R}\langle\varepsilon\rangle)$. Moreover, let L and L' be the connected components of $\mathrm{ext}(\mathcal{L}_i, \boldsymbol{R}\langle\varepsilon\rangle)$ containing $\boldsymbol{x}_\varepsilon$ and $\boldsymbol{x}'_\varepsilon$ respectively and $\boldsymbol{x}_e$ and $\boldsymbol{x}'_e$ ($\in \mathrm{ext}(C_e, \boldsymbol{R}\langle\varepsilon\rangle)$) be the intersections of $\mathrm{ext}(C_e, \boldsymbol{R}\langle\varepsilon\rangle)$ with L and L' respectively. Then, $\lim_\varepsilon \boldsymbol{x}_\varepsilon$ ($\lim_\varepsilon L'$) connects $\lim_\varepsilon \boldsymbol{x}_e$ ($\lim_\varepsilon \boldsymbol{x}'_e$) to $\boldsymbol{x}$ ($\boldsymbol{x}'$). As $\lim_\varepsilon \boldsymbol{x}_e$ and $\lim_\varepsilon \boldsymbol{x}'_e$ are connected in C_e, we conclude that $\boldsymbol{x}$ and $\boldsymbol{x}'$ are also connected in $\mathcal{P} \cup \mathcal{L} \cup \mathcal{R}_e$. The reverse implication is immediate using the above techniques □

From Lemma 8 and Proposition 9, any e lying in the interval $]0, e_0[$ defined above can be used to replace the infinitesimal ε. So, we simply take $e = e_0/2$. For $1 \le i \le n$, we use a subroutine ZeroDimSolve which takes as input $\left\{f - e_0/2, g, \frac{\partial f}{\partial x_j} \text{ for all } j \neq i\right\}$ to compute a zero-dimensional parametrization $\mathfrak{Q}_i$ such that $W(\pi_i, \mathcal{V}_e) = \{\boldsymbol{x} \in Z(\mathfrak{Q}_i) | x_{n+1} \neq 0\}$.

To use Proposition 9, we need to compute $\mathcal{R}_{e_0/2}$, which we refer to the algorithm Roadmap in [28]. This algorithm allows us to compute roadmaps for smooth and bounded real algebraic sets, which is indeed the case of $(\mathcal{V}_{e_0/2} \cup \mathcal{V}_{-e_0/2}) \cap \boldsymbol{R}^{n+1}$. First, we call (another) Union that, on the zero-dimensional parametrizations $\mathfrak{Q}_i$, it computes a zero-dimensional parametrization $\mathfrak{Q}$ encoding $\cup_{i=1}^n Z(\mathfrak{Q}_i)$. Given the polynomials f, g, the value $e_0/2$ and the parametrization $\mathfrak{Q}$, a combination of Union and Roadmap returns a one-dimensional parametrization $\mathfrak{R}$ representing $\mathcal{R}_{e_0/2}$.

Deciding connectivity over $\mathcal{P} \cup \mathcal{L} \cup \mathcal{R}_e$ is done as follows. We use Union to compute a rational parametrization $\mathfrak{S}$ encoding $\mathcal{K} \cup \mathcal{R}_e$. Then, with input $\mathfrak{S}$, $\mathfrak{P}$, $x_{n+1} \neq 0$ and the inequalities $0 < f < e_0$, we use Newton Puiseux expansions and cylindrical algebraic decomposition (see [10, 30]) following [27], taking advantage of the fact that polynomials involved in rational parametrizations of algebraic curves are bivariate. We denote by ConnectivityQuery the subroutine that takes those inputs and returns $\mathfrak{P}$ and isolating boxes of the points defined by $\mathfrak{P}$ which are not connected to other points of $\mathfrak{P}$.

Algorithm 3: IsIsolated

Input: The polynomials $f^{\boldsymbol{A}} \in \boldsymbol{Q}[x_1, \ldots, x_n]$ and $g \in \boldsymbol{Q}[x_1, \ldots, x_{n+1}]$ and the zero-dimensional parametrization $\mathfrak{P}$.
Output: $\mathfrak{P}$ with isolating boxes of the isolated points of $\mathcal{V}^{\boldsymbol{A}} \cap \boldsymbol{R}^{n+1}$

$e_0 \leftarrow$ SpecializationValue$(f^{\boldsymbol{A}}, g)$
for $1 \le i \le n$ **do**
  $\mathfrak{Q}_i \leftarrow$ ZeroDimSolve$\left(\left\{f^{\boldsymbol{A}} - e_0/2, g, \frac{\partial f^{\boldsymbol{A}}}{\partial x_j} \text{ for all } j \neq i\right\}\right)$
  $\mathfrak{R}_i \leftarrow$ ParametricCurve$(f^{\boldsymbol{A}}, i)$
end
$\mathfrak{R} \leftarrow$ Union$(\mathfrak{R}_1, \ldots, \mathfrak{R}_n)$
$\mathfrak{Q} \leftarrow$ Union$(\mathfrak{Q}_1, \ldots, \mathfrak{Q}_n)$
$\mathfrak{R} \leftarrow$ Union(RoadMap$(f^{\boldsymbol{A}} - e_0/2, g, \mathfrak{Q})$, RoadMap$(f^{\boldsymbol{A}} + e_0/2, g, \mathfrak{Q})$)
$\mathfrak{S} \leftarrow$ Union$(\mathfrak{R}, \mathfrak{R})$
$B \leftarrow$ ConnectivityQuery$(\mathfrak{S}, \mathfrak{P}, x_{n+1} \neq 0, 0 < f^{\boldsymbol{A}} < e_0)$
return $(\mathfrak{P}, B)$

4 COMPLEXITY ANALYSIS

All complexity results are given in the number of arithmetic operations in $\boldsymbol{Q}$. Hereafter, we assume that a generic enough matrix $\boldsymbol{A}$ is found from a random choice. In order to end the proof of Theorem 1, we now estimate the arithmetic runtime of the calls to Candidates and IsIsolated.

Complexity of Algorithm 2. Since $W(\pi_i, \mathcal{H}_\varepsilon^{\boldsymbol{A}})$ is the finite algebraic set associated to $\left\langle f^{\boldsymbol{A}} - e, \frac{\partial f^{\boldsymbol{A}}}{\partial x_j} \text{ for all } j \neq i\right\rangle$, its degree is bounded by $d(d-1)^n$ [17]. Consequently, the degree of the output zero-dimensional parametrization lies in $d^{O(n)}$. Using [24, Theorem 6] (which is based on the geometric resolution algorithm in [14]), it is computed within $d^{O(n)}$ arithmetic operations in $\boldsymbol{Q}$. The last step which takes intersections is done using the algorithm in [28, Appendix J.1] ; it does not change the asymptotic complexity.

We have seen that GetNormBound reduces to isolate the real roots of a zero-dimensional parametrization of degree $d^{O(n)}$. This can be done within $d^{O(n)}$ operations by Uspensky's algorithm [23].

Complexity of Algorithm 3. Each call to SpecializationValue reduces to computing critical values of π_i of a smooth algebraic set defined by polynomials of degree $\leq d$. This is done using $(nd)^{O(n)}$ arithmetic operations in $\boldsymbol{Q}$ (see [15]). Using [14] for ZeroDimSolve and [29] for ParametricCurve does not increase the overall complexity. The loop is performed n times ; hence the complexity lies in $(nd)^{O(n)}$. All output zero-dimensional parametrizations have degree bounded by $d^{O(n)}$. Running Union on these parametrizations does not increase the asymptotic complexity. One gets then parametrizations of degree bounded by $nd^{O(n)}$. Finally, using [28] for Roadmap uses $(nd)^{O(n\log(n))}$ arithmetic operations in $\boldsymbol{Q}$ and outputs a rational parametrization of degree lying in $(nd)^{O(n\log(n))}$. The call to ConnectivityQuery, done as explained in [27] is polynomial in the degree of the roadmap.

The final steps which consist in calling Removes and undoing the change of variables does not change the asymptotic complexity.

Summing up altogether the above complexity estimates, one obtains an algorithm using $(nd)^{O(n\log(n))}$ arithmetics operations in $\boldsymbol{Q}$ at most. This ends the proof of Theorem 1.

5 EXPERIMENTAL RESULTS

We report on practical performances of our algorithm. Computations were done on an Intel(R) Xeon(R) CPU E3-1505M v6 @ 3.00GHz with 32GB of RAM. We take sums of squares of n random dense quadrics in n variables (with a non-empty intersection over $\boldsymbol{R}$) ; we obtain *dense quartics* defining a finite set of points. Timings are given in seconds (s.), minutes (m.), hours (h.) and days (d.).

We used Faugère's FGb library for computing Gröbner bases in order to perform algebraic elimination in Algorithms 1, 2 and 3. We also used our C implementation for bivariate polynomial system solving (based on resultant computations) which we need to analyze connectivity queries in roadmaps. Timings for Algorithm 2 are given in the column CAND below. Timings for the computation of the roadmaps are given in the column RMP and timings for the analysis of connectivity queries are given in the column QRI below.

Roadmaps are obtained as the union of critical loci of some maps in slices of the input variety [28]. We report on the highest degree of these critical loci in the column SRMP. The column SQRI reports on the maximum degree of the bivariate zero-dimensional system we need to study to analyze connectivity queries on the roadmap.

None of the examples we considered could be tackled using the implementations of Cylindrical Algebraic Decomposition algorithms in Maple and Mathematica.

We also implemented [2, Alg. 12.16] using the FLINT C library with evaluation/interpolation techniques instead to tackle coefficients involving infinitesimals. This algorithm only computes sample points per connected components. *That implementation was not able to compute sample points of the input quartics for any of our examples.* We then report in the column [BPR] on the degree of the zero-dimensional system which is expected to be solved by [2, BPR]. This is to be compared with columns SRMP and SQRI.

n	CAND	RMP	QRI	total	SRMP	SQRI	[BPR]
4	2 s.	15 s.	33 s.	50 s.	36	359	7290
5	$<$ 10 min.	1h.	7h.	8 h.	108	4644	65 610
6	$< 12h$	2 d.	18 d.	20 d.	308	47952	590 490

REFERENCES

[1] ARNON, D. S. A cluster-based cylindrical algebraic decomposition algorithm. *J. Symb. Comput. 5*, 1/2 (1988), 189–212.

[2] BASU, S., POLLACK, R., AND ROY, M.-F. *Algorithms in Real Algebraic Geometry (Algorithms and Computation in Mathematics).* Springer-Verlag, 2006.

[3] BASU, S., AND ROY, M. Divide and conquer roadmap for algebraic sets. *Discrete & Computational Geometry 52*, 2 (2014), 278–343.

[4] BASU, S., ROY, M., SAFEY EL DIN, M., AND SCHOST, É. A baby step-giant step roadmap algorithm for general algebraic sets. *Foundations of Computational Mathematics 14*, 6 (2014), 1117–1172.

[5] BORCEA, C., AND STREINU, I. Geometric auxetics. *Proc. R. Soc. Lond., A, Math. Phys. Eng. Sci. 471*, 2184 (2015), 24.

[6] BORCEA, C. S., AND STREINU, I. Periodic auxetics: structure and design. *Q. J. Mech. Appl. Math. 71*, 2 (2018), 125–138.

[7] COLLINS, G. E. Quantifier elimination for real closed fields by cylindrical algebraic decomposition: a synopsis. *ACM SIGSAM Bulletin 10*, 1 (1976), 10–12.

[8] CONNELLY, R., AND WHITELEY, W. The stability of tensegrity frameworks. *International Journal of Space Structures 7*, 2 (1992), 153–163.

[9] COSTE, M., AND SHIOTA, M. Thom's first isotopy lemma: a semialgebraic version, with uniform bounds. *RIMS Kokyuroku 815* (1992), 176–189.

[10] DUVAL, D. Rational puiseux expansions. *Compositio Mathematica 70*, 2 (1989), 119–154.

[11] GASPAR, N., REN, X., SMITH, C., GRIMA, J., AND EVANS, K. Novel honeycombs with auxetic behaviour. *Acta Materialia 53*, 8 (2005), 2439 – 2445.

[12] GIANNI, P. M., AND TEO MORA, T. Algebraic solution of systems of polynomial equations using Gröebner bases. In *Applied Algebra, Algebraic Algorithms and Error-Correcting Codes, 5th International Conference, AAECC-5, Menorca, Spain, Proceedings* (1987), pp. 247–257.

[13] GIUSTI, M., HEINTZ, J., MORAIS, J. E., AND PARDO, L. M. When polynomial equation systems can be "solved" fast? In *Applied Algebra, Algebraic Algorithms and Error-Correcting Codes, 11th International Symposium, AAECC-11, Paris, France, Proceedings* (1995), pp. 205–231.

[14] GIUSTI, M., LECERF, G., AND SALVY, B. A Gröbner free alternative for polynomial system solving. *Journal of Complexity 17*, 1 (2001), 154 – 211.

[15] GREUET, A., AND SAFEY EL DIN, M. Probabilistic algorithm for polynomial optimization over a real algebraic set. *SIAM Journal on Optimization 24*, 3 (2014).

[16] GRIMA, J. N., AND EVANS, K. E. Auxetic behavior from rotating triangles. *Journal of Materials Science 41*, 10 (May 2006), 3193–3196.

[17] HEINTZ, J. Definability and fast quantifier elimination in algebraically closed fields. *Theor. Comput. Sci. 24* (1983), 239–277.

[18] JELONEK, Z., AND KURDYKA, K. Quantitative generalized Bertini-Sard theorem for smooth affine varieties. *Discrete & Computational Geometry 34*, 4 (2005), 659–678.

[19] KRONECKER, L. Grundzüge einer arithmetischen theorie der algebraischen grössen. *Journal für die reine und angewandte Mathematik 92* (1882), 1–122.

[20] LAKES, R. Foam structures with a negative poisson's ratio. *Science 235*, 4792 (1987), 1038–1040.

[21] ROTH, B., AND WHITELEY, W. Tensegrity frameworks. *Trans. Am. Math. Soc. 265* (1981), 419–446.

[22] ROUILLIER, F., ROY, M., AND SAFEY EL DIN, M. Finding at least one point in each connected component of a real algebraic set defined by a single equation. *J. Complexity 16*, 4 (2000), 716–750.

[23] ROUILLIER, F., AND ZIMMERMANN, P. Efficient isolation of polynomial's real roots. *Journal of Computational and Applied Mathematics 162*, 1 (2004), 33–50.

[24] SAFEY EL DIN, M. Computing Sampling Points on a Singular Real Hypersurface using Lagrange's System. Research Report RR-5464, INRIA, 2005.

[25] SAFEY EL DIN, M., AND SCHOST, E. Polar varieties and computation of one point in each connected component of a smooth real algebraic set. In *Proc. of the 2003 Int. Symp. on Symb. and Alg. Comp.* (2003), ISSAC '03, ACM, p. 224–231.

[26] SAFEY EL DIN, M., AND SCHOST, É. Properness defects of projections and computation of at least one point in each connected component of a real algebraic set. *Discrete & Computational Geometry 32*, 3 (2004), 417–430.

[27] SAFEY EL DIN, M., AND SCHOST, É. A baby steps/giant steps probabilistic algorithm for computing roadmaps in smooth bounded real hypersurface. *Discrete & Computational Geometry 45*, 1 (2011), 181–220.

[28] SAFEY EL DIN, M., AND SCHOST, É. A nearly optimal algorithm for deciding connectivity queries in smooth and bounded real algebraic sets. *J. ACM 63*, 6 (Jan. 2017), 48:1–48:37.

[29] SCHOST, É. Computing parametric geometric resolutions. *Applicable Algebra in Engineering, Communication and Computing 13*, 5 (2003), 349–393.

[30] SCHWARTZ, J. T., AND SHARIR, M. On the "piano movers" problem. II. general techniques for computing topological properties of real algebraic manifolds. *Advances in Applied Mathematics 4*, 3 (1983), 298 – 351.

[31] VOROBJOV, N. Complexity of computing the local dimension of a semialgebraic set. *Journal of Symbolic Computation 27*, 6 (1999), 565–579.

[32] YANG, W., LI, Z.-M., SHI, W., AND XIE, B.-H. Review on auxetic materials. *Journal of Materials Science 39* (2004), 3269–3279.

LETTERPLACE — a Subsystem of SINGULAR for Computations with Free Algebras via Letterplace Embedding

Viktor Levandovskyy
Lehrstuhl D für Mathematik, RWTH Aachen University
Aachen, Germany
Viktor.Levandovskyy@math.rwth-aachen.de

Hans Schönemann
TU Kaiserslautern
Kaiserslautern, Germany
hannes@mathematik.uni-kl.de

Karim Abou Zeid
Lehrstuhl D für Mathematik, RWTH Aachen University
Aachen, Germany
karim.abou.zeid@rwth-aachen.de

ABSTRACT

We present the newest release of the subsystem of SINGULAR called LETTERPLACE which exists since 2009. It is devoted to computations with finitely presented associative algebras over fields and offers Gröbner(–Shirshov) bases over free algebras via the *Letterplace* correspondence of La Scala and Levandovskyy. This allows to use highly tuned commutative data structures internally and to reuse parts of existing algorithms in the non-commutative situation. The present version has been deeply reengineered, based on the experience with earlier and experimental versions. We offer an unprecedented functionality, some of which *for the first time in the history of computer algebra*. In particular, we present tools for elimination theory (via truncated Gröbner bases and via supporting several kinds of elimination orderings), dimension theory (Gel'fand-Kirillov and global dimension), and for homological algebra (such as syzygy bimodules and lifts for ideals and bimodules) to name a few. Another article in this issue is devoted to the extension of Gröbner bases to the coefficients in principal ideal rings including $\mathbb{Z}$, which is also a part of this release. We report on comparison with other systems and on some advances in the theory. Quite nontrivial examples illustrate the abilities of the system.

CCS CONCEPTS

• **Computing methodologies** → **Special-purpose algebraic systems**; **Algebraic algorithms**; • **Mathematics of computing** → **Mathematical software**.

KEYWORDS

Noncommutative algebra; Groebner bases; Algorithms; Free algebra; Tensor algebra; Computer Algebra System

ACM Reference Format:
Viktor Levandovskyy, Hans Schönemann, and Karim Abou Zeid. 2020. LETTERPLACE — a Subsystem of SINGULAR for Computations with Free Algebras via Letterplace Embedding. In *International Symposium on Symbolic and Algebraic Computation (ISSAC '20), July 20–23, 2020, Kalamata, Greece.* ACM, New York, NY, USA, 7 pages. https://doi.org/10.1145/3373207.3404056

ISSAC '20, July 20–23, 2020, Kalamata, Greece

ACM ISBN 978-1-4503-7100-1/20/07…$15.00
https://doi.org/10.1145/3373207.3404056

1 THE BASICS

All rings are assumed to be associative and unital, but not necessarily commutative. K stands for a field, X for the set $\{x_1, \ldots, x_n\}$, although for theoretical considerations it might be just enumerable.

La Scala and Levandovskyy [18, 19] established a one-to-one correspondence between the ideals of the free associative algebra $K\langle X\rangle$ and the so-called *letterplace ideals* of the infinitely generated commutative algebra $K[X \mid \mathbb{N}] = K[\{x_i(j) : x_i \in X, j \in \mathbb{N}]$. The latter is a difference ring with the natural injective endomorphism $\sigma : x_i(j) \mapsto x_i(j+1)$ for any i; letterplace ideals are stable under the action of σ. The mentioned correspondence extends to generating sets and, in particular, to Gröbner bases of ideals and submodules of free bimodules [3, 17, 24, 26]. Though developed alternatively to the way generalized Buchberger's algorithm works, nowadays it is possible to formulate the whole theory in a way, similar to Buchberger's. Nevertheless, merits of Letterplace approach lie in the predominantly commutative setup of data structures, which together with its functionalities can be widely reused in the Letterplace context. As an example of such we demonstrate new monomial orderings, available for usage. Also the creation of critical pairs and the criteria for discarding them can be seen in an almost one-to-one correspondence.

1.1 Models of computation

.

Since the algebras under consideration are not Noetherian, a typical Gröbner basis computation will not terminate. There are examples, where a finite Gröbner basis of an ideal (given by a finite generating set) exists for some specific orderings, and is infinite for the other. And there are situation, where a finite generating set has always an infinite Gröbner basis.

In order to deal with this situation, in the realm of associative algebras one employs a *bound*. It is often called a *degree bound*, but we propose - armed by a Letterplace wisdom - to call it rather a *length bound*, where the length of a monomial element from the algebra is literally its' length as the word in the alphabet. Whereas the *degree* might mean the weighted degree, imposed by the weights $w_i \in \mathbb{N}$ on each variable x_i, i.e. $\deg_w(x_{i_1} \cdot \ldots \cdot x_{i_k}) := \sum_{j=1}^{k} w_{i_j}$. Even when a generating set in the input is $\mathbb{N}$-graded, Buchberger's algorithm can recognize this only when a fixed monomial ordering respects the same grading.

Therefore a length bound $d \in \mathbb{N}$ is usually provided as an argument to such a computation, then the restricted Letterplace ring $K[X \mid \{1 \ldots d\}]$ is a finitely generated commutative Noetherian

ring, so any instance of Gröbner basis (even a difference Gröbner basis) will necessarily terminate.

On the other hand, Pritchard [28] (not widely known) has shown, that if a finite Gröbner basis of a given ideal exists with respect to a fixed monomial ordering, it will be found by generalized Buchberger's algorithm in a finite number of steps.

1.2 Encompassing Bimodules

Since we are interested not only in ideals, but in finitely presented modules, we have to work with *free bimodule of finite rank* over free and finitely presented algebras. For such an algebra R, let ε_i denote the i-th canonical generator of a free bimodule. Notably, it commutes *only* with the constants from the ground field. Then the free bimodule of rank $r \in \mathbb{N}$ is $\mathcal{F}_r := \bigoplus_{i=1}^{r} R\varepsilon_i R$. The Gröbner bases theory, developed in [4] and its Letterplace counterpart generalize to the setting of finitely (or countably) generated subbimodules of $\mathcal{F}_r$. We have implemented it and use it for working with e.g. (bi-)syzygy modules and (bi-)transformation matrices. The classical Singular types `vector` and `module` use the symbol `gen(i)` for the i-th canonical basis vector, which commutes with everything by default. We have extended this for Letterplace by introducing the symbol `ncgen(i)` for the i-th canonical generator of the free bimodule (ε_i above), which commutes only with constants. A combination `ncgen(i)·gen(i)` allows us to treat free bimodules effectively.

1.3 Monomial Orderings

One of the most pleasant effects of using Letterplace techniques is the representability of data in terms of commutative data structures. We utilized it especially in the work with monomial orderings. It turned out, that due to requirements of the Letterplace setup not all orderings (see, for example the taxonomy of such in [27]) can be realized. Nevertheless, many of useful orderings have been implemented — to the best of our knowledge, for the first time in a publicly available software project. The initialization of a free algebra happens in two steps. In the first one, a commutative ring is created, and in the second one, a free algebra is built from it subject to the provided length bound. From the commutative ring we extract information on a ground field or ring, on the names and the order of variables and on the monomial ordering. There is a bijection between the names for orderings, used in the commutative ring by Singular and their interpretation after the initialization of the free algebra. Namely, we provide the following monomial orderings, built over the ordered list of variables $x_1, \ldots, x_n$, which is specified by the user:

`dp`: degree right lexicographical ordering;
`Dp`: degree left lexicographical ordering;
`Wp(w)`: weighted degree left lexicographical ordering, with `w` being a strictly positive vector of weights for the variables;
`lp`: left total elimination ordering;
`rp`: right total elimination ordering;
`(a(v),<)` extra weight ordering extension, with `v` being a vector of nonnegative weights for the variables and `<` is another ordering from the above.

Moreover, for modules (bimodules) we offer both position-over-term and term-over-position constructions involving the monomial orderings as above.

1.4 Length-incompatible orderings

When working with monomial orderings, which are not length-compatible (which were studied by David Green e.g. in [11] though in a different context and with other motivation), the following error message can appear:

```
? degree bound of Letterplace ring is 10, but at least
11 is needed for this multiplication
```

After such a message any current computation, based on Gröbner bases is stopped. This is not a bug, but an indication that internally a potentially non-Noetherian reduction has been invoked. There is no other possibility to treat such a situation automatically. Practical advises include increasing the length bound and keep tail reduction active by setting `option(redTail)`.

1.5 Fundamental Functionality

With the current release, we offer the following functionalities for ideals and subbimodules of free bimodule of a finite rank.

`twostd(F)`: a two-sided Gröbner basis of F;
`reduce(p, G)`: a normal form of a vector or a polynomial p with respect to a two-sided Gröbner basis G;
`syz(F)`: a generating set of a syzygy bimodule of F;
`modulo(M, F)`: kernel of a bimodule homomorphism, defined by M into a bimodule, presented by the generators of F;
`lift(M, N)`: computation of a bi-transformation matrix between a module and its submodule, in other words expressing generators of a submodule N in terms of generators of a module M;
`liftstd(F, T[, S])`: computation of a two-sided Gröbner basis and a bi-transformation matrix T with (optionally) a syzygy bimodule S of F;
`rightStd(F)`: a right Gröbner basis of F.

Moreover, given a two-sided Gröbner basis G of an ideal, one can pass to the factor algebra of $K\langle X\rangle$ by $\langle G\rangle$: the data type `qring` offers such a passage and supports arithmetic operations. All of the above functionality is also provided for factor algebras – that is *for all finitely presented algebras*.

A *division with remainder algorithm* is provided via the classical pair `reduce` (producing a remainder) and `lift` (computing a two-sided presentation of an element in an ideal or a bimodule).

Letterplace technique is the key technology, used in the kernel. However, due to the new user-friendly interface, the user will not see much of Letterplace, but a native free algebra instead.

2 ADVANCED FUNCTIONALITY

Numerous applications can be approached with the functionality already present in the kernel. Below we give several examples of the usage. Using Letterplace in the kernel, we provide the following libraries, written in the interpreter C-like language of Singular. After extensively testing the important procedures we reimplemented the most time-consuming parts in the kernel.

`freegb.lib` is the main initialization library; it also contains numerous legacy, conversion and other technical routines.
`fpadim.lib` addresses computations of vector space dimensions, bases of finite dimensional or restricted algebras as well as finite Hilbert series.
`fpalgebras.lib` contains many relations of predefined algebras including various group algebras of finitely generated groups.
`fpaprops.lib` is one of our flagships. Suppose, that one has a finite Gröbner basis of an ideal $I \subset K\langle X\rangle$. Then there are procedures for the computations of Gel'fand–Kirillov dimension and an upper bound for the global dimension (the values of both are in $\mathbb{N}_0 \cup \{\infty\}$) of a finitely presented algebra $K\langle X\rangle/I$, as well as checks, whether this algebra is left/right/weak Noetherian and prime or semiprime [25]. Algebraic substitutions (ring morphisms) are offered as well.
`ncHilb.lib` is a third-party library, presenting tools for computing multi-graded Hilbert series of not necessary finitely presented algebras.

3 EXAMPLE SESSION

Consider the ideal $\mathcal{I} = \langle f_1 = yx - 3xy - 3z,\ f_2 = zx - 2xz + y,\ f_3 = zy - yz - x\rangle \subset \mathbb{Q}\langle z, y, x\rangle$. Let us initialize the free algebra and compute a Gröbner basis of I with respect to the degree left lexicographic ordering with $z > y > x$. This is done in two steps: at first, a commutative ring over $\mathbb{Q}$ is created. From this ring, the names of variables, their sequence as well as the monomial ordering are read. Finally, `freeAlgebra` creates the free algebra from this information (stored in a ring) subject to the explicit length bound.

```
LIB "freegb.lib"; //initialization of free algebras
ring r = 0,(z,y,x),Dp; // degree left lex ord on z>y>x
ring R = freeAlgebra(r,5); // put length bound to 5
ideal I = y*x - 3*x*y - 3*z, z*x - 2*x*z +y, z*y-y*z-x;
option(redSB); // Groebner basis will be minimal
option(redTail); // Groebner basis will be tail-reduced
ideal J = twostd(I); // compute a two-sided GB of I
J; // print generators of J
```

The output is a finite Gröbner basis of 9 polynomials

$$\{4xy + 3z,\ 3xz - y,\ 4yx - 3z,\ 2y^2 - 3x^2,\ 2yz + x,$$
$$3zx + y, 2zy - x,\ 3z^2 - 2x^2,\ 4x^3 + x\}.$$

Note, that the same generating set produces an infinite Gröbner basis over $\mathbb{Z}\langle z, y, x\rangle$ as we report in [22]. As we see, original generators have decomposed. Computing their expressions in terms of the Gröbner basis above is a typical application of the command `lift`. However, since bimodule presentation is involved, we need to activate a free bimodule of a fixed rank.

```
setring r;
ring R2 = freeAlgebra(r,8,9); // 9 = #elements of J
ideal I = imap(R,I); // transfers I from R to R2
ideal J = imap(R,J);
matrix P = lift(J,I);
print(transpose(P[1]));
> -3/4*ncgen(1),0,1/4*ncgen(3)
```

This means, that the first generator of I is presented as a linear combination of the 1st and the 3rd generators of J:

$$yx - 3xy - 3z = -\frac{3}{4}\cdot(4xy + 3z) + \frac{1}{4}\cdot(4yx - 3z).$$

In order to obtain the expressions for generators of J is terms of I, we can use `liftstd`

```
matrix T;  ideal J2 = liftstd(I, T);
print(T[7][3]);
> -30*z*x*ncgen(3)*y+5*z*ncgen(3)*x*y+...
```

where the latter is the beginning of the lengthy expression of the 7th element of J through the 3rd element of I.

The form of leading monomials of J raises a conjecture, that $\mathbb{Q}\langle x, y, z\rangle/J$ is finite dimensional $\mathbb{Q}$-vector space. Let us check it:

```
setring R;
LIB "fpadim.lib"; // load the library for K-dimensions
lpKDim(J); // determinte the K-dimension of R/J
> 5
```

So, the dimension is 5. What is the canonical monomial $\mathbb{Q}$-basis?

```
lpMonomialBasis(5,0,J); // compute all monomials
// of length up to 5 in Q<x,y,z>/J
```

which results in $\{1, z, y, x, x^2\}$. In a finite-dimensional algebra every element is clearly algebraic. What is the minimal polynomial of, say, $y + z$?

```
poly p = y+z;
ideal B = 1, p, NF(p^2,J), NF(p^3,J), NF(p^4,J);
LIB "bfun.lib";
list L = linReduceIdeal(B,1); //looks for lin. dep.
L[2][1];
> 24/13*gen(4)+gen(2)
reduce(24/13*p^3+p, J);
> 0
```

what means that the minimal polynomial of $p = y + z$ is $t^3 + \frac{13}{24}t$. In the code, `linReduceIdeal` computed the linear dependencies among the elements of the ideal B, treated as a list. Into B we have entered the normal forms of the consecutive powers of p. As we see, 3rd power of p is sufficient in this computation.

4 THE HIGH ART OF ELIMINATION

In the first Weyl algebra $A_1(\mathbb{Q}) = \mathbb{Q}\langle x, \partial \mid \partial x = x\partial + 1\rangle$, consider the subalgebra S, generated by $\{x\partial^2, x^2\partial\}$. A very natural question in algebra (or ring theory) is *how to describe S as a finitely presented algebra*? In other words, what is the kernel of the homomorphism of $\mathbb{Q}$-algebras

$$\mathbb{Q}\langle a, b\rangle \to \mathbb{Q}\langle x, \partial\rangle/\langle \partial x - x\partial - 1\rangle,\ a \mapsto x\partial^2,\ b \mapsto x^2\partial.$$

Below, we give a conjectural answer and explain why it is only conjectural.

Another natural question, this time rather from the side of differential equations, is *does the Euler derivation $x\partial$ belong to S*? Below, we present an affirmative answer with the explicit presentation.

```
LIB "freegb.lib";
ring r = 0,(x,d,a,b,c),(a(1,1,0,0,0),dp);
ring R = freeAlgebra(r,40);
```

We use an extra weight monomial ordering, eliminating variables $\{x, d\}$ and set up the length bound 40 for the Letterplace ring `R`. c will stand for the Euler derivation $x\partial$, b for $x^2\partial$ and a for $x\partial^2$. We enter all these relations into the ideal I and compute its two-sided Gröbner basis, which we denote by J.

```
option(redSB);option(redTail);
ideal I = d*x-c-1, x*d -c, c*d-a, x*c-b;
ideal J = twostd(I);
```

J has 244 elements up to leading length 40, what suggests that this Gröbner basis will probably be infinite. Nevertheless, we find the following elements after eliminating variables (by the Elimination Lemma as in e.g. [5]):

$$\{cb - bc - b,\ ca - ac + a,\ ba - ab + 3c^2 - c,\ c^3 - ab + c^2\}.$$

We conjecture, that these relations completely describe the algebra, generated by $\{a, b, c\}$. Now, let us attack this ideal of relations with the left total elimination ordering in the algebra $\mathbb{Q}\langle a, b, c\rangle$ with $c \succ b \succ a$

```
ring r2 = 0,(c,a,b),lp;
ring R2 = freeAlgebra(r2,20);    ideal I2;
I2 = c*b-b*c-b,c*a-a*c+a,b*a-a*b+3*c^2-c,c^3-a*b+c^2;
option(redSB);option(redTail);
ideal J2 = twostd(I2);
```

The ideal `J2` has 10 generators of leading length up to 5, so it is a finite Gröbner basis of `I2`. The most interesting, however is the last generator. From it we extract the *proof* of the fact that $x\partial \in S$:

$$c = -\tfrac{1}{40}(\,6(ab)^2 - 21ba^2b + 24(ba)^2 - 9b^2a^2 - 32ab - 76ba\,).$$

Hence the other 9 generators of the ideal `J2` are expressed in variables $\{a, b\}$ only and *conjecturally* provide the complete description S as finitely presented algebra. Why can we say only conjecturally? It remains to prove that the mentioned 9 relations (i.e. polynomials in $\mathbb{Q}\langle a, b\rangle$ from the ideal `J2`), which look as

$$ab^3 - 3bab^2 + 3b^2ab - b^3a - 6b^2, \ldots,$$

$9a^2bab - 108ba^2ba + 171baba^2 - 72b^2a^3 + 34a^2b - 800aba - 638ba^2 - 104a$

generate S. Thus we have to make a conjecture, that an infinite Gröbner basis of the ideal J above can be written in terms of finitely many families with parametric exponents, which we need to write down explicitly. Then we need to prove by hands that these families indeed constitute an infinite Gröbner basis of the ideal J. Only then can we apply the Elimination Lemma in order to conclude, that the 9 elements in a, b, obtained from 4-generated ideal `I2` as above, are *all* relations of the algebra S.

Questions similar to those above can be effectively attacked with the *assistance* of our implementation. We stress that this assistance, in the case where no finite Gröbner basis is found, helps to make a conjecture on the form of the infinite Gröbner basis. And no other ways of proving such statements except proving by hands are yet known to us.

In general, the kernel of a ring morphism (see Theorem 5.3), the preimage of an ideal (two-sided or one-sided [20]) under a ring morphism, algebraic dependencies, algebraicity of elements and many more become accessible with working Gröbner bases-based elimination [5].

A practicing user has to be aware that — in distinct contrast to computations with Noetherian algebras, which are close to commutative — instead of "one click" automated solution one has a task, requiring human guidance. In such a setup Gröbner bases tend to be infinite, and even uncomputable because of the issue, described in Section 1.4. Therefore the vast choice of monomial orderings for elimination is of huge importance.

Remark 4.1. Consider the algebra, generated by $\{a, b, c\}$ subject to conjectural relations from the above:

$$\{cb - bc - b,\ ca - ac + a,\ ba - ab + 3c^2 - c,\ c^3 - ab + c^2\}.$$

The Gröbner basis property of the ideal of relations imply, that we are dealing with the factor algebra of a G-algebra [23] by a two-sided ideal

$$K\langle a, b, c \mid ba = ab - 3c^2 + c,\ cb = bc + b,\ ca = ac - a\rangle / \langle c^3 - ab + c^2\rangle.$$

Algebras of this type, called *GR*-algebras, are very pleasant: they are Noetherian and possess computable finite Gröbner bases for their ideals. Another subsystem of Singular called Plural [12, 23] offers a very broad spectrum of functionality for such algebras. We stress that in the category of *GR*-algebras one does not experience problems with computability, discussed above. Also, concrete computations with modules directly in a *GR*-algebra will be, of course, much faster than addressing such from Letterplace.

5 ALGORITHMIC DEVELOPMENTS

The Gel'fand-Kirillov dimension [15] measures the growth of algebras and modules. It is one of few dimensions, which exist and are sometimes computable over general finitely presented algebras. As soon as a finite Gröbner basis of the ideal of relations is known, the Gel'fand-Kirillov dimension of such an algebra is either a natural number or ∞ (which corresponds to at least exponential growth and is present in a free algebra in at least two generators itself).

5.1 Better usage of Ufnarovski graph

Using the Ufnarovski graph [33], we developed a new method to compute the Gel'fand-Kirillov dimension. Given the Ufnarovski graph, which is a regular directed graph, one has to count the maximum number of distinct cycles occuring in a single route. There is one exception — if there are two distinct cycles with a common vertex in the graph, then one is done immediately (i.e. the GKdim is ∞). Note that in general, computing the total number of simple cycles in a directed graph is a problem also known as #CYCLE and it cannot be solved in polynomial time unless P=NP [1]. This does not apply to our case, because of the exception mentioned above.

Every neighbor of every vertex in the input graph is only evaluated once and thus Algorithm 1 has a linear runtime of $O(|V| + |E|)$.

5.2 Computing Gel'fand–Kirillov dimension of finitely presented bimodules

Let $R = K\langle X\rangle$ be a free algebra. Ufnarovski has developed a method for computing the important Gel'fand–Kirillov dimension of an algebra (and a cyclic bimodule) R/I for a two-sided ideal $I \subset R$, which is given via a *finite monomial* generating set. We refer to such algorithm as to GKDim-MonomialIdeal. It is somewhat strange, that the problem of generalizing this approach from a cyclic to a finitely presented bimodule has not been addressed before in the situation of a free algebra. We present an algorithmic solution.

Algorithm 1 Maximum cycle count

Input: A directed graph (V, E).
Output: The maximum number of distinct cycles in a single route in (V, E); or ∞ if there is a route in (V, E) which contains two distinct cycles with a common vertex.
Note: The maximum cycle count values are shared globally during the whole algorithm. Every other variable (visited, cyclic and the graph) is local to each recursion step and is only passed forward to subsequent recursion steps.

For every vertex $v \in V$, if it has not been computed yet, compute the maximum cycle count from v as described below. Finally return the maximum value.

(1) Mark v as *visited*.
(2) For every unvisited neighbor w of v, recursively compute the maximum cycle count (1) from w if it has not been computed yet.
(3) For every visited neighbor w of v (i.e. a neighbor that leads to a cycle):
  (a) If any vertex of the cycle is marked as *cyclic* (i.e. it is part of a different cycle), set the maximum cycle count from v to ∞ and stop here.
  (b) Mark all vertices of the cycle as *cyclic*.
  (c) Remove all edges from the cycle.
  (d) For every vertex of the cycle, recursively compute the maximum cycle count (1) from that vertex if it has not been computed yet.
(4) Set the maximum cycle count from v to the maximum of the maximum cycle count of the neighbors in (2) and 1 + the maximum cycle count of the neighbors in (3).

Algorithm 2 Gel'fand–Kirillov Dimension GKDim-module(G)

Input: $G \subset \bigoplus_{i=1}^{r} R\varepsilon_i R$, a finite two-sided Gröbner basis of a subbimodule with respect to a term-over-position ordering, based on a positively weighted degree ordering
Output: Gel'fand–Kirillov dimension (from $\mathbb{N} \cup \{\infty\}$) of $\bigoplus_{i=1}^{r} R\varepsilon_i R/\langle G\rangle$
$G \leftarrow \mathrm{lm}(G)$
for $i = 1$ to r **do**
  $F_i \leftarrow \{g \in G \mid g = m\varepsilon_i m',\ \text{for } m, m' \in \langle X\rangle\}$
return Sup($\{$GKDim-MonomialIdeal$(F_i) : 1 \le i \le r\}$)

Recall, that a $\mathbb{Z}$-filtration on an K-algebra A is a family of K-vector spaces $\{A_i, i \in \mathbb{Z}\}$, such that $A = \cup A_i$, $A_i \subseteq A_{i+1}$ and $A_i \cdot A_j \subseteq A_{i+j}$ for $i, j \in \mathbb{Z}$. It is called *finite-dimensional*, if $\dim_K A_i < \infty$ for all i.

From such a filtration one creates the *associated graded algebra* $gr(A) := \oplus_{i\in\mathbb{Z}} A_{i+1}/A_i$. A filtration on an A-bimodule M is also a is a family of K-vector spaces $\{M_i : i \in \mathbb{Z}\}$, such that $M = \cup M_i$, $M_i \subseteq M_{i+1}$ and $A_i \cdot M_j, M_i \cdot A_j \subseteq M_{i+j}$ for $i, j \in \mathbb{Z}$. This results in the *associated graded (bi-)module* $gr(M) := \oplus_{i\in\mathbb{Z}} M_{i+1}/M_i$.

Lemma 5.1. *For a monomial ordering $\prec$ on a K-algebra A, there is an associated filtration on A, constructed as follows. Let us fix a multiplicative K-basis $\mathcal{M}$ of* monomials *of A, then $\forall \mu \in \mathcal{M}$ we define $\mathcal{F}_\mu := {}_K\langle\{\nu \in \mathcal{M} : \nu \preceq \mu\}\rangle$. If $\prec_w$ is a positively weighted degree ordering on $R := K\langle X\rangle$, then $\{F_\mu : \mu \in \mathbb{N}_0\}$ is a finite dimensional filtration on R. Moreover, any term-over-position ordering, based on $\prec_w$, results in an induced finite dimensional $\mathbb{N}_0$- filtration on the free bimodule of finite rank $\bigoplus_{i=1}^{r} R\varepsilon_i R$.*

Lemma 5.2. *Algorithm 2 is correct and terminates.*

Proof. By Lemma 5.1 and the input specification it follows, that $gr(R) = R$ and $gr(M)$ is finitely generated as R-module.
By [15, Prop. 6.6] we have

$$\mathrm{GKdim}_R(M) = \mathrm{GKdim}_{gr(R)}(gr(M)) = \mathrm{GKdim}_R(gr(M)).$$

Since $gr(M)$ is the module, presented by the subbimodule of leading terms of a Gröbner basis, we have to determine the Gel'fand–Kirillov dimension of a finitely generated bimodule, presented by a monomial subbimodule of the free bimodule of a finite rank. The classical Proposition [15, Prop. 5.1] establishes the supremum argument over the component ideals. The termination is evident. □

5.3 Repairing error in the book "Gröbner Bases in Ring Theory"

The book by Huishi Li "Gröbner Bases in Ring Theory" [25] was a motivating companion to us. It contains a number of examples, illustrating the presented algorithms. We have detected computationally, that in Example 5 (Chapter 5.3, p. 169), there is a mistake, which we correct by theoretical means here. We stick to the notations, used in the book.

Theorem 5.3. *Over the free associative algebra $A = K\langle X_1, X_2\rangle$ consider the family of ideals $I_n = \langle X_2^n X_1\rangle$ for $n \in \mathbb{N}_0$. Then the following holds*

- *if $n = 0$,* $\mathrm{GKdim}\, A/I_0 = 1 = \text{gl. dim}\, A/I_0$,
- *if $n = 1$,* $\mathrm{GKdim}\, A/I_1 = 2 = \text{gl. dim}\, A/I_0$,
- *for $n \ge 2$,* $\mathrm{GKdim}\, A/I_n = \infty$ *and* $\text{gl. dim}\, A/I_n = 2$.

Moreover, for $n \ge 2$ A/I_n contains a free algebra in two variables.

Remark 5.4. In the book [25] it has been wrongly stated that for $n \ge 1$ one has $\mathrm{GKdim}\, A/I_n = 2$. Moreover, this information was used in Corollary 7.7 (p. 186) and in Example 2 (p. 194), where it was stated that $\text{gl. dim}\, A/I_n = 2$ for $n \ge 1$ holds as well, while after our correction only $\text{gl. dim}\, A/I_n \le 2$ for $n \ge 2$ follows directly. Nevertheless, we prove by additional computer-supported arguments, that the equality holds indeed.

Proof. The first claim follows since $A/I_0 \cong K[X_1]$. We note, that $I_0 = \langle X_1\rangle \supset I_1 = \langle X_2X_1\rangle \supset \ldots \supset I_n = \langle X_2^n X_1\rangle \supset \ldots$, hence for all n we have a natural surjection $K\langle X_1, X_2\rangle/I_{n+1} \to K\langle X_1, X_2\rangle/I_n$, thus for all n $\mathrm{GKdim}\, A/I_{n+1} \ge \mathrm{GKdim}\, A/I_n$.

Let $n = 2$, then the construction of the Ufnarovski Graph shows that $X_1X_2 \to X_2X_1 \to X_1X_2$ and $X_1X_2 \to X_2X_1 \to X_1^2 \to X_1X_2$ are two different cycles with a common vertex. Thus by [25, Theorem 3.1] $\mathrm{GKdim}\, A/I_2 = \infty$ and by the above, $\mathrm{GKdim}\, A/I_n \ge \mathrm{GKdim}\, A/I_2 = \infty$ for all $n \ge 2$.

At the same time, for $n \geq 1$ from the Ufnarovski Graph and [25, Theorem 7.4] we infer, that gl. dim $A/I_n \leq 2$. By the same result, for $n = 1$, since GKdim $A/I_1 = 2$ we have the equality gl. dim $A/I_1 = 2$. Now we show, that for $n \geq 2$ A/I_n contains a free algebra in two generators. Consider a homomorphism of K-algebras

$$\varphi : K\langle a, b\rangle \to K\langle X_1, X_2\rangle / \langle X_2^n X_1\rangle, \quad a \mapsto X_2X_1, b \mapsto X_1X_2.$$

We claim that φ is injective. As we know by e. g. Borges-Borges [5],

$$\ker \varphi = \langle X_2X_1 - a, X_1X_2 - b, X_2^nX_1\rangle \cap K\langle a, b\rangle,$$

while the ideal belongs to $K\langle X_1, X_2, a, b\rangle$. A lengthy technical computation by hands delivers, that a Gröbner basis of the ideal above with respect to a monomial ordering, eliminating $\{a, b\}$ is

$$\{X_2X_1 - a, X_1X_2 - b, X_2^{n-1}a, bX_2^{n-2}a, X_2b - aX_2, X_1a - bX_1\},$$

which has zero intersection with $K\langle a, b\rangle$. Therefore $\ker \varphi = \{0\}$ and for any $n \geq 3$ the algebra A/I_n contains a free algebra, e. g. generated by $\{X_2X_1, X_1X_2\}$.

In a similar (and yet easier) fashion we can show, that $\{X_1, X_1X_2\}$ generate a free subalgebra of A/I_n with $n \geq 2$.

In order to prove that gl. dim$(A/I_n) = 2$, it is enough to provide an explicit free resolution. We consider the monomial ideal $J = \langle X_1X_2, X_2X_1\rangle \subset A/I_n$, which is given by its Gröbner basis.

The methodology, we used in the proof, is as follows: we did computations with Letterplace for several n (in the code below $n = 3$ is used), conjectured the pattern (which is possible due to finiteness of Gröbner bases), and made a proof by executing Gröbner bases by hands. Since the latter is rather technical, such computations of syzygies (addressed in details in [4]) are omitted.

```
LIB "freegb.lib";
ring r = 0,(X1,X2),(c,dp);
ring R = freeAlgebra(r,10,7);
int n = 3;  ideal In = twostd(X2^n*X1);
qring Q = In; // Q = A/In = A/ <X2^3*X1>
ideal J = X2*X1, X1*X2;  J = twostd(J); J;
> X1*X2, X2*X1
option(redSB);option(redTail);
module S1, S2, S3;
S1 = syz(J); S2 = syz(S1); S3 = syz(S2);
```

The first syzygy bimodule of J is generated by the column vectors

$$\begin{pmatrix} \varepsilon_1X_2^{n-1}X_1 & \varepsilon_1X_2X_1 & X_2X_1\varepsilon_1 & \varepsilon_1X_1 & X_2\varepsilon_1 & 0 \\ 0 & -X_1X_2\varepsilon_2 & -\varepsilon_2X_1X_2 & -X_1\varepsilon_2 & -\varepsilon_2X_2 & X_2^{n-1}\varepsilon_2 \end{pmatrix},$$

the second syzygy bimodule is generated by the columns

$$\begin{pmatrix} 0 & \varepsilon_1X_2X_1 & \varepsilon_1X_1X_2 & \varepsilon_1X_2^{n-1}X_1 & 0 \\ \varepsilon_2X_2^{n-1}X_1 & X_2^{n-1}\varepsilon_2X_1 & 0 & X_2^{n-1}\varepsilon_2X_2X_1 & 0 \\ 0 & -X_2^n\varepsilon_3 & 0 & 0 & 0 \\ 0 & 0 & X_2^{n-1} & 0 & \varepsilon_4X_2^{n-1}X_1 \\ 0 & 0 & 0 & -X_2^n\varepsilon_5 & 0 \\ -X_2\varepsilon_6 & 0 & 0 & 0 & -X_2X_1\varepsilon_6 \end{pmatrix},$$

and the third syzygy bimodule – by the single column

$$\left(X_2^{n-1}\varepsilon_1, 0, -\varepsilon_3X_2^{n-1}X_1, 0, 0\right)^T.$$

It is easy to see, that the column subbimodules, generated by sets above, are given by minimal bimodule generating sets. Also, the 4th syzygy bimodule can be only zero since the single column cannot be annihilated other than by zero. Therefore we have obtained a free bimodule resolution of the A/I_n-bimodule $(A/I_n)/J$ of an appropriate length, what finishes the proof. □

6 TIMINGS

In the paper [18] we have compared computer algebra systems, able to compute over free algebras, on the collection on carefully selected examples. It turned out, that Opal, Bergman [10] and GBNP [7] have been outperformed by Magma [6] and Singular [8]. Therefore since that time we compare our implementations of Singular:Letterplace with two implementations in Magma, namely a Buchberger-like – which is the most correct comparison – and an $F4$-like implementation, which is a priori faster [19, 24]. However, the latter (since many years) remains the only big shot of functionality, available in Magma apart from the *vector enumeration* (which seems to trace back to the famous code of Steve Linton). On the contrast, we have developed a vast functionality including syzygies, lifts, elimination and other demanding applications.

There are other implementations of Gröbner bases over free algebra, which are, however, not easy to acquire, to install and to use. Still, we mention the package NCPoly for ApCoCoA [16, 34] and a C++ library called NCAlgebra [14], interfaced via Mathematica.

The following examples were executed on a Debian GNU/Linux 10 machine with an Intel Core i7-9700K CPU and 64GB of memory. We have used Singular 4.1.2 and Magma V2.24-10. The execution time is given in seconds. The examples we use have been described in our previous papers [18, 19, 24]. We utilize the SDEval Benchmarking Toolkit [13], extending the SymbolicData project [32] for the automated, transparent and *reproducible* comparison. In order to reproduce the timings on the mentioned examples, one needs to download the single archived file from

http://www.math.rwth-aachen.de/~levandov/issac2020,

and to install SDEval with SymbolicData, the sources of which can be found at

https://symbolicdata.github.io/SDEval.

The video presentation of the abilities of the Toolkit can be helpful:

https://www.youtube.com/watch?v=CctmrfisZso.

As for criteria for discarding useless critical pairs in a Buchberger-like algorithm, the product criterion means just that the leading monomials of elements, building a pair, do not have a non-trivial overlap, so such pairs are not even built. On the contrary, the chain criterion is the most important one in the implementation. See [22] for a detailed discussion on the criteria in a more general setting, encompassing the one considered here.

7 CONCLUSION

With our implementation in [21] we present – for the first time in history of computer algebra – the rich and fast infrastructure for computations with finitely presented algebras over fields, supported by Singular. Already the preliminary versions have been used in computations, which led to publications like [9, 29].

This computational infrastructure is available in a broader context to the visionary open source system OSCAR [31]. We are also aware of several wrappers of Letterplace in SageMath [30].

Example	Singular	Magma (BB)	Magma (F4)
lascala_neuh_d10	30.35	26.23	13.62
serre-f4-d15	5.45	62.58	8.96
serre-ha11-d15	11.27	49.63	5.80
serre-eha112-d13	2.05	3.41	1.72
4nilp5s-d8	36.68	55.43	9.54
braidXY	114.19	163.72	4.20
ug2-x1x2x3x4	1.10	21.60	0.83
serre-e6-d15	22.76	154.03	40.99
braid3-11	1.79	2.29	0.64
ufn3	70.22	3.36	2.26
ls3nilp-d10	0.72	3.57	1.98

We plan to provide more functions for matrices and make our implementation available to the system HomAlg [2], specializing in homological computations, as a backend.

Notably, there is a recent parallel development of free non-commutative Gröbner bases over $\mathbb{Z}$ by our team and Tobias Metzlaff [22], where the functionality is almost as broad as the one we have described for the case of fields.

8 ACKNOWLEDGEMENTS

The authors are grateful to Eva Zerz, Leonard Schmitz (RWTH Aachen), Tobias Metzlaff (INRIA), Michela Ceria and Teo Mora (Genova) and Jorge Martìn-Morales (Zaragoza) for fruitful discussions.

The first and third authors (V. Levandovskyy and K .Abou Zeid) have been supported by Project II.6 of SFB-TRR 195 "Symbolic Tools in Mathematics and their Applications" of the German Research Foundation (DFG).

REFERENCES

[1] Sanjeev Arora and Boaz Barak. 2009. *Computational Complexity: A Modern Approach* (1st ed.). Cambridge University Press, USA.
[2] Mohamed Barakat, Sebastian Gutsche, and Markus Lange-Hegermann. 2019. homalg - A homological algebra meta-package for computable Abelian categories. https://homalg-project.github.io/homalg_project/homalg/.
[3] George M. Bergman. 1977. The diamond lemma for ring theory. *Adv. Math.* 29 (1977), 178–218. https://doi.org/10.1016/0001-8708(78)90010-5
[4] Holger Bluhm and Martin Kreuzer. 2007. Computation of two-sided syzygies over non-commutative rings. *Contemp. Math.* 421 (2007), 45–64.
[5] M. A. Borges and M. Borges. 1998. Gröbner bases property on elimination ideal in the noncommutative case. In *Gröbner bases and applications*, B. Buchberger and F. Winkler (Eds.). Cambridge University Press, 323–337.
[6] W. Bosma, J. Cannon, and C. Playoust. 1997. The Magma algebra system. I: The user language. *Journal of Symbolic Computation* 24, 3-4 (1997), 235–265.
[7] Arjeh M. Cohen, J. W. Knopper, and T. GAP Team. 2016. GBNP, computing Gröbner bases of noncommutative polynomials (Refereed GAP package). https://gap-packages.github.io/gbnp/.
[8] Wolfram Decker, Gert-Martin Greuel, Gerhard Pfister, and Hans Schönemann. 2020. Singular 4-1-3 — A computer algebra system for polynomial computations. http://www.singular.uni-kl.de.
[9] C. Eder, V. Levandovskyy, J. Schanz, S. Schmidt, A. Steenpass, and M. Weber. 2019. Existence of quantum symmetries for graphs on up to seven vertices: a computer based approach. https://arxiv.org/abs/1906.12097.
[10] J. Backelin et. al. 2006. The Gröbner basis calculator Bergman. http://servus.math.su.se/bergman/.
[11] David J. Green. 2003. *Gröbner bases and the computation of group cohomology*. Springer.
[12] Gert-Martin Greuel, Viktor Levandovskyy, Olexander Motsak, and Hans Schönemann. 2019. Plural. A Singular 4-1-2 Subsystem for Computations with Non-commutative Polynomial Algebras. http://www.singular.uni-kl.de.
[13] Albert Heinle and Viktor Levandovskyy. 2015. The SDEval Benchmarking Toolkit. *ACM Communications in Computer Algebra* 49, 1/4 (2015), 1–9. https://doi.org/10.1145/2768577.2768578
[14] J.W. Helton and M. Stankus. 2015. NCGB, a Noncommutative Gröbner Basis Package for Mathematica. http://www.math.ucsd.edu/~ncalg/
[15] Günter R. Krause and Thomas H. Lenagan. 2000. *Growth of algebras and Gelfand-Kirillov dimension. Revised ed.* Providence, RI: American Mathematical Society.
[16] Martin Kreuzer. 2013. ApCoCoA, a computer algebra framework. http://https://apcocoa.uni-passau.de/
[17] Roberto La Scala. 2014. Extended letterplace correspondence for nongraded noncommutative ideals and related algorithms. *Int. J. Algebra Comput.* 24, 8 (2014), 1157–1182.
[18] Roberto La Scala and Viktor Levandovskyy. 2009. Letterplace ideals and non-commutative Gröbner bases. *Journal of Symbolic Computation* 44, 10 (2009), 1374–1393. https://doi.org/doi:10.1016/j.jsc.2009.03.002
[19] Roberto La Scala and Viktor Levandovskyy. 2013. Skew polynomial rings, Gröbner bases and the letterplace embedding of the free associative algebra. *Journal of Symbolic Computation* 48, 1 (2013), 110–131. http://dx.doi.org/10.1016/j.jsc.2012.05.003
[20] Viktor Levandovskyy. 2006. Intersection of Ideals with Non-commutative Subalgebras. In *Proc. ISSAC'06*, J.-G. Dumas (Ed.). ACM Press, 212–219.
[21] Viktor Levandovskyy, Karim Abou Zeid, and Hans Schönemann. 2020. Singular:Letterplace — A Singular 4-1-3 Subsystem for Non-commutative Finitely Presented Algebras. http://www.singular.uni-kl.de.
[22] Viktor Levandovskyy, Tobias Metzlaff, and Karim Abou Zeid. 2020. Computation of free non-commutative Gröbner Bases over $\mathbb{Z}$ with Singular:Letterplace. In *Proc. ISSAC'20*. ACM Press. to appear.
[23] Viktor Levandovskyy and Hans Schönemann. 2003. Plural - a computer algebra system for noncommutative polynomial algebras. In *Proc. ISSAC'03*. ACM Press, 176–183.
[24] Viktor Levandovskyy, Grischa Studzinski, and Benjamin Schnitzler. 2013. Enhanced Computations of Gröbner Bases in Free Algebras as a New Application of the Letterplace Paradigm. In *Proc. of the International Symposium on Symbolic and Algebraic Computation (ISSAC'13)*, Manuel Kauers (Ed.). ACM Press, 259 – 266.
[25] Huishi Li. 2002. *Noncommutative Gröbner bases and filtered-graded transfer*. Springer.
[26] Teo Mora. 1994. An introduction to commutative and noncommutative Gröbner bases. *Theor. Comput. Sci.* 134, 1 (1994), 131–173. https://doi.org/10.1016/0304-3975(94)90283-6
[27] Teo Mora. 2016. *Solving Polynomial Equation Systems IV: Volume 4, Buchberger Theory and Beyond* (1st ed.). Cambridge University Press.
[28] F. Leon Pritchard. 1996. The ideal membership problem in non-commutative polynomial rings. *J. Symb. Comput.* 22, 1 (1996), 27–48. https://doi.org/10.1006/jsco.1996.0040
[29] Leonard Schmitz and Viktor Levandovskyy. 2020. Formally Verifying Proofs for Algebraic Identities of Matrices. To appear in *Proc. of the Conference on Intelligent Computer Mathematics* (CICM) 2020.
[30] W. A. Stein et al. 2020. *Sage Mathematics Software*. The Sage Development Team.
[31] The OSCAR Team. 2020. The OSCAR project. https://oscar.computeralgebra.de.
[32] The SymbolicData Project. 2019. https://symbolicdata.github.io.
[33] Victor Ufnarovski. 1995. *Combinatorial and Asymptotic Methods of Algebra*. Algebra-VI (A.I. Kostrikin and I.R. Shafarevich, Eds), Encyclopedia of Mathematical Sciences, Vol. 57. Springer.
[34] Xingqiang Xiu. 2013. Ncpoly package for ApCoCoA. https://apcocoa.uni-passau.de/wiki/index.php/Category:Package_ncpoly

Computation of Free Non-commutative Gröbner Bases over $\mathbb{Z}$ with Singular:Letterplace

Viktor Levandovskyy
Lehrstuhl D für Mathematik, RWTH Aachen University
Aachen, Germany
Viktor.Levandovskyy@math.rwth-aachen.de

Tobias Metzlaff
AROMATH, INRIA Méditerranée
Université Côte d'Azur
Sophia Antipolis, France
tobias.metzlaff@inria.fr

Karim Abou Zeid
Lehrstuhl D für Mathematik, RWTH Aachen University
Aachen, Germany
karim.abou.zeid@rwth-aachen.de

ABSTRACT

The extension of Gröbner bases concept from polynomial algebras over fields to polynomial rings over rings allows to tackle numerous applications, both of theoretical and of practical importance. Gröbner and Gröbner-Shirshov bases can be defined for various non-commutative and even non-associative algebraic structures. We study the case of associative rings and aim at free algebras over principal ideal rings. We concentrate ourselves on the case of commutative coefficient rings without zero divisors (i.e. a domain). Even working over $\mathbb{Z}$ allows one to do computations, which can be treated as universal for fields of arbitrary characteristic. By using the systematic approach, we revisit the theory and present the algorithms in the implementable form. We show drastic differences in the behavior of Gröbner bases between free algebras and algebras, close to commutative. Even the formation of critical pairs has to be reengineered, together with the criteria for their quick discarding. We present an implementation of algorithms in the Singular subsystem called Letterplace, which internally uses Letterplace techniques (and Letterplace Gröbner bases), due to La Scala and Levandovskyy. Interesting examples accompany our presentation.

CCS CONCEPTS

• **Computing methodologies** → **Special-purpose algebraic systems**; **Algebraic algorithms**; • **Mathematics of computing** → **Mathematical software**.

KEYWORDS

Non-commutative algebra; Gröbner bases; Coefficients in rings; Algorithms; Computer Algebra System

ACM Reference Format:
Viktor Levandovskyy, Tobias Metzlaff, and Karim Abou Zeid. 2020. Computation of Free Non-commutative Gröbner Bases over $\mathbb{Z}$ with Singular:Letterplace. In *International Symposium on Symbolic and Algebraic Computation (ISSAC '20), July 20–23, 2020, Kalamata, Greece.* ACM, New York, NY, USA, 8 pages. https://doi.org/10.1145/3373207.3404052

ISSAC '20, July 20–23, 2020, Kalamata, Greece

ACM ISBN 978-1-4503-7100-1/20/07…$15.00
https://doi.org/10.1145/3373207.3404052

INTRODUCTION

In the recent years a somewhat strange attitude has established itself around Gröbner bases: non-commutative generalizations of various concepts, related to algorithms and, in particular, Gröbner bases, are often met with sceptical expressions like "as expected", "straightforward", "more or less clear" and so on. This is not true in general for generalizations to various flavours of non-commutativity require deep analysis of procedures (algorithms) based on very good knowledge of properties of rings and modules over them. Characteristically, in this paper we demonstrate in e.g. Example 2.4 and 2.5 how *intrinsically different* Gröbner bases over $\mathbb{Z}\langle X\rangle$ are even when compared with Gröbner bases over $\mathbb{Q}\langle X\rangle$, not taking the commutative case into account. An example can illustrate this better than a thousand words: the same set $\{2x, 3y\}$ delivers a finite strong Gröbner basis $\{3x, 3y, yx, xy\}$ over $\mathbb{Z}\langle x, y\rangle$ and an infinite Gröbner basis over $\mathbb{Z}\langle x, y, z_1, \ldots, z_m\rangle$ for any $m \geq 1$, containing e.g. $xz_i^k y, yz_i^k x$ for any natural k.

In his recent articles and in the book [22] Teo Mora has presented "a manual for creating your own Gröbner bases theory" over *effective* associative rings. This development is hard to underestimate, for it presents a unifying theoretical framework for handling very general rings. The theory of non-commutative Gröbner bases was developed by many prominent scientists since the Diamond Lemma of G. Bergman [4]. Especially L. Pritchard [24] proved versions of the PBW Theorem and advanced the theory of bimodules, also over rings. On the other hand, procedures and even algorithms related to Gröbner bases in such frameworks are still very complicated. Therefore, when aiming at implementation, one faces the classical dilemma: generality versus performance. Perhaps the most general implementation which exists is the JAS system by H. Kredel [11]. In our attempts we balance the generality with the performance; based on Singular, we utilize its' long and successful experience with data structures and algorithms in commutative algebra. Notably, the recent years have seen the in-depth development of Gröbner bases in commutative algebras with coefficients in principal ideal rings (O. Wienand, G. Pfister, A. Frühbis-Krüger, A. Popescu, C. Eder, T. Hofmann and others), see e.g. [8–10, 20]. This required massive changes in the structure of algorithms; ideally, one has one code for several instances of Gröbner bases with specialization to individual cases. In particular, the very generation of critical pairs and the criteria for discarding them without much effort were intensively studied. These developments were additional motivation for us in the task of attacking Gröbner bases in free algebras over commutative principal ideal rings, with $\mathbb{Z}$ at the first place. Currently, to the best of our knowledge, no computer algebra system is able to

do such computations. Also, a number of highly interesting applications wait to be solved: in studying representation theory of a finitely presented algebra (i.e. the one, given by generators and relations), computations over $\mathbb{Z}$ remain valid after specification to *any* characteristic and thus encode a universal information. In the system Felix by Apel et al. [2], such computations were experimentally available, though not documented. In his paper [1], Apel demonstrates Gröbner bases of several nontrivial examples over $\mathbb{Z}\langle X\rangle$, the correctness of which we can easily confirm now.

Our secret weapon is the *Letterplace technology* [12–14, 18], which allows the usage of commutative data structures at the lowest level of algorithms. We speak, however, in theory, the language of free algebras over rings, since this is mutually bijective with the language of Letterplace.

This paper is organized as follows: In the first chapter we fix the notations which are necessary when dealing with polynomial rings. Subsequently, in the second chapter we generalize the notion of Gröbner bases for our setup, present a theoretical version of Buchberger's algorithm and give examples to visualize significant differences compared to the field case or the commutative case. Implementation of Buchberger's algorithm depends on and benefits from the choice of pairs, which we will discuss in the third chapter. This is followed up by computational examples and discussion on the implementational aspects.

1 PRELIMINARIES

All rings are assumed to be associative and unital, but not necessarily commutative. We want to discuss non-commutative Gröbner bases over the integers $\mathbb{Z}$. Equivalently one can take any commutative Euclidean domain or principal ideal domain[1] $\mathcal{R}$.

We work towards an implementation and therefore we are interested in *algorithms*, which *terminate* after a finite number of steps. Since $\mathbb{Z}\langle X\rangle$ is not Noetherian, there exist finite generating sets whose Gröbner bases are infinite with respect to any monomial well-ordering. Therefore, our typical computation is executed subject to the *length bound* (where length is meant literally, applied to *words* from the free monoid $\langle X\rangle$), specified in the input, and therefore terminates per assumption. Thus, we talk about *algorithms* in this sense.

Our main goal is to obtain an algorithm to construct a Gröbner basis over such a ring, finding or adjusting criteria for critical pairs and setting up an effective method to implement Buchberger's algorithm in the computer algebra system Singular. The problem of applying the statements of commutative Gröbner basis over Euclidean domains and principal ideal rings, such as in [9, 10, 20, 21], are divisibility conditions of leading monomials.

Let $X = \{x_1, \ldots, x_n\}$ denote the finite alphabet with n letters. We set $\mathcal{P} = \mathcal{R}\langle X\rangle$, the free $\mathcal{R}$-algebra of X, where all words on X form a basis $\mathcal{B} = \langle X\rangle$ of $\mathcal{P}$ as a free $\mathcal{R}$-module (from now on we say shortly "$\mathcal{B}$ is an $\mathcal{R}$-basis"). Moreover, let $\mathcal{P}^e = \mathcal{P} \otimes_{\mathcal{R}} \mathcal{P}^{\mathrm{opp}}$ be the free enveloping $\mathcal{R}$-algebra with basis $\mathcal{B}^e = \{u \otimes v \mid u, v \in \mathcal{B}\}$. The natural action $\mathcal{P}^e \times \mathcal{P} \to \mathcal{P}$, $(u \otimes v, t) \mapsto (u \otimes v)t := utv$ makes a bimodule $\mathcal{P}$ into a left $\mathcal{P}^e$-module. We call the elements of $\mathcal{B}$ *monomials*.

[1] This concept can be extended to principal ideal rings. It was done in [8] for the commutative case with so-called annihilator polynomials.

Let $\preceq$ be a monomial well-ordering on $\mathcal{B}$. With respect to $\preceq$, a polynomial $f \in \mathcal{P} \setminus \{0\}$ has a *leading coefficient* $\mathrm{lc}(f) \in \mathcal{R}$, a *leading monomial* $\mathrm{lm}(f) \in \mathcal{B}$ and a *leading term* $\mathrm{lt}(f) = \mathrm{lc}(f)\,\mathrm{lm}(f) \neq 0$. We denote by $|w|$ the length of the word $w \in \mathcal{B}$. An ordering $\preceq$ is called *length-compatible*, if $u \preceq v$ implies $|u| \leq |v|$. Every subset $\mathcal{G} \subseteq \mathcal{P}$ yields a two-sided ideal, the *ideal of leading terms* $\mathrm{L}(\mathcal{G}) = \langle \mathrm{lt}(f) \mid f \in \mathcal{G} \setminus \{0\}\rangle$.

Naturally, the notions of coefficient, monomial and term carry over to an element $h \in \mathcal{P}^e$ by considering $h \cdot 1 \in \mathcal{P}$.

Definition 1.1. Let $u, v \in \mathcal{B}$. We say, that u and v have an *overlap*, if there exist monomials $t_1, t_2 \in \mathcal{B}$, such that at least one of the four cases

$$(1)\ ut_1 = t_2v \qquad (2)\ t_1u = vt_2 \qquad (3)\ t_1ut_2 = v \qquad (4)\ u = t_1vt_2$$

holds. Additionally, we say, that u and v have a *non-trivial overlap*, if in the first two cases $|t_1| < |v|$ and $|t_2| < |u|$. In the third, respectively fourth case, we say that u *divides* v, respectively v *divides* u. The set of all elements which are divisible by both u and v is denoted by $\mathrm{cm}(u, v)$ (cm: *common multiple*). The set of all minimal, non-trivial elements which are divisible by both u and v is denoted by $\mathrm{lcm}(u, v)$ (lcm: *least ...*), i.e. $t \in \mathrm{lcm}(u, v)$, if and only if there exist $\tau_u, \tau_v \in \mathcal{B}^e$, such that $t = \tau_u u = \tau_v v$, representing non-trivial overlaps of u and v, and if $t, \tilde{t} \in \mathrm{lcm}(u, v)$ with $\tilde{t} = \tau t$ for some $\tau \in \mathcal{B}^e$, then $t = \tilde{t}$ and $\tau = 1 \otimes 1$. If there are only trivial overlaps, then $\mathrm{lcm}(u, v) = \emptyset$. Moreover, if $\mathrm{lm}(g)$ divides $\mathrm{lm}(f)$ for $f, g \in \mathcal{P}$, then $\mathrm{lm}(g) \preceq \mathrm{lm}(f)$ holds.

2 NON-COMMUTATIVE GRÖBNER BASES

A *Gröbner basis* $\mathcal{G} \subseteq \mathcal{P} \setminus \{0\}$ is a generating set for a two-sided ideal $\mathcal{I} \subseteq \mathcal{P}$ with the property $\mathrm{L}(\mathcal{I}) \subseteq \mathrm{L}(\mathcal{G})$. In the field case, this guarantees the existence of a so-called Gröbner representation, which we will recall subsequently, and for any $f \in \mathcal{I} \setminus \{0\}$ the existence of an element $g \in \mathcal{G}$, such that $\mathrm{lt}(g)$ divides $\mathrm{lt}(f)$.

Definition 2.1. Let $f, g \in \mathcal{P} \setminus \{0\}$, $\mathcal{G} \subseteq \mathcal{P} \setminus \{0\}$ be a countable set and $\mathcal{I} \subseteq \mathcal{P}$ be an ideal. Fix a monomial well-ordering $\preceq$.

We say that g *lm-reduces* f, if $\mathrm{lm}(g)$ divides $\mathrm{lm}(f)$ with $\mathrm{lm}(f) = \tau\,\mathrm{lm}(g)$ for some $\tau \in \mathcal{B}^e$ and there are $a, b \in \mathcal{R}$, $a \neq 0$ and $|b| < |\,\mathrm{lc}(f)|$ (in the Euclidean norm), such that $\mathrm{lc}(f) = a\ \mathrm{lc}(g) + b$. Then the *lm-reduction* of f by g is given by $f - a\tau g$.

We say that f has a *strong Gröbner representation* w.r.t. $\mathcal{G}$, if $f = \sum_{i=1}^{m} h_i g_i$ with $m \in \mathbb{N}$, $g_i \in \mathcal{G}$, $h_i \in \mathcal{P}^e$ and there exists a unique $1 \leq j \leq m$, such that $\mathrm{lm}(f) = \mathrm{lm}(h_j g_j)$ and $\mathrm{lm}(f) > \mathrm{lm}(h_i g_i)$ for all $i \neq j$ where $h_i \neq 0$.

$\mathcal{G}$ is called a *strong Gröbner basis* for $\mathcal{I}$, if $\mathcal{G}$ is a Gröbner basis for $\mathcal{I}$ and for all $f' \in \mathcal{I} \setminus \{0\}$ there exists $g' \in \mathcal{G}$, such that $\mathrm{lt}(g')$ divides $\mathrm{lt}(f')$.

Those lm-reductions are the key to obtain a remainder after division through a set $\mathcal{G}$ (usually a generating set) and used in Buchberger's algorithm to construct a Gröbner basis from $\mathcal{G}$. In this sense, the idea of a Gröbner basis is to deliver a unique remainder when dividing through it. Since we operate in a polynomial ring of multiple variables, the expression "reduction" is more justified than "division" to describe a chain of lm-reductions. The outcome of such a reduction, i.e. the remainder of the division, is then known as a *normal form*.

The following strong normal form algorithm uses lm-reductions and can be compared to the normal form algorithms in algebras over fields (cf. [15]).

NormalForm

input: $f \in \mathcal{P} \setminus \{0\}$, $\mathcal{G} \subseteq \mathcal{G}$ finite and partially ordered
output: normal form of f w.r.t. $\mathcal{G}$
01: $h = f$
02: **while** $h \neq 0$ **and** $\mathcal{G}_h = \{g \in \mathcal{G} \mid g \text{ lm-reduces } h\} \neq \emptyset$ **do**
03: choose $g \in \mathcal{G}_h$
04: choose $a,\ b \in \mathcal{R}$ with:
$a \neq 0$, $\mathrm{lc}(h) = a\,\mathrm{lc}(g) + b$ and $|b| < |\mathrm{lc}(h)|$
05: choose $\tau \in \mathcal{B}^e$ with $\mathrm{lm}(h) = \tau\,\mathrm{lm}(g)$
06: $h = h - a\tau g$, the lm-reduction of h by g
07: **end while**
08: **return** h

A normal form of the zero-polynomial is always unique and zero. Termination and correctness are analogous to the classical proof.

The output of the algorithm is in general not unique, but depends on the choice of elements $g \in \mathcal{G}_h$ which are used for reduction.

We confirm, that the proof of the following theorem carries over verbatim from the commutative case in [20].

Theorem 2.2. *Let $\mathcal{G} \subseteq \mathcal{P} \setminus \{0\}$ and $\{0\} \neq \mathcal{I} \subseteq \mathcal{P}$. Then the following statements with respect to $\mathcal{G}$ and $\preceq$ are equivalent:*

(1) $\mathcal{G}$ is a strong Gröbner basis for $\mathcal{I}$.
(2) Every $f \in \mathcal{I} \setminus \{0\}$ has a strong Gröbner representation.
(3) Every $f \in \mathcal{P} \setminus \{0\}$ has a unique normal form after reduction.

An earlier non-commutative version was also proven by Pritchard for non-strong Gröbner bases in [24].

Such a strong Gröbner basis can be computed with Buchberger's algorithm using syzygy relations between leading terms of generating polynomials. In the field case, the computation is done with S-polynomials. However, this does not suffice, when leading coefficients are non-invertible.

Definition 2.3. Let $f,\ g \in \mathcal{P} \setminus \{0\}$. There exist $\tau_f,\ \tau_g \in \mathcal{B}^e$, such that $\tau_f\,\mathrm{lm}(f) = \tau_g\,\mathrm{lm}(g) \in \mathrm{cm}(\mathrm{lm}(f),\ \mathrm{lm}(g))$. Furthermore, let $a = \mathrm{lcm}(\mathrm{lc}(f),\ \mathrm{lc}(f))$ and $a_f,\ a_g \in \mathcal{R}$, such that $a = a_f\,\mathrm{lc}(f) = a_g\,\mathrm{lc}(g)$. In a Euclidean domain, the least common multiple is uniquely determined up to a sign and so are $a_f,\ a_g$. Then an *S-polynomial* of f and g is defined as

$$\mathrm{spoly}(f,\ g) := a_f\tau_f f - a_g\tau_g g.$$

It is known from the commutative case over rings (e.g. [20]), that it does not suffice to take such S-polynomials to obtain a strong Gröbner basis. Let $\mathcal{I} = \langle f = 3x,\ g = 2y\rangle$. Then every S-polynomial of f and g is zero, but clearly $xy = fy - xg \in \mathcal{I}$ has a leading term which is neither divisible by $\mathrm{lt}(f)$ nor $\mathrm{lt}(g)$. Thus, $\{f,\ g\}$ is not a strong Gröbner basis for $\mathcal{I}$. The problematic polynomial xy is constructed by looking at the greatest common divisor of the leading coefficients of f and g.

Let $b = \gcd(\mathrm{lc}(f),\ \mathrm{lc}(f))$ and $b_f,\ b_f \in \mathcal{R}$, such that $b = b_f\,\mathrm{lc}(f) + b_g\,\mathrm{lc}(g)$ (the Bézout identity for the leading coefficients). As above, b is unique in a Euclidean domain as a greatest common divisor, although the Bézout coefficients $b_f,\ b_g$ may not be, but depend on the implementation of a Euclidean algorithm. A *G-polynomial* of f and g is defined as

$$\mathrm{gpoly}(f,\ g) := b_f\tau_f f + b_g\tau_g g.$$

So far everything seems to work out as in the commutative case. We consider some examples to see, that this assumption is wrong.

Example 2.4. Let $f = 2xy,\ g = 3yz \in \mathbb{Z}\langle x,\ y,\ z\rangle$. Usually we would compute an S-polynomial $3fz - 2xg = 0$ and a G-polynomial

$$\mathrm{gpoly}(f,\ g) := (-1)\cdot 2xy\cdot z + 1\cdot x\cdot 3yz = xyz$$

and add them to $\{f,\ g\}$ to obtain a strong Gröbner basis for $\mathcal{I} = \langle f,\ g\rangle \subseteq \mathcal{P}$. But clearly

$$\mathrm{gpoly}'(f,\ g) := (-1)\cdot 2xy\cdot w\cdot yz + 1\cdot xy\cdot w\cdot 3yz = xywyz$$

is also a G-polynomial of $f,\ g$ for every $w \in \mathcal{B}$ and must be added to the basis. In other words there is no finite Gröbner basis for $\mathcal{I}$ and we have to be satisfied with computing up to a fixed maximal leading monomial or word length. Note that in the case of gpoly we computed a G-polynomial in the canonical way by looking for a non-trivial overlap of xy and yz. In the case of gpoly$'$ we ignored this overlap. In the commutative case this is irrelevant, because $\mathrm{gpoly}(f,\ g)$ divides $\mathrm{gpoly}'(f,\ g)$. Furthermore, in the field case this is also irrelevant, because we do not need G-polynomials.

Example 2.5. A similar problem occurs with S-polynomials. Let $f = 2xy + x,\ g = 3yz + z$. Then $\mathrm{spoly}(f,\ g) = 3fz - 2xg = xz$ is an S-polynomial of f and g. However, so are all polynomials

$$\mathrm{spoly}'(f,\ g) := 3fwyz - 2xywg = 3xwyz - 2xywz$$

for any monomial $w \in \mathcal{B}$. Now we can reduce $\mathrm{spoly}'(f,\ g)$ to

$$(\mathrm{spoly}'(f,\ g) - xwg) + fwz = -2xywz + fwz = xwz$$

which is not reducible any further. Therefore, we have to add $\mathrm{spoly}'(f,\ g)$ to the basis. And even this is not enough. For $f = 2xy + x$ we see that

$$\mathrm{spoly}''(f,\ f) := fwxy - xywf = xwxy - xywx \neq 0$$

is an S-polynomial of f with itself which does not reduce any further and we need $\mathrm{lm}(f)w\,\mathrm{lm}(f) \in \mathrm{cm}(\mathrm{lm}(f),\ \mathrm{lm}(f))$, although it is clearly not contained in $\mathrm{lcm}(\mathrm{lm}(f),\ \mathrm{lm}(f))$. So even principal ideals do not have finite strong Gröbner bases in general! Such behavior of S-polynomials does not occur for non-commutative polynomials over fields.

Also, note that we do not consider any further extensions of the leading monomials, meaning that the S- and G-polynomial corresponding to $t \in \mathrm{lcm}(\mathrm{lm}(f),\ \mathrm{lm}(g))$ or $\mathrm{lm}(f)w\,\mathrm{lm}(g)$ make any further (trivial) overlap relations τt or $\tau(\mathrm{lm}(f)w\,\mathrm{lm}(g))$ for $\tau \in \mathcal{B}^e$ redundant. Therefore, in the definition of $\mathrm{lcm}(x,\ y)$ we stress the importance of the minimality.

The previous example shows that we have to consider all possible S- and G-polynomials, but those are infinitely many. Moreover, the set $\mathrm{cm}(\mathrm{lm}(f),\ \mathrm{lm}(g))$ contains too many elements that are redundant whereas the set $\mathrm{lcm}(\mathrm{lm}(f),\ \mathrm{lm}(g))$ is too small. The following definition is made to classify two types of S- and G-polynomials, namely those corresponding to non-trivial overlap relations and those corresponding to trivial ones.

Definition 2.6. Let $f, g \in \mathcal{P} \setminus \{0\}$ and $a_f, a_g, b_f, b_g \in \mathcal{R}$ as in 2.3. We distinguish between the following two cases.

If $\mathrm{lm}(f)$ and $\mathrm{lm}(g)$ have a non-trivial overlap, then there exist $t \in \mathrm{LCM}(\mathrm{lm}(f), \mathrm{lm}(g))$ and $\tau_f, \tau_g \in \mathcal{B}^e$, such that $t = \tau_f \mathrm{lm}(f) = \tau_g \mathrm{lm}(g)$. Furthermore, we assume that $\tau_f = 1 \otimes t_f$, $\tau_g = t_g \otimes 1$ or $\tau_f = 1 \otimes 1$, $\tau_g = t_g \otimes t'_g$ for $t_f, t_g, t'_g \in \mathcal{B}$ with $|t_f| < |\mathrm{lm}(g)|$, $|t_g|, |t'_g| < |\mathrm{lm}(f)|$. We define a *first type S-polynomial* of f and g w.r.t. t as

$$\mathrm{spoly}_1^t(f, g) := a_f \tau_f f - a_g \tau_g g$$

and a *first type G-polynomial* of f and g w.r.t. t as

$$\mathrm{gpoly}_1^t(f, g) := b_f \tau_f f + b_g \tau_g g.$$

If such τ_f, τ_g do not exist then we set the first type S- and G-polynomials both to zero. Since two monomials may have several non-trivial overlaps, these τ_f, τ_g are not unique. More precisely, this results from $\mathcal{P}$ not being a unique (but merely a finite) factorization domain.

For any $w \in \mathcal{B}$ we define the *second type S-polynomial* of f and g w.r.t. w by

$$\mathrm{spoly}_2^w(f, g) := a_f f w \,\mathrm{lm}(g) - a_g \,\mathrm{lm}(f) w g$$

and the *second type G-polynomial* of f and g w.r.t. w as

$$\mathrm{gpoly}_2^w(f, g) := b_f f w \,\mathrm{lm}(g) + b_g \,\mathrm{lm}(f) w g.$$

Remark 2.7. Clearly, it only makes sense to consider first type S- and G-polynomials if there is a non-trivial overlap of the leading monomials. However, as Example 2.4 shows, we always need to consider second type S- and G-polynomials. For any $w \in \mathcal{B}$ we have $\mathrm{lm}(f) w \,\mathrm{lm}(g) \in \mathrm{CM}(\mathrm{lm}(f), \mathrm{lm}(g))$ and $\mathrm{lm}(g) w \,\mathrm{lm}(f) \in \mathrm{CM}(\mathrm{lm}(f), \mathrm{lm}(g))$, which are distinct in general. Therefore, we need to consider both $\mathrm{spoly}_2^w(f, g)$ and $\mathrm{spoly}_2^w(g, f)$ and the same holds for second type G-polynomials. Also, note that the set of first type S- and G-polynomials is finite, because our monomial ordering is a well-ordering, whereas the set of second type S- and G-polynomials is infinite. Therefore, we need to fix an upper bound for the length of monomials which may be involved.

It is important to point out, that the elements τ_f, τ_g are not uniquely determined. Take for example $f = 2xyx + y$, $g = 3x + 1$. Then $t := xyx = \mathrm{lm}(f) = xy \,\mathrm{lm}(g) \in \mathrm{LCM}(\mathrm{lm}(f), \mathrm{lm}(f))$, but also $t = \mathrm{lm}(g)yx$ and thus $\mathrm{spoly}_1^t(f, g) = -3f + 2gyx = 2yx - 3y$ and $(\mathrm{spoly}_1^t)'(f, g) = -3f + 2xyg = 2xy - 3y$ are both first type S-polynomials with different leading monomials.

A finite set $\mathcal{G} \subseteq \mathcal{P}$ is called *length-bounded strong Gröbner basis* for an ideal $\mathcal{I}$, if there is a Gröbner basis $\mathcal{G}'$ for $\mathcal{I}$, such that $\mathcal{G} \subseteq \mathcal{G}'$ contains precisely the elements of $\mathcal{G}'$ of length smaller or equal to d for some $d \in \mathbb{N}$.

The following algorithm uses Buchberger's criterion 2.8 as a characterization for strong Gröbner bases, which we will prove subsequently. It computes S- and G-polynomials up to a fixed degree and reduces them with the algorithm NormalForm in order to obtain a length-bounded strong Gröbner basis for an input ideal.

BuchbergerAlgorithm

input: $\mathcal{I} = \langle f_1, \ldots, f_k \rangle \subseteq \mathcal{R}\langle X \rangle$, $d \in \mathbb{N}$, NormalForm
output: length-bounded strong Gröbner basis $\mathcal{G}$ for $\mathcal{I}$

$\mathcal{G} = \{f_1, \ldots, f_k\}$
$\mathcal{L} = \{\mathrm{spoly}_1^t(f_i, f_j), \mathrm{gpoly}_1^t(f_i, f_j) \mid \forall\, t\ ^*,\ i,\ j\}$
$\mathcal{L} = \mathcal{L} \cup \{\mathrm{spoly}_2^w(f_i, f_j), \mathrm{gpoly}_2^w(f_i, f_j) \mid \forall\, w\ ^{**},\ i,\ j\}$
while $\mathcal{L} \neq \emptyset$ **do**
  choose $h \in \mathcal{L}$
  $\mathcal{L} = \mathcal{L} \setminus \{h\}$
  $h = \mathrm{NormalForm}(h, \mathcal{G})$
  if $h \neq 0$ **then**
    $\mathcal{G} = \mathcal{G} \cup \{h\}$
    for $g \in \mathcal{G}$ **do**
      $\mathcal{L} = \mathcal{L} \cup \{\mathrm{spoly}_1^t(g, h), \mathrm{gpoly}_1^t(g, h) \mid \forall\, t\ ^*\}$
      $\mathcal{L} = \mathcal{L} \cup \{\mathrm{spoly}_1^t(h, g), \mathrm{gpoly}_1^t(h, g) \mid \forall\, t\ ^*\}$
      $\mathcal{L} = \mathcal{L} \cup \{\mathrm{spoly}_2^w(g, h), \mathrm{gpoly}_2^w(g, h) \mid \forall\, w\ ^{***}\}$
      $\mathcal{L} = \mathcal{L} \cup \{\mathrm{spoly}_2^w(h, g), \mathrm{gpoly}_2^w(h, g) \mid \forall\, w\ ^{***}\}$
    end do
  end if
end while
return $\mathcal{G}$

* $t \in \mathrm{LCM}$, such that $|t| < d$
** $w \in \mathcal{B}$, such that $|\mathrm{lm}(f_i)| + |w| + |\mathrm{lm}(f_j)| < d$
*** $w \in \mathcal{B}$, such that $|\mathrm{lm}(h)| + |w| + |\mathrm{lm}(g)| < d$

For the algorithm to terminate we need the set $\mathcal{L}$ to eventually become empty. This happens, if and only if after finitely many steps every S- and G-polynomial based on any combination of leading terms has normal form zero w.r.t $\mathcal{G}$, i.e. there exists a chain of lm-reductions, such that the current S- or G-polynomial reduces to zero. However, lm-reductions only use polynomials of smaller or equal length and all of these are being computed. Therefore, the algorithm terminates.

For the correctness of the algorithm we still need a version of Buchberger's criterion. More precisely, we want $\mathcal{G}$ to be a Gröbner basis for $\mathcal{I}$, if and only if for every pair $f, g \in \mathcal{G}$ all their S- and G-polynomials reduce to zero. Moreover, we only want to consider first and second type S- and G-polynomials, i.e. only use $t \in \mathrm{CM}(\mathrm{lm}(f), \mathrm{lm}(g))$, such that one of the following four cases

$$(1)\ t = \mathrm{lm}(f)t'_f = t_g \,\mathrm{lm}(g) \qquad (2)\ t = \mathrm{lm}(f) = t_g \,\mathrm{lm}(g)t'_g$$
$$(3)\ t = t_f \,\mathrm{lm}(f) = \mathrm{lm}(g)t'_g \qquad (4)\ t = t_f \,\mathrm{lm}(f)t'_f = \mathrm{lm}(g)$$

holds for $t_f, t'_f, t_g, t'_g \in \mathcal{B}$. This excludes all cases where t is not minimal, i.e. $t = \tau t'$ for $\tau \in \mathcal{B}^e$ and t' satisfying one of the above four cases. Pritchard has proven in [24], that for a generating set of the left syzygy module (which is not finitely generated in general) we may use only minimal syzygies.

Lemma 2.8. *Let $\mathcal{G} \subseteq \mathcal{P} \setminus \{0\}$. Then $\mathcal{G}$ is a strong Gröbner basis for $\mathcal{I} := \langle \mathcal{G} \rangle$, if and only if for every pair $f, g \in \mathcal{G}$ their first and second type S- and G-polynomials reduce to zero w.r.t. $\mathcal{G}$.*

Proof. The idea of the proof goes back to [20]; we only need to show the "if" part. Let $f \in \mathcal{I} \setminus \{0\}$ with $f = \sum_i h_i g_i$ for some $h_i \in \mathcal{P}^e$. We set $t := \max(\mathrm{lm}(h_i g_i))$ and $M := \{i \in \mathbb{N} \mid \mathrm{lm}(h_i g_i) =$

$t\}$. Clearly $\mathrm{lm}(f) \preceq t$ and we may assume that there is no other representation of f where t is smaller. Without loss of generality let $M = \{1, \ldots, m\}$. Showing, that M contains exactly one element, proves the lemma and can be done by contradiction as follows. We omit the technical details. The setup allows to choose a representation of f, where the coefficient sum $\sum_{i=1}^{m} |\mathrm{lc}(h_i)\,\mathrm{lc}(g_i)|$ is minimal for fixed t. If M contains more than one element, one can consider the first two polynomials in $\mathcal{G}$ (those shall be g_1 and g_2), that occur in the representation of f with $T = \tau_1\,\mathrm{lm}(g_1) = \tau_2\,\mathrm{lm}(g_2)$. Here, $\tau_1, \tau_2 \in \mathcal{B}^e$ and T results from one the above four cases (1), ..., (4). By analyzing $\mathrm{spoly}(g_1, g_2)$ and $\mathrm{gpoly}(g_1, g_2)$ and using the intrinsic properties of the Euclidean domain $\mathcal{R}$, we obtain a representation $h_1g_1 + h_2g_2 = \sum_j h'_j g_j$, which has a smaller coefficient sum than our original representation. This is in contradiction to the choice made. □

It is possible to define monic[2] and reduced Gröbner bases [19, 23] in our setup. Let $\mathcal{G} \subseteq \mathcal{P} \setminus \{0\}$. It is called a *reduced Gröbner basis*, if

(1) every $g \in \mathcal{G}$ has leading coefficient with signum 1,
(2) $\mathrm{L}(\mathcal{G} \setminus \{g\}) \subsetneq \mathrm{L}(\mathcal{G})$ for every $g \in \mathcal{G}$, and
(3) $\mathrm{lt}(\mathrm{tail}(g)) \notin \mathrm{L}(\mathcal{G})$ for every $g \in \mathcal{G}$.

The first condition states that, in the case of $\mathcal{R} = \mathbb{Z}$, every element of a reduced Gröbner basis has leading coefficient in $\mathbb{Z}_+$. The second condition is sometimes referred to as "simplicity" and means that the leading ideal becomes strictly smaller when removing an element, thus no element is useless. The third condition, "tail-reduced", is required in the classical field case with commutative polynomials to ensure that a reduced Gröbner basis is unique. However, this does not suffice in our setup: for instance, Pritchard gave a counterexample in [24].

Let $f = 2y^2$, $g = 3x^2 + y^2$ and $\mathcal{I} = \langle f, g\rangle$. Then $\{f, g\}$ is a Gröbner basis for $\mathcal{I}$ with respect to any ordering $x \succ y$ and satisfies the above three conditions. On the other hand, this is also true for $\{f, g'\}$ where $g' = g - f = 3x^2 - y^2$, so we have two different reduced Gröbner bases for $\mathcal{I}$. In the field case the polynomial g is not tail-reduced. This example can be used in both the commutative and non-commutative case.

When implementing a version of Buchberger's algorithm, one should always aim to have a reduced Gröbner basis as an output. In fact this is more practical, because removing elements, which are not simplified or tail reduced speeds up the computation, since we do not need to consider them in critical pairs.

Lemma 2.9. *Suppose, that $\mathcal{G} \subset R\langle X\rangle$ is a result of a Gröbner basis computation up to a length bound $d \in \mathbb{N}$, and thus finite. $\mathcal{G}$ is a strong Gröbner basis of the ideal it generates, if and only if a Gröbner basis computation up to a length bound $2d-1$ does not change $L(\mathcal{G})$.*

Proof. It suffices to prove the "if" part. Assume that $\mathcal{G}'$ is a result of a computation up to degree $2d-1$ and $\mathrm{L}(\mathcal{G}) = \mathrm{L}(\mathcal{G}')$. This means that all overlap relations of length $2d-1$, which are precisely the non-trivial overlap relations for polynomials of degree up to d, do not enlarge the leading ideal. In other words, all first kind S- and G-polynomials reduce to zero. Because $\mathcal{G}$ is finite and since for a Gröbner basis over fields or respectively for a "weak" (not strong) Gröbner basis over rings, we only need non-trivial overlap relations, this is the characterizing property of a Gröbner basis. □

If we additionally assume that a Gröbner basis computation up to degree $2d$ does not change $\mathrm{L}(\mathcal{G})$, then this means that the trivial overlap relations $\mathrm{lm}(f)\,\mathrm{lm}(g)$, which are of length $\leq 2d$, do not add new polynomials to the basis. It remains to prove that this suffices for all trivial overlap relations $\mathrm{lm}(f)w\,\mathrm{lm}(g)$ with $w \in \mathcal{B}$ to be irrelevant. Moreover, we need to take the divisibility condition $\mathrm{lt}(g) \mid \mathrm{lt}(f)$ into account. As a consequence we could replace "Gröbner basis" with "strong Gröbner basis" in Lemma 2.9.

3 CRITICAL PAIRS

To improve the procedure BuchbergerAlgorithm, we need criteria to determine which pairs of polynomials of the input set yield S- and G-polynomials, which reduce to zero. In the following we will recall the criteria for discarding critical pairs known from the commutative case and analyze, which of them can be applied in the case $\mathcal{R}\langle X\rangle$.

Remark 3.1. First we consider the case where $t := \mathrm{lm}(f)$ is divisible by (or even equals to) $\mathrm{lm}(g)$. Then $\mathrm{lcm}(\mathrm{lm}(f), \mathrm{lm}(g))$ contains exactly one element, namely t, because it is the only minimal element that is divisible by both leading monomials. Therefore, $\mathrm{spoly}_1^t(f, g)$ and $\mathrm{gpoly}_1^t(f, g)$ are the only first type S- and G-polynomials. However, these are not uniquely determined, we might have more overlap relations of $\mathrm{lm}(f), \mathrm{lm}(g)$, as we have seen in the previous example of Remark 2.7, and we still need second type S-polynomials.

The following Lemma has the obvious consequence that G-polynomials are redundant over fields.

Lemma 3.2. *(cf. [10, 20]) Let $f, g \in \mathcal{P} \setminus \{0\}$. If $\mathrm{lc}(f) \mid \mathrm{lc}(g)$ in $\mathcal{R}$, then every G-polynomial of f and g is redundant.*

Proof. By the hypothesis we have $b = \mathrm{lcm}(\mathrm{lc}(f), \mathrm{lc}(g)) = \mathrm{lc}(f)$. Let $r \in \mathcal{R}$, such that $r\,\mathrm{lc}(f) = \mathrm{lc}(g)$. Then $\mathrm{lc}(f) = (nr+1)\,\mathrm{lc}(f) - n\,\mathrm{lc}(g)$ yields any possible Bézout identity for b, where $n \in \mathbb{Z}$. Thus, with $t = \tau_f\,\mathrm{lm}(f) = \tau_g\,\mathrm{lm}(g)$, every G-polynomial of f and g has shape $\mathrm{gpoly}(f, g) = (nr+1)\tau_f f - n\tau_g g = \mathrm{lc}(f)t + n(r\tau_f\,\mathrm{tail}(f) - \tau_g\,\mathrm{tail}(g)) + \tau_f\,\mathrm{tail}(f)$. Subtracting $\tau_f f$, we can reduce this to $n(r\tau_f\,\mathrm{tail}(f) - \tau_g\,\mathrm{tail}(g))$. Note that $r\tau_f\,\mathrm{tail}(f) - \tau_g\,\mathrm{tail}(g)$ is an S-polynomial of f and g. Hence, every G-polynomial of f and g reduces to zero, after we compute their S-polynomials. □

For $f \in \mathcal{P}\setminus\{0\}$ we define recursively $\mathrm{tail}^0(f) := f$ and $\mathrm{tail}^i(f) := \mathrm{tail}(\mathrm{tail}^{i-1}(f))$ for $i \geq 1$ when $\mathrm{tail}^{i-1}(f) \neq 0$.

Lemma 3.3. *(Buchberger's product criterion, cf. [10, 20]) Let $f, g \in \mathcal{P} \setminus \{0\}$ and $w \in \mathcal{B}$, such that*

(1) $\mathrm{lc}(f)$ and $\mathrm{lc}(g)$ are coprime over $\mathcal{R}$,
(2) $\mathrm{lm}(f)$ and $\mathrm{lm}(g)$ only have trivial overlaps and
(3) for all $i,\ j \geq 1$, w does not satisfy:
$$\mathrm{lm}(\mathrm{tail}^i(f))w\,\mathrm{lm}(g) = \mathrm{lm}(f)w\,\mathrm{lm}(\mathrm{tail}^j(g))\ .$$

Then $s := \mathrm{spoly}_2^w(f, g)$ reduces to zero w.r.t. $\{f, g\}$.

Proof. Under the assumptions (1) and (2) we have $s = fw\,\mathrm{lt}(g) - \mathrm{lt}(f)wg = fw(g - \mathrm{tail}(g)) - (f - \mathrm{tail}(f))wg = \mathrm{tail}(f)wg - fw\,\mathrm{tail}(g)$.

[2] An element is *monic* if it's leading coefficient is 1

Note that $\mathrm{tail}(f)wg$ reduces to zero w.r.t. g and $fw\,\mathrm{tail}(g)$ reduces to zero w.r.t. f.

By (3) we can assume without loss of generality that $\mathrm{lt}(s) = \mathrm{lt}(\mathrm{tail}(f))w\,\mathrm{lt}(g)$. Then s reduces to $s' := s - \mathrm{lt}(\mathrm{tail}(f))wg$ and $\mathrm{lm}(s') \prec \mathrm{lm}(s)$. Again by (3) there is no cancellation of leading terms and, since $\prec$ is a well ordering, we iteratively see that s reduces to zero. □

Remark 3.4. The commutative version of Buchberger's product (cf. [10, 20]) criterion states, that the S-polynomial reduces to zero, if the leading terms are coprime over $K[X]$.

Condition (3), or rather its negation, describes a very specific relation between the terms of f and g. There is only a finite amount of $w \in \mathcal{B}$, that satisfy such relation and are at the same time considered in BuchbergerAlgorithm, because we only compute up to a certain length.

The version over fields for this criterion is much simpler, because then we only consider w to be the empty word which clearly satisfies (3). Moreover, (1) is redundant and Buchberger's product criterion states that an S-polynomial reduces to zero when the leading monomials have only trivial overlap relations.

We consider further situation where we might find applications for criteria.

Example 3.5. If $\mathrm{lm}(f)$ and $\mathrm{lm}(g)$ do not overlap and the leading coefficients are not coprime, i.e. $\mathrm{lcm}(\mathrm{lc}(f), \mathrm{lc}(g)) \neq 1$, then we can make no *a priori* statement about reduction. This only applies to second type S- and G-polynomials. Take for example $f = 4xy + x$, $g = 6zy+z \in \mathbb{Z}\langle X\rangle = \mathbb{Z}\langle x, y, z\rangle$ in the degree left lexicographical ordering with $x \succ y \succ z$. Then $\mathrm{spoly}_2^1(f, g) = 3fzy - 2xyg = 3xzy-2xyz$ and $\mathrm{gpoly}_2^1(f, g) = (-1)fzy+1xyg = 2xyzy+xyz-xzy$ both do not reduce any further and thus must be added to the Gröbner basis just as any other second type S- and G-polynomial.

Also, for first type S- and G-polynomials no statement can be made when the leading coefficients are not coprime. For example in the case of $f = 4xy+y$, $g = 6yz+y$ we have $\mathrm{spoly}_1^{xyz}(f, g) = 3fz-2xg = 3yz-2xy$ and $\mathrm{gpoly}_1^{xyz}(f, g) = (-1)fz+1xg = 2xyz-yz+xy$ which do not reduce any further.

Remark 3.6. Recall that the pair $\{f, g\}$ can be replaced in the commutative case (cf. [10]) by $\{\mathrm{spoly}(f, g), \mathrm{gpoly}(f, g)\}$, if $t = \mathrm{lm}(f) = \mathrm{lm}(g)$ (cf. [10]). Now, if $\mathrm{lm}(f) = \mathrm{lm}(g)$ then in the definition of first type S- and G-polynomials we have $\tau_f = \tau_g = 1 \otimes 1$ and therefore $\mathrm{spoly}_1^t(f, g) = a_f f - a_g g$ and $\mathrm{gpoly}_1^t(f, g) = b_f f + b_g g$. This yields a linear equation

$$\begin{pmatrix}\mathrm{spoly}_1^t(f, g)\\ \mathrm{gpoly}_1^t(f, g)\end{pmatrix} = \begin{pmatrix}a_f & -a_g\\ b_f & b_g\end{pmatrix}\begin{pmatrix}f\\ g\end{pmatrix},$$

where the defining matrix has determinant $a_f b_g + a_g b_f = 1$, and thus is invertible over $\mathcal{R}$. Hence, we can obtain f and g from their S- and G- polynomial and replace them. The importance of this statement was discussed for the commutative case in [10] and translates equivalently to the non-commutative one.

The following two lemmata are chain criteria, which are based on the idea to have two critical pairs and derive a third one from them under certain conditions. The commutative versions for both criteria were proven in [10].

Lemma 3.7. *(Buchberger's S-chain criterion, cf. [10, 20]) Let $\mathcal{G} \subseteq \mathcal{P}\setminus\{0\}$ and $f, g, h \in \mathcal{G}$. For $a, b \in \{f, g, h\}$ let $\mathrm{LCM}(\mathrm{lm}(a), \mathrm{lm}(b)) \neq \emptyset$ and fix $T_{ab} \in \mathrm{LCM}(\mathrm{lm}(a), \mathrm{lm}(b))$ and choose $\tau_{ab} \in \mathcal{B}^e$ with $\tau_{ab}\,\mathrm{lm}(a) = T_{ab}$. There exist $\tau_{ba} \in \mathcal{B}^e$, such that $\tau_{ba}\,\mathrm{lm}(b) = T_{ab}$. We assume that $T_{ab} = T_{ba}$. Furthermore, let*

(1) $T_{hg} = T_{gh}$ be divisible by both T_{hf} and T_{gf} with $\delta_{gf}T_{hf} = T_{hg}$ and $\delta_{hf}T_{gf} = T_{gh}$ for some δ_{gf}, $\delta_{hf} \in \mathcal{B}^e$,
(2) $\mathrm{lc}(f) \mid \mathrm{lcm}(\mathrm{lc}(g), \mathrm{lc}(h))$ over $\mathcal{R}$ and
(3) $\mathrm{spoly}_1^{T_{fg}}(f, g)$ and $\mathrm{spoly}_1^{T_{fh}}(f, h)$ both have strong Gröbner representations w.r.t. $\mathcal{G}$.

Then $\mathrm{spoly}_1^{T_{gh}}(f, g)$ has a strong Gröbner representation w.r.t. $\mathcal{G}$.

Proof. Let $c_{ab} := \dfrac{\mathrm{lcm}(\mathrm{lc}(a), \mathrm{lc}(b))}{\mathrm{lc}(a)}$ for $a, b \in \{f, g, h\}$. Then one can check by hand, that

$$\frac{c_{hg}}{c_{hf}}\delta_{gf}\,\mathrm{spoly}_1^{T_{fh}}(f, h) - \frac{c_{gh}}{c_{gf}}\delta_{hf}\,\mathrm{spoly}_1^{T_{fg}}(f, g)$$
$$= c_{gh}\delta_{hf}\tau_{gf}g - c_{hg}\delta_{gf}\tau_{hf}h + \left(\frac{c_{hg}c_{fh}}{c_{hf}}\delta_{gf}\tau_{fh} - \frac{c_{gh}c_{fg}}{c_{gf}}\delta_{hf}\tau_{fg}\right)f.$$

Using the relations for monomial expressions τ_{ab}, T_{ab}, δ_{ab} and coeffiecients c_{ab}, we see that the first term equals $\mathrm{spoly}_1^{T_{gh}}(g, h)$ and we obtain

$$\mathrm{spoly}_1^{T_{gh}}(g, h) = \frac{c_{hg}}{c_{hf}}\delta_{gf}\,\mathrm{spoly}_1^{T_{fh}}(f, h) - \frac{c_{gh}}{c_{gf}}\delta_{hf}\,\mathrm{spoly}_1^{T_{fg}}(f, g),$$

which shows that $\mathrm{spoly}_1^{T_{gh}}(g, h)$ has a strong Gröbner representation w.r.t. $\mathcal{G}$. This works analogously for second type S-polynomials $\mathrm{spoly}_2^{w}(g, h)$ or $\mathrm{spoly}_2^{\tilde{w}}(h, g)$, if we choose w or $\tilde{w}$, such that either $\mathrm{lm}(g)w\,\mathrm{lm}(h) = T_{hg}$ or $\mathrm{lm}(h)\tilde{w}\,\mathrm{lm}(g) = T_{hg}$. □

We give a similar criterion for G-polynomials, which can be proven in a manner similar to 3.7.

Lemma 3.8. *(Buchberger's G-chain criterion, cf. [10, 20]) Let $\mathcal{G} \subseteq \mathcal{P} \setminus \{0\}$ and $f, g, h \in \mathcal{G}$. We retain the notations T_{ab} and τ_{ab} from the above. Let*

(1) $T_{hg} = T_{gh}$ be divisible by both T_{hf} and T_{gf} with $\delta_{gf}T_{hf} = T_{hg}$ and $\delta_{hf}T_{gf} = T_{gh}$ for some δ_{gf}, $\delta_{hf} \in \mathcal{B}^e$ and
(2) $\mathrm{lc}(f) \mid gcd(\mathrm{lc}(g), \mathrm{lc}(g))$ with $d := \dfrac{gcd(\mathrm{lc}(g), \mathrm{lc}(g))}{\mathrm{lc}(f)}$.

Then $\mathrm{gpoly}_1^{T_{gh}}(g, h)$ has a strong Gröbner representation w.r.t. $\mathcal{G}$.

We conclude that the well-known criteria for S- and G-polynomials from the commutative case can also be applied in the non-commutative case with modifications, if we distinguish between first and second type S- and G-polynomials. Computations can show how hard these requirements are to be fulfilled compared to the commutative case by specifically counting the number of applications of product and chain criteria.

4 EXAMPLES

We give examples for Gröbner bases that have been computed up to a certain length bound over the integers. These examples also show that although computing over $\mathbb{Z}$ delivers infinite results much more often than when computing over fields, non-commutative Gröbner bases over $\mathbb{Z}$ can be finite as well.

For the examples in this Section, which we take from [1], let $\mathcal{P} = \mathbb{Z}\langle x, y, z\rangle$ with the degree left lexicographical ordering and $x \succ y \succ z$ (if not indicated otherwise).

Example 4.1. We consider the ideal $\mathcal{I} = \langle f_1 = yx - 3xy - 3z,\ f_2 = zx - 2xz + y,\ f_3 = zy - yz - x\rangle \subset \mathcal{P}$. We investigated it over $\mathbb{Q}\langle x, y, z\rangle$ in [17] (in the same issue of the proceedings), where we also comment in details on syntax and commands of Singular:Letterplace.

```
LIB "freegb.lib"; //initialization of free algebras
ring r = integer,(z,y,x),Dp; //degree left lex ord z>y>x
ring R = freeAlgebra(r,7); // length bound is 7
ideal I = y*x - 3*x*y - 3*z, z*x - 2*x*z +y, z*y-y*z-x;
option(redSB); // Groebner basis will be minimal
option(redTail); // Groebner basis will be tail-reduced
ideal J = twostd(I); // compute a two-sided GB of I
J; // print generators of J
```

The output has plenty of elements in each degree (which is the same as length because of the degree ordering), what hints at potentially infinite Gröbner basis (what we confirm below) and the elements, which can be subsequently constructed, are

$$\{f_1,\ f_2,\ f_3,\ 12xy + 9z,\ 9xz - 3y,\ 6y^2 - 9x^2,\ 6yz + 3x,$$
$$3z^2 + 2y^2 - 5x^2,\ 6x^3 - 3yz,\ 4x^2y + 3xz,\ 3x^2z + 3xy + 3z,$$
$$2xy^2 + 3x^3 + 3yz + 3x,\ 3xyz + 3y^2 - 3x^2,\ 2y^3 + x^2y + 3xz,$$
$$2x^4 + y^2 - x^2,\ 2x^3y + 3y^2z + 3xy + 3z,\ x^2yz + xy^2 - x^3,$$
$$xy^2z - y^3 + x^2y,\ x^5 - y^3z - xy^2 + x^3,\ y^3z^2 - x^4y,$$
$$x^4z + x^3y + 2y^2z + x^2z + 3xy + 3z,\ xy^3z - y^4 + x^4 - y^2 + x^2,$$
$$xy^4z - y^5 + x^2y^3,\ xy^5z - y^6 + x^4y^2 + y^4 + x^4 + 2y^2 - 2x^2\}.$$

Indeed, we can show that $\forall i \geq 2$ I contains an element with the leading monomial xy^iz. Therefore this Gröbner basis is infinite, but can be presented in finite terms. Note, that the original generators have been preserved in a Gröbner basis, while over $\mathbb{Q}$ (see [17]) they were decomposed. Also, over $\mathbb{Q}$ the input ideal has a finite Gröbner basis of degree at most 3.

Example 4.2. Let $\mathcal{I} = \langle f_1 = yx - 3xy - z,\ f_2 = zx - xz + y,\ f_3 = zy - yz - x\rangle \subset \mathcal{P}$. Then $\mathcal{I}$ has a finite strong Gröbner basis, namely

$$\{f_1,\ f_2,\ f_3,\ 8xy + 2z,\ 4xz - 2y,\ 4yz + 2x,$$
$$2x^2 - 2y^2,\ 4y^2 - 2z^2,\ 2z^3 - 2xy\}.$$

As we can see, the leading coefficients of the Gröbner basis above might vanish, if we pass to the field of characteristic 2. Therefore the bimodule $M := \mathbb{Z}\langle x, y, z\rangle/I$ might have nontrivial 2-torsion, i.e. there is a nonzero submodule $T_2(M) := \{p \in M : \exists n \in \mathbb{N}_0\ 2^n \cdot p \in I\}$. By adopting the classical method of Caboara and Traverso for computing colon (or quotient) ideals to our situation, where we use the fact that the ground ring is central (i.e. commutes with all variables), we do the following:

```
LIB "freegb.lib"; //we will use position-over-term order
ring r = integer,(x,y,z),(c,dp);
ring R = freeAlgebra(r,7,2); // 2==number of components
ideal I = y*x - 3*x*y - z, z*x - x*z +y, z*y-y*z-x;
option(redSB); option(redTail);
ideal J = twostd(I);    module N;
N = 2*ncgen(1)*gen(1)+ncgen(2)*gen(2),J*ncgen(1)*gen(1);
module SN = twostd(N);   SN;
```

Above, `gen(i)` stands for the i-th canonical basis vector (commuting with everything) and `ncgen(i)` - for the i-th canonical generator of the free bimodule, which commutes only with constants. The output, which is a list of vectors, looks as follows:

```
...
SN[9]=[0,z*z*z*ncgen(2)-x*y*ncgen(2)]
SN[10]=[2*ncgen(1),ncgen(2)]
SN[11]=[z*y*ncgen(1)-y*z*ncgen(1)-x*ncgen(1)]
...
```

From this output we gather all vectors with 0 in the first component `ncgen(1)`, which results into an ideal, whose Gröbner basis is

$$\{zy - yz - x,\ zx - xz + y,\ yx + xy,\ 2yz + x,$$
$$2xz - y,\ 2y^2 - z^2,\ 4xy + z,\ x^2 - y^2,\ z^3 - xy\}.$$

Another colon computation does not change this ideal, therefore it is the saturation ideal of I at 2, denoted by $L = I : 2^\infty \subset \mathbb{Z}\langle x, y, z\rangle$. It is the presentation for the 2-torsion submodule $T_2(M) = \mathbb{Z}\langle x, y, z\rangle L/I$ and, moreover, $2 \cdot L \subset I \subset L$ holds.

Example 4.3. In this example we have to run a Gröbner basis of $\langle f_1 = zy - yz + z^2,\ f_2 = zx + y^2,\ f_3 = yx - 3xy\rangle$ up to length bound 11, in order to prove with the Lemma 2.9 that we have computed a finite Gröbner basis. We use degree right lexicographical ordering and obtain $\{f_1,\ f_2,\ f_3,\ 2y^3 + y^2z - 2yz^2 + 2z^3\}\cup$

$$\{y^2z^2 - 4yz^3 + 6z^4,\ y^4 + 27xy^2z - 54xyz^2 + 54xz^3,$$
$$54xy^2z - y^3z - 108xyz^2 + 108xz^3 + 62yz^3 - 124z^4,\ 14z^5,$$
$$14yz^3 - 28z^4,\ 2yz^4 - 6z^5,\ 2xyz^3 - 4xz^4,\ xy^3z,\ 2z^6,\ 2xz^5\}.$$

As we can see from the leading terms, the corresponding module might have 2- and 7-torsion submodules.

There have been 17068 critical pairs created, and internal total degree of intermediate elements was 11. The product criterion has been used 196 times, while the chain criterion was invoked 36711 times. Totally, up to 2.9 GB of memory was allocated.

In the contrast, the Gröbner basis computation of the same input over $\mathbb{Q}$ considered only 14 critical pairs, went up to total degree 6 of intermediate elements, used no product criterion and 9 times the chain criterion with less than 1 MB of memory. The result is $\{f_1,\ f_2,\ f_3\} \cup \{z^5, yz^3 - 2z^4\}\cup$

$$\{2y^3 + y^2z - 2yz^2 + 2z^3, y^2z^2 - 2z^4, xy^2z - 2xyz^2 + 2xz^3\}.$$

This demonstrates once again, how technically involved computations with free algebras over rings as coefficients are.

5 IMPLEMENTATION

We have created a powerful implementation called Letterplace [16] in the framework of Singular [7]. Its' extension to coefficient rings like $\mathbb{Z}$ addresses the following functions with the current

release for ideals and subbimodules of a free bimodule of a finite rank. We provide a vast family of monomial and module orderings including three kinds of orderings eliminating variables or free bimodule components.

`twostd`: a two-sided Gröbner basis; when executed with respect to an elimination ordering, it allows to eliminate variables [6], and thus to compute kernels of ring morphisms and preimages of ideals under such morphisms;

`reduce (NF)`: a normal form of a vector or a polynomial with respect to a two-sided Gröbner basis;

`syz`: a generating set of a syzygy bimodule [5] of an input;

`modulo`: kernel of a bimodule homomorphism;

`lift`: computation of a transformation matrix between a module and its submodule, in other words expressing generators of a submodule in terms of generators of a module;

`liftstd`: computation of a two-sided Gröbner basis and a transformation matrix of a given ideal or subbimodule and, optionally, a syzygy bimodule.

6 CONCLUSION AND FUTURE WORK

Following Mora's "manual for creating own Gröbner basis theory" [22], we have considered the case of free non-commutative Gröbner bases for ideals and bimodules over $\mathbb{Z}\langle X\rangle$. We have derived novel information on the building critical pairs and on criteria to discard them when possible. Armed with this theoretical and algorithmic knowledge, we have created an implementation in a Singular subsystem Letterplace, which offers a rich functionality at a decent speed. We are not aware of yet other systems or packages, which can do such computations.

In this paper we have demonstrated several important applications of our algorithms and their implementation, in particular the determination of torsion submodules with respect to natural numbers.

A further adaptation of our implementation to the explicitly given $\mathbb{Z}/m\mathbb{Z}$ is planned, as well as the development (also a theoretical) of one-sided Gröbner bases in factor algebras (over fields, Letterplace already offers `rightstd`). More functions for dealing with matrices will make possible the usage of our implementation as a backend from the system HomAlg [3]. This system performs homological algebra computations within computable Abelian categories and uses other computer algebra systems as backends for concrete calculations with matrices over rings. Also big systems like SageMath [25] and OSCAR [26] can use our implementation as backend.

7 ACKNOWLEDGEMENTS

The authors are grateful to Hans Schönemann, Gerhard Pfister (Kaiserslautern), Anne Frühbis-Krüger (Oldenburg), Leonard Schmitz, Eva Zerz (RWTH Aachen) and Evelyne Hubert (INRIA) for fruitful discussions.

The first and third authors (V. Levandovskyy and K. Abou Zeid) have been supported by Project II.6 of SFB-TRR 195 "Symbolic Tools in Mathematics and their Applications" of the German Research Foundation (DFG).

The work of the second author (T. Metzlaff) has been supported by European Union's Horizon 2020 research and innovation programme under the Marie Skłodowska-Curie Actions, grant agreement 813211 (POEMA).

REFERENCES

[1] J. Apel. 2000. Computational ideal theory in finitely generated extension rings. *Theor. Comput. Sci.* 244, 1-2 (2000), 1–33.
[2] J. Apel and U. Klaus. 1991. *FELIX* – an assistant for algebraists. In *Proc. ISSAC'91*. ACM Press, 382–389. See also http://felix.hgb-leipzig.de.
[3] Mohamed Barakat, Sebastian Gutsche, and Markus Lange-Hegermann. 2019. homalg - A homological algebra meta-package for computable Abelian categories. https://homalg-project.github.io/homalg_project/homalg/.
[4] George M. Bergman. 1977. The diamond lemma for ring theory. *Adv. Math.* 29 (1977), 178–218. https://doi.org/10.1016/0001-8708(78)90010-5
[5] Holger Bluhm and Martin Kreuzer. 2007. Computation of two-sided syzygies over non-commutative rings. *Contemp. Math.* 421 (2007), 45–64.
[6] M. A. Borges and M. Borges. 1998. Gröbner bases property on elimination ideal in the noncommutative case. In *Gröbner bases and applications*, B. Buchberger and F. Winkler (Eds.). Cambridge University Press, 323–337.
[7] Wolfram Decker, Gert-Martin Greuel, Gerhard Pfister, and Hans Schönemann. 2020. Singular 4-1-3 — A computer algebra system for polynomial computations. http://www.singular.uni-kl.de.
[8] Christian Eder and Tommy Hofmann. 2019. Efficient Gröbner Bases Computation over Principal Ideal Rings. https://arxiv.org/abs/1906.08543.
[9] Christian Eder, Gerhard Pfister, and Adrian Popescu. 2016. New Strategies for Standard Bases over Z. https://arxiv.org/abs/1609.04257.
[10] Christian Eder, Gerhard Pfister, and Adrian Popescu. 2018. Standard Bases over Euclidean Domains. https://arxiv.org/abs/1811.05736.
[11] Heinz Kredel. 2015. Parametric Solvable Polynomial Rings and Applications. In *Proc. CASC'15*, Vladimir P. Gerdt, Wolfram Koepf, Werner M. Seiler, and Evgenii V. Vorozhtsov (Eds.). Springer International Publishing, Cham, 275–291. https://doi.org/10.1007/978-3-319-24021-3_21
[12] Roberto La Scala. 2014. Extended letterplace correspondence for nongraded noncommutative ideals and related algorithms. *Int. J. Algebra Comput.* 24, 8 (2014), 1157–1182.
[13] Roberto La Scala and Viktor Levandovskyy. 2009. Letterplace ideals and non-commutative Gröbner bases. *Journal of Symbolic Computation* 44, 10 (2009), 1374–1393. https://doi.org/10.1016/j.jsc.2009.03.002
[14] Roberto La Scala and Viktor Levandovskyy. 2013. Skew polynomial rings, Gröbner bases and the letterplace embedding of the free associative algebra. *Journal of Symbolic Computation* 48, 1 (2013), 110–131. http://dx.doi.org/10.1016/j.jsc.2012.05.003
[15] Viktor Levandovskyy. 2005. Non–commutative computer algebra for polynomial algebras: Gröbner bases, applications and implementation. http://kluedo.ub.uni-kl.de/volltexte/2005/1883/.
[16] Viktor Levandovskyy, Karim Abou Zeid, and Hans Schönemann. 2020. Singular:Letterplace — A Singular 4-1-3 Subsystem for Non-commutative Finitely Presented Algebras. http://www.singular.uni-kl.de.
[17] Viktor Levandovskyy, Hans Schönemann, and Karim Abou Zeid. 2020. Letterplace — a Subsystem of Singular for computations with free algebras via Letterplace embedding. In *Proc. ISSAC'20*. ACM Press. to appear.
[18] Viktor Levandovskyy, Grischa Studzinski, and Benjamin Schnitzler. 2013. Enhanced Computations of Gröbner Bases in Free Algebras as a New Application of the Letterplace Paradigm. In *Proc. of the International Symposium on Symbolic and Algebraic Computation (ISSAC'13)*, Manuel Kauers (Ed.). ACM Press, 259 – 266.
[19] Huishi Li. 2012. Algebras defined by monic Gröbner bases over rings. *International Mathematical Forum* 7 (2012), 1427–1450.
[20] Daniel Lichtblau. 2012. Effective computation of strong Gröbner bases over Euclidean domains. *Illinois Journal of Mathematics* 56 (2012), 177–194.
[21] Thomas Markwig, Yue Ren, and Oliver Wienand. 2015. Standard bases in mixed power series and polynomial rings over rings. *Journal of Symbolic Computation* 79 (09 2015). https://doi.org/10.1016/j.jsc.2016.08.009
[22] Teo Mora. 2016. *Solving Polynomial Equation Systems IV: Volume 4, Buchberger Theory and Beyond* (1st ed.). Cambridge University Press.
[23] Franz Pauer. 2007. Gröbner bases with coefficients in rings. *Journal of Symbolic Computation* 42 (2007), 1003 – 1011. https://doi.org/10.1016/j.jsc.2007.06.006
[24] F. Leon Pritchard. 1996. The ideal membership problem in non-commutative polynomial rings. *J. Symb. Comput.* 22, 1 (1996), 27–48. https://doi.org/10.1006/jsco.1996.0040
[25] W. A. Stein et al. 2020. *Sage Mathematics Software*. The Sage Development Team.
[26] The OSCAR Team. 2020. The OSCAR project. https://oscar.computeralgebra.de.

Some Properties of Multivariate Differential Dimension Polynomials and their Invariants

Alexander Levin
levin@cua.edu
The Catholic University of America
Washington, D. C. 20064, USA

ABSTRACT

In this paper we obtain new results on multivariate dimension polynomials of differential field extensions associated with partitions of basic sets of derivations. We prove that the coefficient of the summand of the highest possible degree in the canonical representation of such a polynomial is equal to the differential transcendence degree of the extension. We also give necessary and sufficient conditions under which the multivariate dimension polynomial of a differential field extension of a given differential transcendence degree has the simplest possible form. Furthermore, we describe some relationships between a multivariate dimension polynomial of a differential field extension and dimensional characteristics of subextensions defined by subsets of the basic sets of derivations.

CCS CONCEPTS

• **Computing methodologies → Symbolic and algebraic manipulation**.

KEYWORDS

Differential field extension, differential dimension polynomial, differential transcendence degree

ACM Reference Format:
Alexander Levin. 2020. Some Properties of Multivariate Differential Dimension Polynomials and their Invariants. In *International Symposium on Symbolic and Algebraic Computation (ISSAC '20), July 20–23, 2020, Kalamata, Greece.* ACM, New York, NY, USA, 8 pages. https://doi.org/10.1145/3373207.3404013

1 INTRODUCTION

Differential dimension polynomials introduced in [5] by E. Kolchin play the same role in differential algebra as Hilbert polynomials play in commutative algebra and algebraic geometry. An important feature of differential dimension polynomials is that they describe in exact terms the freedom degree of a continuous dynamic system as well as the number of arbitrary constants in the general solution of a system of algebraic partial differential equations. The following fundamental result introduces and describes main properties of a (univariate) dimension polynomial of a finitely generated differential field extension. This theorem combines the results obtained in [6, Chapter II, Sections 12 and 13] adjusted to the case of differential fields of characteristic zero.

Theorem 1.1. *Let K be a differential field of characteristic zero, that is, a field considered together with the action of a set $\Delta = \{\delta_1, \dots, \delta_m\}$ of mutually commuting derivations of K into itself. Let Θ denote the free commutative semigroup of all power products of the form $\theta = \delta_1^{k_1} \dots \delta_m^{k_m}$ $(k_i \geq 0)$, let* ord $\theta = \sum_{i=1}^{m} k_i$, *and for any $r \geq 0$, let $\Theta(r) = \{\theta \in \Theta \mid \text{ord } \theta \leq r\}$. Let $L = K\langle \eta_1, \dots, \eta_n \rangle$ be a differential field extension of K generated by a finite set $\eta = \{\eta_1, \dots, \eta_n\}$. (As a field, $L = K(\{\theta\eta_j | \theta \in \Theta, 1 \leq j \leq n\})$.) Then there exists a polynomial $\omega_{\eta|K}(t) \in \mathbb{Q}[t]$ such that*

(i) $\omega_{\eta|K}(r) = \text{tr. deg}_K K(\{\theta\eta_j | \theta \in \Theta(r), 1 \leq j \leq n\})$ *for all sufficiently large $r \in \mathbb{Z}$;*

(ii) $\deg \omega_{\eta|K} \leq m$ *and $\omega_{\eta|K}(t)$ can be represented as* $\omega_{\eta|K}(t) = \sum_{i=0}^{m} a_i \binom{t+i}{i}$ *where $a_0, \dots, a_m \in \mathbb{Z}$;*

(iii) $d = \deg \omega_{\eta|K}$, a_m *and a_d do not depend on the choice of the system of Δ-generators η of the extension L/K. Moreover, a_m is equal to the differential transcendence degree of L over K (denoted by* Δ-tr. $\deg_K L$*), that is, to the maximal number of elements $\xi_1, \dots, \xi_k \in L$ such that the set $\{\theta\xi_i | \theta \in \Theta, 1 \leq i \leq k\}$ is algebraically independent over K.*

The polynomial $\omega_{\eta|K}$ is called the *differential dimension polynomial* of the differential field extension L/K associated with the system of differential generators η. The invariants $d = \deg \omega_{\eta|K}$ and a_d in the last part of the theorem are called the *differential* (or Δ-) *type* and *typical differential* (or Δ-) *transcendence degree* of the extension L/K; they are denoted by Δ-type$_K L$ and Δ-t. tr. deg$_K L$, respectively.

Differential dimension polynomials provide a power tool for the study of systems of algebraic differential equations. For a wide class of such systems, the dimension polynomial of the corresponding differential field extension expresses the strength of the system of equations in the sense of A. Einstein. This concept, that was introduced in [1] as an important qualitative characteristic of a system of PDEs, can be expressed as a certain differential dimension polynomial, as it is shown in [13]. Another important application of differential dimension polynomials is based on the fact that if P is a prime (in particular, linear) differential ideal of a finitely generated differential algebra R over a differential field K and L is the quotient field of R/P treated as a differential overfield of K, then the differential dimension polynomial of the extension L/K characterizes the ideal P; assigning such polynomials to prime differential ideals has led to a number of new results on the Krull-type dimension of differential algebras and differential field extensions (see, for

ISSAC '20, July 20–23, 2020, Kalamata, Greece

ACM ISBN 978-1-4503-7100-1/20/07…$15.00
https://doi.org/10.1145/3373207.3404013

example, [3], [4], [12] and [7, Chapter 7]). It should be also added that the dimension polynomial associated with a finitely generated differential field extension carries certain differential birational invariants, that is, numbers that do not change when we switch to another finite system of generators of the extension. These invariants are closely connected to some other important characteristics; for example, one of them is the differential transcendence degree of the extension. Among recent works on univariate differential dimension polynomials one has to mention the work of O. Sanchez [14] on the evaluation of the coefficients of a differential dimension polynomial, the work of J. Freitag, O. Sanchez and W. Li on the definability of Kolchin polynomials, and works of M. Lange-Hegermann [8] and [9], where the author introduced a differential dimension polynomial of a characterizable (not necessarily prime) differential ideal and a countable differential polynomial that generalizes the concept of differential dimension polynomial.

In 2001 the author introduced a concept of a multivariate differential dimension polynomial of a finitely generated differential field extension associated with a partition of the set of basic derivations Δ (see [10]). The proof of the corresponding existence theorem that generalizes the first two parts of Theorem 1.1, was based on a special type of reduction in a ring of differential polynomials that takes into account the partition of Δ. It was also shown that a multivariate differential dimension polynomial carries essentially more differential birational invariants of the corresponding differential field extension than its univariate counterpart. As it is demonstrated in Example 3.18, a multivariate dimension polynomial associated with an algebraic differential equation with parameters can carry all this parameters, while the univariate dimension polynomial determines just some relation between the parameters. Therefore, there is a strong motivation for the study of multivariate differential dimension polynomials and their invariants. The main difficulty in this study is due to the fact that a multivariate dimension polynomial of a prime differential polynomial ideal is determined by a characteristic set with respect to several term orderings. Such sets were introduced in [10], but the corresponding theory is in its infancy. Another problem, that is partially solved in this paper, is to characterize invariants of multivariate dimension polynomials and to find relationships between invariants of such polynomials associated with different partitions of the basic set of derivations.

In this paper we obtain new results on multivariate differential dimension polynomials of differential field extensions associated with partitions of the basic sets of derivations. We give necessary and sufficient conditions under which the multivariate dimension polynomial of a differential field extension of a given differential transcendence degree has the simplest possible form. This result (Theorem 3.16) generalizes the corresponding property of univariate differential dimension polynomials proved in [15]. We also prove that the coefficient of the summand of the highest possible degree in the canonical representation of a multivariate dimension polynomial is equal to the differential transcendence degree of the extension. Furthermore, we obtain some relationships between a multivariate dimension polynomial of a differential field extension and dimensional characteristics of subextensions defined by subsets of the basic sets of derivations.

2 PRELIMINARIES

Throughout the paper $\mathbb{Z}$, $\mathbb{N}$ and $\mathbb{Q}$ denote the sets of all integers, all nonnegative integers and all rational numbers, respectively. If M is a finite set, then Card M will denote the number of elements of M. By a ring we always mean an associative ring with unity. Every ring homomorphism is unitary (maps unity onto unity), every subring of a ring contains the unity of the ring, and every algebra over a commutative ring is unitary. Unless otherwise indicated, every field is supposed to have zero characteristic.

A *differential ring* is a commutative ring R considered together with a finite set Δ of mutually commuting derivations of R into itself. The set Δ is called a basic set of the differential ring R that is also called a Δ-ring. A subring (ideal) R_0 of a Δ-ring R is called a differential (or Δ-) subring of R (respectively, a differential (or Δ-) ideal of R) if $\delta(R_0) \subseteq R_0$ for any $\delta \in \Delta$. If a differential (Δ-) ring is a field, it is called a differential (or Δ-) field. In what follows, Θ (or Θ_Δ if we want to indicate the basic set) denotes the free commutative semigroup generated by Δ (that is, if $\Delta = \{\delta_1, \dots, \delta_m\}$, then $\Theta = \{\theta = \delta_1^{k_1} \dots \delta_m^{k_m} \mid k_1, \dots, k_m \in \mathbb{N}\}$).

If R is a Δ-ring and $S \subseteq R$, then the smallest Δ-ideal of R containing S is denoted by $[S]$ (as an ideal, it is generated by the set $\{\theta\xi \mid \xi \in S\}$). If the set S is finite, $S = \{\xi_1, \dots, \xi_q\}$, we say that the Δ-ideal $I = [S]$ is finitely generated, write $I = [\xi_1, \dots, \xi_q]$) and call $\xi_1, \dots, \xi_q$ differential (or Δ-) generators of I. If a Δ-ideal is prime (in the usual sense), it is called a *prime* differential (or Δ-) *ideal*.

Let R_1 and R_2 be two differential rings with the same basic set $\Delta = \{\delta_1, \dots, \delta_m\}$. (More rigorously, we assume that there exist injective mappings of the set Δ into the sets of mutually commuting derivations of the rings R_1 and R_2. For convenience we will denote the images of elements of Δ under these mappings by the same symbols $\delta_1, \dots, \delta_m$). A ring homomorphism $\phi : R \longrightarrow S$ is called a *differential* (or Δ-) *homomorphism* if $\phi(\delta a) = \delta\phi(a)$ for any $\delta \in \Delta$, $a \in R$.

If K is a Δ-field and K_0 a subfield of K which is also a Δ-subring of K, then K_0 is said to be a differential (or Δ-) subfield of K, and K is called a differential (or Δ-) field extension or a Δ-overfield of K_0. We also say that we have a Δ-field extension K/K_0. In this case, if $S \subseteq K$, then the intersection of all Δ-subfields of K containing K_0 and S is the unique Δ-subfield of K containing K_0 and S and contained in every Δ-subfield of K containing K_0 and S. It is denoted by $K_0\langle S\rangle$ or by $K_0\langle S\rangle_\Delta$ if we want to indicate the set of basic derivations Δ. If $K = K_0\langle S\rangle$ and the set S is finite, $S = \{\eta_1, \dots, \eta_n\}$, then K is said to be a finitely generated Δ-field extension of K_0 with the set of Δ-generators $\{\eta_1, \dots, \eta_n\}$. In this case we write $K = K_0\langle\eta_1, \dots, \eta_n\rangle$. It is easy to see that the field $K_0\langle\eta_1, \dots, \eta_n\rangle$ coincides with the field $K_0(\{\theta\eta_i | \theta \in \Theta, 1 \le i \le n\})$.

Let L/K be a Δ-field extension. We say that a set $U \subseteq L$ is Δ-*algebraically dependent* over K, if the family $\{\theta(u) \mid u \in U, \theta \in \Theta\}$ is algebraically dependent over K. Otherwise, the family U is said to be Δ-*algebraically independent* over K. An element $u \in L$ is said to be Δ-algebraic over K if the set $\{u\}$ is Δ-algebraically dependent over K. A maximal Δ-algebraically independent over K subset of L is called a differential (or Δ-) transcendence basis of L over K (or of the extension L/K). It is known (see [6, Chapter II]) that every system of Δ-generators of L/K contains a Δ-transcendence basis of L over K and if L/K is finitely generated as a Δ-field extension,

then all Δ-transcendence bases have the same number of elements called the *differential (or Δ-) transcendence degree* of L over K; it is denoted by Δ-tr. $\deg_K L$.

If K is a Δ-field and $Y = \{y_1, \dots, y_n\}$ is a finite set of symbols, then one can consider the countable set of symbols $\Theta Y = \{\theta y_j | \theta \in \Theta, 1 \le j \le n\}$ and the polynomial ring $R = K[\{\theta y_j | \theta \in \Theta, 1 \le j \le n\}]$ in the set of indeterminates ΘY. This polynomial ring is naturally viewed as a Δ-ring where $\delta(\theta y_j) = (\delta\theta) y_j$ ($\delta \in \Delta$, $\theta \in \Theta$, $1 \le j \le n$) and the elements of Δ act on the coefficients of the polynomials as they act in K. The ring R is called the *ring of differential polynomials* in differential (Δ-) indeterminates $y_1, \dots, y_n$ over the Δ-field K. This ring is denoted by $K\{y_1, \dots, y_n\}$; its elements are called differential (or Δ-) polynomials.

MULTIVARIATE NUMERICAL POLYNOMIALS OF SUBSETS OF $\mathbb{N}^m$

Definition 2.1. *A polynomial $f(t_1, \dots, t_p)$ in p variables ($p \ge 1$) with rational coefficients is said to be* numerical *if $f(t_1, \dots, t_p) \in \mathbb{Z}$ for all sufficiently large $t_1, \dots, t_p \in \mathbb{Z}$, that is, there exists $(s_1, \dots, s_p) \in \mathbb{Z}^p$ such that $f(r_1, \dots, r_p) \in \mathbb{Z}$ whenever $(r_1, \dots, r_p) \in \mathbb{Z}^p$ and $r_i \ge s_i$ ($1 \le i \le p$).*

Clearly, every polynomial with integer coefficients is numerical. As an example of a numerical polynomial in p variables with non-integer coefficients one can consider $\prod_{i=1}^{p} \binom{t_i}{m_i}$ $(m_1, \dots, m_p \in \mathbb{Z})$, where $\binom{t}{k} = \frac{t(t-1)\dots(t-k+1)}{k!}$ for any $k \in \mathbb{Z}, k \ge 1$, $\binom{t}{0} = 1$, and $\binom{t}{k} = 0$ if k is a negative integer.

If f is a numerical polynomial in p variables ($p > 1$), then $\deg f$ and $\deg_{t_i} f$ ($1 \le i \le p$) will denote the total degree of f and the degree of f relative to the variable t_i, respectively. The following theorem gives the "canonical" representation of a numerical polynomial in several variables.

Theorem 2.2. *Let $f(t_1, \dots, t_p)$ be a numerical polynomial in p variables $t_1, \dots, t_p$, and let $\deg_{t_i} f = m_i$ ($1 \le i \le p$). Then the polynomial $f(t_1, \dots, t_p)$ can be represented as*

$$f(t_1, \dots t_p) = \sum_{i_1=0}^{m_1} \dots \sum_{i_p=0}^{m_p} a_{i_1 \dots i_p} \binom{t_1 + i_1}{i_1} \dots \binom{t_p + i_p}{i_p} \qquad (1)$$

with integer coefficients $a_{i_1 \dots i_p}$ that are uniquely defined by the numerical polynomial.

In the rest of this section we deal with subsets of $\mathbb{N}^m$ where the positive integer m is represented as a sum of p nonnegative integers $m_1, \dots, m_p$ ($p \ge 1$). In other words, we fix a partition $(m_1, \dots, m_p)$ of m.

If $A \subseteq \mathbb{N}^m$, then for any $r_1, \dots, r_p \in \mathbb{N}$, $A(r_1, \dots, r_p)$ will denote the subset of A that consists of all m-tuples $(a_1, \dots, a_m)$ such that $a_1 + \dots + a_{m_1} \le r_1$, $a_{m_1+1} + \dots + a_{m_1+m_2} \le r_2, \dots, a_{m_1+\dots+m_{p-1}+1} + \dots + a_m \le r_p$. Furthermore, we shall associate with the set A a set $V_A \subseteq \mathbb{N}^m$ that consists of all m-tuples $v = (v_1, \dots, v_m) \in \mathbb{N}^m$ that are not greater than or equal to any m-tuple from A with respect to the product order on $\mathbb{N}^m$. (Recall that the product order on the set $\mathbb{N}^k$ ($k \in \mathbb{N}, k \ge 1$) is a partial order $\le_P$ on $\mathbb{N}^k$ such that $c = (c_1, \dots, c_k) \le_P c' = (c'_1, \dots, c'_k)$ if and only if $c_i \le c'_i$ for all $i = 1, \dots, k$. If $c \le_P c'$ and $c \ne c'$, we write $c <_P c'$). Clearly, an element $v = (v_1, \dots, v_m) \in \mathbb{N}^m$ belongs to V_A if and only if for any element $(a_1, \dots, a_m) \in A$ there exists $i \in \mathbb{N}, 1 \le i \le m$, such that $a_i > v_i$.

The following two theorems proved in [7, Chapter 2] generalize the well-known Kolchin's result on the numerical polynomials of subsets of $\mathbb{N}^m$ (see [6, Chapter 0, Lemma 17]) and give an explicit formula for the numerical polynomials in p variables associated with a finite subset of $\mathbb{N}^m$.

Theorem 2.3. *Let A be a subset of $\mathbb{N}^m$ where $m = m_1 + \dots + m_p$ for some nonnegative integers $m_1, \dots, m_p$ ($p \ge 1$). Then there exists a numerical polynomial $\omega_A(t_1, \dots, t_p)$ in p variables with the following properties:*

(i) *$\omega_A(r_1, \dots, r_p) = \mathrm{Card}\, V_A(r_1, \dots, r_p)$ for all sufficiently large $(r_1, \dots, r_p) \in \mathbb{N}^p$ (i. e., there exist $(s_1, \dots, s_p) \in \mathbb{N}^p$ such that the equality holds for all $(r_1, \dots, r_p) \in \mathbb{N}^p$ such that $(s_1, \dots, s_p) \le_P (r_1, \dots, r_p)$).*

(ii) *$\deg \omega_A \le m$ and $\deg_{t_i} \omega_A \le m_i$ for $i = 1, \dots, p$.*

(iii) *$\deg \omega_A = m$ if and only if the set A is empty. In this case $\omega_A(t_1, \dots, t_p) = \prod_{i=1}^{p} \binom{t_i + m_i}{m_i}$.*

(iv) *ω_A is a zero polynomial if and only if $(0, \dots, 0) \in A$.*

Definition 2.4. *The polynomial $\omega_A(t_1, \dots, t_p)$ is called the dimension polynomial of the set $A \subseteq \mathbb{N}^m$ associated with the partition $(m_1, \dots, m_p)$ of m.*

Theorem 2.5. *Let $A = \{a_1, \dots, a_n\}$ be a finite subset of $\mathbb{N}^m$ where n is a positive integer and $m = m_1 + \dots + m_p$ for some nonnegative integers $m_1, \dots, m_p$ ($p \ge 1$). Let $a_i = (a_{i1}, \dots, a_{im})$ $(1 \le i \le n)$ and for any $l \in \mathbb{N}$, $0 \le l \le n$, let $\Gamma(l, n)$ denote the set of all l-element subsets of the set $\mathbb{N}_n = \{1, \dots, n\}$. Furthermore, for any $\sigma \in \Gamma(l, p)$, let $\bar{a}_{\sigma j} = \max\{a_{ij} | i \in \sigma\}$ $(1 \le j \le m)$ and $b_{\sigma j} = \sum_{h \in \sigma_j} \bar{a}_{\sigma h}$. Then*

$$\omega_A(t_1, \dots, t_p) = \sum_{l=0}^{n} (-1)^l \sum_{\sigma \in \Gamma(l,n)} \prod_{j=1}^{p} \binom{t_j + m_j - b_{\sigma j}}{m_j} \qquad (2)$$

Remark 2.6. Clearly, if $A \subseteq \mathbb{N}^m$ and A' is the set of all minimal elements of the set A with respect to the product order on $\mathbb{N}^m$, then the set A' is finite and $\omega_A(t_1, \dots, t_p) = \omega_{A'}(t_1, \dots, t_p)$. Thus, Theorem 2.5 gives an algorithm that allows one to find the dimension polynomial of any subset of $\mathbb{N}^m$ (with a given representation of m as a sum of p positive integers): one should first find the set of all minimal points of the subset and then apply Theorem 2.5.

3 MULTIVARIATE DIFFERENTIAL DIMENSION POLYNOMIALS AND THEIR INVARIANTS

Let K be a differential (Δ-) field whose basic set of derivations Δ is represented as the union of p nonempty disjoint subsets ($p \ge 1$):

$$\Delta = \Delta_1 \bigcup \dots \bigcup \Delta_p \qquad (3)$$

where $\Delta_i = \{\delta_{i1}, \dots, \delta_{im_i}\}$ for $i = 1, \dots, p$ ($m_1 + \dots + m_p = m$ where $m = \mathrm{Card}\, \Delta$). Thus, we fix a partition of Δ.

Let Θ_i denote the free commutative semigroup generated by Δ_i ($1 \le i \le p$) and let Θ be the free commutative semigroup generated by the whole set Δ. For any $\theta = \delta_{11}^{k_{11}} \dots \delta_{1m_1}^{k_{1m_1}} \delta_{21}^{k_{21}} \dots \delta_{pm_p}^{k_{pm_p}} \in \Theta$, the numbers $\mathrm{ord}_i\, \theta = \sum_{j=1}^{m_i} k_{ij}$ ($i = 1, \dots, p$) and $\mathrm{ord}\, \theta = \sum_{i=1}^{p} \mathrm{ord}_i\, \theta$

will be called the *order of θ with respect to Δ_i* and the *order of θ*, respectively. If $\theta, \theta' \in \Theta$, we say that θ' divides θ (or that θ is a multiple of θ') and write $\theta' \mid \theta$ if there exists $\theta'' \in \Theta$ such that $\theta = \theta''\theta'$. As usual, the least common multiple of elements $\theta_1 = \prod_{i=1}^{p}\prod_{j=1}^{m_i}\delta_{ij}^{k_{ij1}}, \dots, \theta_q = \prod_{i=1}^{p}\prod_{j=1}^{m_i}\delta_{ij}^{k_{ijq}} \in \Theta$ is the element $\theta = \prod_{i=1}^{p}\prod_{j=1}^{m_i}\delta_{ij}^{k_{ij}}$, where $k_{ij} = \max\{k_{ijl} | 1 \le l \le q\}$ $(1 \le i \le p, 1 \le j \le m_i)$, denoted by $\operatorname{lcm}(\theta_1, \dots, \theta_q)$.

If $r_1, \dots, r_p, r \in \mathbb{N}$, the sets $\{\theta \in \Theta \mid \operatorname{ord}_i\ \theta \le r_i \text{ for } i = 1, \dots, p\}$ and $\{\theta \in \Theta \mid \operatorname{ord}\ \theta \le r\}$ will be denoted by $\Theta(r_1, \dots, r_p)$ and $\Theta(r)$, respectively. If $\xi \in K$ and $\Theta' \subseteq \Theta$, then $\Theta'\xi$, will denote the set $\{\theta(\xi) \mid \theta \in \Theta'\}$.

We consider p orderings $<_1, \cdots <_p$ of the semigroup Θ defined as follows. If $\theta = \delta_{11}^{k_{11}} \dots \delta_{pm_p}^{k_{pm_p}}$ and $\theta' = \delta_{11}^{l_{11}} \dots \delta_{pm_p}^{l_{pm_p}}$ are elements of Θ, then $\theta <_i \theta'$ if and only if the vector
$(\operatorname{ord}_i\ \theta, \operatorname{ord}\theta, \operatorname{ord}_1\ \theta, \dots, \operatorname{ord}_{i-1}\ \theta, \operatorname{ord}_{i+1}\ \theta, \dots, \operatorname{ord}_p\ \theta, k_{i1},$
$\dots, k_{im}, k_{11}, \dots, k_{1m_1}, k_{21}, \dots, k_{i-1,m_{i-1}}, k_{i+1,1}, \dots, k_{pm_p})$
is less than the vector
$(\operatorname{ord}_i\ \theta', \operatorname{ord}\theta', \operatorname{ord}_1\ \theta', \dots, \operatorname{ord}_{i-1}\ \theta', \operatorname{ord}_{i+1}\ \theta', \dots, \operatorname{ord}_p\ \theta',$
$l_{i1}, \dots, l_{im}, l_{11}, \dots, l_{1m_1}, l_{21}, \dots, l_{i-1,m_{i-1}}, l_{i+1,1}, \dots, l_{pm_p})$
with respect to the lexicographic order on $\mathbb{N}^{m+p+1}$.
Let $K\{y_1, \dots, y_n\}$ be the ring of Δ-polynomials in Δ-indeterminates $y_1, \dots, y_n$ over K. Then the elements θy_i $(\theta \in \Theta, 1 \le i \le n)$ will be called terms, and the set of all terms ΘY will be considered together with p orderings that correspond to the orderings of Θ and are denoted by the same symbols $<_1, \dots, <_p$. These orderings of ΘY are defined as follows. $\theta y_j <_i \theta' y_k$ $(\theta, \theta' \in \Theta, 1 \le j, k \le n, 1 \le i \le p)$ if and only if $\theta <_i \theta'$ or $\theta = \theta'$ and $j < k$. By the ith order of a term $u = \theta y_j$ we mean the number $\operatorname{ord}_i u = \operatorname{ord}_i \theta$. The number $\operatorname{ord}\ u = \operatorname{ord}\ \theta$ is called the order of u.

We say that a term $u = \theta y_i$ is divisible by a term $v = \theta' y_j$ and write $v \mid u$, if $i = j$ and $\theta' \mid \theta$. For any terms $u_1 = \theta_1 y_j, \dots, u_q = \theta_q y_j$ with the same Δ-indeterminate y_j, the term $\operatorname{lcm}(\theta_1, \dots, \theta_q)y_j$ is called the least common multiple of $u_1, \dots, u_q$, it is denoted by $\operatorname{lcm}(u_1, \dots, u_q)$.

If $A \in K\{y_1, \dots, y_n\}$, $A \notin K$, and $1 \le i \le p$, then the highest with respect to the ordering $<_i$ term that appears in A is called the i-leader of the Δ-polynomial A. It is denoted by $u_A^{(i)}$. If A is written as a polynomial in one variable $u_A^{(1)}$, $A = I_d(u_A^{(1)})^d + I_{d-1}(u_A^{(1)})^{d-1} + \cdots + I_0$ ($I_d, I_{d-1}, \dots, I_0$ do not contain $u_A^{(1)}$), then I_d is called the *leading coefficient* of the Δ-polynomial A and the partial derivative $\partial A/\partial u_A^{(1)} = dI_d(u_A^{(1)})^{d-1} + (d-1)I_{d-1}(u_A^{(1)})^{d-2} + \cdots + I_1$ is called the *separant* of A. The leading coefficient and the separant of A are denoted by I_A and S_A, respectively.

Definition 3.1. *Let A and B be Δ-polynomials in $K\{y_1, \dots, y_n\}$. We say that A has lower rank then B and write* $\operatorname{rk}\ A < \operatorname{rk}\ B$ *if either $A \in K, B \notin K$, or the vector $(u_A^{(1)}, \deg_{u_A^{(1)}} A, \operatorname{ord}_2 u_A^{(2)}, \dots, \operatorname{ord}_p u_A^{(p)})$ is less than the vector $(u_B^{(1)}, \deg_{u_B^{(1)}} B, \operatorname{ord}_2 u_B^{(2)}, \dots, \operatorname{ord}_p u_B^{(p)})$ with respect to the lexicographic order ($u_A^{(1)}$ and $u_B^{(1)}$ are compared with respect to $<_1$ and all other coordinates are compared with respect to the natural order on $\mathbb{N}$). If the two vectors are equal (or $A \in K$ and $B \in K$) we say that the Δ-polynomials A and B are of the same rank and write* $\operatorname{rk}\ A = \operatorname{rk}\ B$.

Definition 3.2. *Let A and B be Δ-polynomials in $K\{y_1, \dots, y_n\}$ and $A \notin K$. We say that B is reduced with respect to A if the following two conditions hold.*

(i) *B does not contain any term $\theta u_A^{(1)}$ $(\theta \in \Theta, \theta \ne 1)$ such that $ord_i(\theta u_A^{(i)}) \le ord_i u_B^{(i)}$ for $i = 2, \dots, p$.*

(ii) *If B contains $u_A^{(1)}$, then either there exists j, $2 \le j \le p$, such that $ord_j u_B^{(j)} < ord_j u_A^{(j)}$ or $ord_j u_A^{(j)} \le ord_j u_B^{(j)}$ for all $j = 2, \dots, p$ and $deg_{u_A^{(1)}} B < deg_{u_A^{(1)}} A$.*

A Δ-polynomial B is said to be reduced with respect to a set $\mathcal{A} \subseteq K\{y_1, \dots, y_n\}$ if B is reduced with respect to every element of $\mathcal{A}$.

Remark 3.3. The last definition shows that a Δ-polynomial B is not reduced with respect to a Δ-polynomial A $(A \notin K)$ if either B contains a term $\theta u_A^{(1)}$ $(\theta \in \Theta, \theta \ne 1)$ such that $ord_i(\theta u_A^{(i)}) \le ord_i u_B^{(i)}$ for $i = 2, \dots, p$ or B contains $u_A^{(1)}$ and in this case $ord_j u_A^{(j)} \le ord_j u_B^{(j)}$ for $j = 2, \dots, p$ and $deg_{u_A^{(1)}} A \le deg_{u_A^{(1)}} B$. This observation is helpful if one would like to show that a Δ-polynomial is not reduced with respect to some other Δ-polynomial.

Definition 3.4. *A set of Δ-polynomials $\mathcal{A}$ is called autoreduced if $\mathcal{A} \bigcap K = \emptyset$ and every element of $\mathcal{A}$ is reduced with respect to any other element of this set.*

The following two statements are proved in [10] (see[10, Theorem 4.5] and [10, Theorem 4.6]).

Proposition 3.5. *Every autoreduced set is finite.*

Proposition 3.6. *Let $\mathcal{A} = \{A_1, \dots, A_r\}$ be an autoreduced set in $K\{y_1, \dots, y_n\}$ and $B \in K\{y_1, \dots, y_n\}$. Then there are a Δ-polynomial B_0 and nonnegative integers p_i, q_i $(1 \le i \le r)$ such that B_0 is reduced with respect to $\mathcal{A}$, $rk\ B_0 \le rk\ B$, and $\prod_{i=1}^{r} I_{A_i}^{p_i} S_{A_i}^{q_i} B \equiv B_0\ (mod[\mathcal{A}])$.*

In what follows, while considering an autoreduced set $\mathcal{A} = \{A_1, \dots, A_r\}$, we always assume that its elements are arranged in order of increasing rank: $\operatorname{rk}\ A_1 < \cdots < \operatorname{rk}\ A_r$.

Definition 3.7. *Let $\mathcal{A} = \{A_1, \dots, A_r\}$ and $\mathcal{B} = \{B_1, \dots, B_s\}$ be two autoreduced sets. Then $\mathcal{A}$ is said to have lower rank than $\mathcal{B}$ if one of the following two cases holds:*

(i) *There exists $k \in \mathbb{N}$ such that $k \le \min\{r, s\}$,* $\operatorname{rk}\ A_i = \operatorname{rk}\ B_i$ *for $i = 1, \dots, k-1$ and* $\operatorname{rk}\ A_k < \operatorname{rk}\ B_k$.

(ii) *$r > s$ and* $\operatorname{rk}\ A_i = \operatorname{rk}\ B_i$ *for $i = 1, \dots, s$.*

If $r = s$ and $\operatorname{rk}\ A_i = \operatorname{rk}\ B_i$ *for $i = 1, \dots, r$, then $\mathcal{A}$ is said to have the same rank as $\mathcal{B}$.*

The statements of Propositions 3.8 and 3.10 below can be obtained by mimicking the proofs of the corresponding statements for classical Ritt-Kolchin autoreduced sets (see [7, Proposition 5.3.10 and Lemma 5.3.12]).

Proposition 3.8. *Every nonempty family of autoreduced sets contains an autoreduced set of lowest rank.*

Definition 3.9. *Let J be a Δ-ideal of the ring of Δ-polynomials $K\{y_1, \dots, y_n\}$. Then an autoreduced subset of J of lowest rank is called a characteristic set of the ideal J.*

Proposition 3.10. *Let $\mathcal{A} = \{A_1, \dots, A_d\}$ be a characteristic set of a Δ-ideal J of the ring of Δ-polynomials $R = K\{y_1, \dots, y_n\}$. Then*

an element $B \in J$ is reduced with respect to the set $\mathcal{A}$ if and only if $B = 0$. In particular, $I_A \notin J$ and $S_A \notin J$ for every $A \in \mathcal{A}$.

Let K be a Δ-field and $L = K\langle \eta_1, \dots, \eta_n \rangle$ a finitely generated Δ-extension of K with a set of Δ-generators $\eta = \{\eta_1, \dots, \eta_n\}$. Then there exists a natural Δ-homomorphism ϕ_η of the ring of Δ-polynomials $K\{y_1, \dots, y_n\}$ onto the Δ-subring $K\{\eta_1, \dots, \eta_n\}$ of L such that $\phi_\eta(a) = a$ for any $a \in K$ and $\phi_\eta(y_j) = \eta_j$ for $j = 1, \dots, n$. If $A \in K\{y_1, \dots, y_n\}$, then $\phi_\eta(A)$ is called the *value* of A at η and is denoted by $A(\eta)$. Obviously, $P = \operatorname{Ker} \phi_\eta$ is a prime Δ-ideal of $K\{y_1, \dots, y_n\}$. It is called the *defining ideal* of η. If we consider the quotient field Q of $R = K\{y_1, \dots, y_n\}/P$ as a Δ-field (where $\delta(\frac{u}{v}) = \frac{v\delta(u) - u\delta(v)}{v^2}$ for any $u, v \in R$), then this quotient field is naturally Δ-isomorphic to the field L. The Δ-isomorphism of Q onto L is identical on K and maps the images of the Δ-indeterminates $y_1, \dots, y_n$ in the factor ring R onto the elements $\eta_1, \dots, \eta_n$, respectively.

Let K be a differential (Δ-) field, $\operatorname{Card} \Delta = m$, and let a partition (3) of Δ be fixed: $\Delta = \Delta_1 \bigcup \cdots \bigcup \Delta_p$ ($p \geq 1$), where $\Delta_i = \{\delta_{i1}, \dots, \delta_{im_i}\}$ ($1 \leq i \leq p$). Furthermore, let $L = K\langle \eta_1, \dots, \eta_n \rangle$ be a Δ-field extension of K generated by a finite set $\eta = \{\eta_1, \dots, \eta_n\}$. Let P be the defining ideal of η and $\mathcal{A} = \{A_1, \dots, A_d\}$ a characteristic set of P.

For any $r_1, \dots, r_p \in \mathbb{N}$, let

$$U'_{r_1 \dots r_p} = \{u \in \Theta Y \mid \operatorname{ord}_i u \leq r_i \text{ for } i = 1, \dots, p \text{ and } u \text{ is not a derivative of any } u^{(1)}_{A_i} \text{ (that is, } u \neq \theta u^{(1)}_{A_i} \text{ for any } \theta \in \Theta; i = 1, \dots, d)\}$$

and let

$$U''_{r_1 \dots r_p} = \{u \in \Theta Y \mid \operatorname{ord}_i u \leq r_i \text{ for } i = 1, \dots, p \text{ and for every } \theta \in \Theta, A \in \mathcal{A} \text{ such that } u = \theta u^{(1)}_A, \text{ there exists } i \in \{2, \dots, p\} \text{ such that } \operatorname{ord}_i(\theta u^{(i)}_A) > r_i\}.$$

(If $p = 1$, $U'_{r_1} = \{u \in \Theta Y \mid \operatorname{ord}_1 u \leq r_1$ and u is not a derivative of any $u^{(1)}_{A_i}\}$ and $U''_{r_1} = \emptyset$.) Furthermore, for any $(r_1 \dots r_p) \in \mathbb{N}^p$, let $U_{r_1 \dots r_p} = U'_{r_1 \dots r_p} \bigcup U''_{r_1 \dots r_p}$.

The following theorem proved in [10, Section 5] establishes the existence and describes the form of a multivariate dimension polynomial associated with a finite system of Δ-generators of a Δ-field extension and with a partition of the set Δ. We give an extended version of this result that follows from the proof of [10, Theorem 5.1].

Theorem 3.11. *With the above notation,*

(i) *For all sufficiently large $(r_1 \dots r_p) \in \mathbb{N}^p$, the set $U_{r_1 \dots r_p}$ is a transcendence basis of $K(\bigcup_{j=1}^{n} \Theta(r_1, \dots, r_p)\eta_j)$ over K.*

(ii) *There exist numerical polynomials $\omega_{\eta|K}(t_1, \dots, t_p)$ and $\phi_{\eta|K}(t_1, \dots, t_p)$ in p variables such that $\omega_{\eta|K}(r_1, \dots, r_p) = \operatorname{Card} U'_{r_1 \dots r_p}$ and $\phi_{\eta|K}(r_1, \dots, r_p) = \operatorname{Card} U''_{r_1 \dots r_p}$ for all sufficiently large $(r_1 \dots r_p) \in \mathbb{N}^p$, so that the polynomial $\Phi_{\eta|K}(t_1, \dots, t_p) = \omega_{\eta|K}(t_1, \dots, t_p) + \phi_{\eta|K}(t_1, \dots, t_p)$ has the property that $\Phi_\eta(r_1, \dots, r_p) = \operatorname{tr.deg}_K K(\bigcup_{j=1}^{n} \Theta(r_1, \dots, r_p)\eta_j)$ for all sufficiently large $(r_1, \dots, r_p) \in \mathbb{N}^p$.*

(iii) $\deg_{t_i} \Phi_{\eta|K} \leq m_i$ ($1 \leq i \leq p$), *so that* $\deg \Phi_{\eta|K} \leq m$ *and the polynomial $\Phi_{\eta|K}(t_1, \dots, t_p)$ can be represented as*

$$\Phi_{\eta|K}(t_1, \dots, t_p) = \sum_{i_1=0}^{m_1} \dots \sum_{i_p=0}^{m_p} a_{i_1 \dots i_p} \binom{t_1 + i_1}{i_1} \dots \binom{t_p + i_p}{i_p} \quad (4)$$

where $a_{i_1 \dots i_p} \in \mathbb{Z}$ for all $i_1 \dots i_p$.

(iv) *$\phi_{\eta|K}(t_1, \dots, t_p)$ is an alternating sum of polynomials in p variables of the form*

$$\phi_{j;k_1,\dots,k_q} = \binom{t_1 + m_1 - b_{1j}}{m_1} \dots \binom{t_{k_1-1} + m_{k_1-1} - b_{k_1-1,j}}{m_{k_1-1}} \left[\binom{t_{k_1} + m_{k_1} - a_{k_1,j}}{m_{k_1}} - \binom{t_{k_1} + m_{k_1} - b_{k_1,j}}{m_{k_1}} \right] \cdot \binom{t_{k_1+1} + m_{k_1+1} - b_{k_1+1,j}}{m_{k_1+1}} \dots \binom{t_{k_q-1} + m_{k_q-1} - b_{k_q-1,j}}{m_{k_q-1}} \cdot \left[\binom{t_{k_q} + m_{k_q} - a_{k_q,j}}{m_{k_q}} - \binom{t_{k_q} + m_{k_q} - b_{k_q,j}}{m_{k_q}} \right] \dots \binom{t_p + m_p - b_{pj}}{m_p},$$

so that $\deg \phi_{\eta|K} < m$

Definition 3.12. *Numerical polynomial $\Phi_{\eta|K}(t_1, \dots, t_p)$, whose existence is established by the last theorem, is called a differential (or Δ-) dimension polynomial of the differential field extension $L = K\langle \eta_1, \dots, \eta_n \rangle$ associated with the set of Δ-generators $\eta = \{\eta_1, \dots, \eta_n\}$ and with partition* (3) *of the basic set of derivations Δ.*

Remark 3.13. With the notation of the last theorem, if $\eta_1, \dots, \eta_n$ are Δ-algebraically independent over K, then

$$\Phi_{\eta|K}(t_1, \dots, t_p) = n \prod_{i=1}^{p} \binom{t_i + m_i}{m_i}. \quad (5)$$

Indeed, all elements $\delta_{11}^{k_{11}} \dots \delta_{1m_1}^{k_{1m_1}} \delta_{21}^{k_{21}} \dots \delta_{pm_p}^{k_{pm_p}}$ such that $\sum_{j=1}^{m_i} k_{ij} \leq r_i$ ($1 \leq i \leq p$) form a transcendence basis of $K(\bigcup_{j=1}^{n} \Theta(r_1, \dots, r_p)\eta_j)$ over K. By Theorem 2.3 (iii), the number of such elements is $n \prod_{i=1}^{p} \binom{r_i + m_i}{m_i}$, so we arrive at formula (5).

Remark 3.14. Theorem 3.11 shows that the main problem in computing the multivariate Δ-dimension polynomial $\Phi_{\eta|K}$ is constructing a characteristic set of the defining Δ-ideal of the Δ-field extension. If this ideal is linear (that is, the defining system of differential equations on the generators of the extension is linear), then this problem was solved in [11] by an algorithm for constructing a Gröbner basis with respect to several term orderings (see [11, Algorithm 1] and [11, Theorem 3.10]). In the nonlinear case the problem of generalizing the Ritt-Kolchin algorithm to the case of autoreduced sets with respect to several term orderings defined above is still open.

For any permutation $(j_1, \dots, j_p)$ of the set $\{1, \dots, p\}$ ($p \geq 1$), let $\leq_{j_1,\dots,j_p}$ denote the corresponding lexicographic order on $\mathbb{N}^p$ such that $(r_1, \dots, r_p) \leq_{j_1,\dots,j_p} (s_1, \dots, s_p)$ if and only if either $r_{j_1} < s_{j_1}$ or there exists $k \in \mathbb{N}$, $1 \leq k \leq p - 1$ such that $r_{j_\nu} = s_{j_\nu}$ for $\nu = 1, \dots, k$ and $r_{j_{k+1}} < s_{j_{k+1}}$. If E is a finite subset of $\mathbb{N}^p$, then E' will denote the set of all p-tuples $e \in E$ that are maximal

elements of E with respect to one of the $p!$ orders $\leq_{j_1,\ldots,j_p}$. Say, if $E = \{(3,0,2),(2,1,1),(0,1,4),(1,0,3),(1,1,6),(3,1,0),(1,2,0)\} \subseteq \mathbb{N}^3$, then $E' = \{(3,0,2),(3,1,0),(1,1,6),(1,2,0)\}$.

The following result gives differential birational invariants carried by a multivariate dimension polynomial of a differential field extension. In particular, it shows that multivariate differential dimension polynomials carry essentially more such invariants than their univariate counterparts.

Theorem 3.15. *Let K be a differential field with a basic set of derivations Δ and let partition (3) of the set Δ into the union of p disjoint sets ($p \geq 1$) be fixed. Let $L = K\langle\eta_1,\ldots,\eta_n\rangle$ be a Δ-field extension of K with the finite set of Δ-generators $\eta = \{\eta_1,\ldots,\eta_n\}$ and let*

$$\Phi_{\eta|K}(t_1,\ldots,t_p) = \sum_{i_1=0}^{m_1}\cdots\sum_{i_p=0}^{m_p} a_{i_1\ldots i_p}\binom{t_1+i_1}{i_1}\cdots\binom{t_p+i_p}{i_p} \quad (6)$$

be the corresponding differential dimension polynomial. Let $E_\eta = \{(i_1\ldots i_p) \in \mathbb{N}^p \mid 0 \leq i_k \leq m_k\ (k = 1,\ldots,p)$ and $a_{i_1\ldots i_p} \neq 0\}$. Then $d = \deg \Phi_{\eta|K}$, $a_{m_1\ldots m_p}$, the elements $(k_1,\ldots,k_p) \in E'_\eta$, the corresponding coefficients $a_{k_1\ldots k_p}$, and the coefficients of the terms of total degree d in $\Phi_{\eta|K}$ do not depend on the system of Δ-generators η.

Proof. The fact that the elements $(k_1,\ldots,k_p)$ of the set E'_η and the corresponding coefficients $a_{k_1\ldots k_p}$ do not depend on the system of Δ-generators η of L/K is established in the proof of Theorem 5.3 of [10] using the observation that if $\zeta = \{\zeta_1,\ldots,\zeta_q\}$ is another system of Δ-generators of L/K, then there exists $(s_1,\ldots,s_p) \in \mathbb{N}^p$ such that $\Phi_{\eta|K}(r_1,\ldots,r_p) \leq \Phi_{\zeta|K}(r_1+s_1,\ldots,r_p+s_p)$ and $\Phi_{\zeta|K}(r_1,\ldots,r_p) \leq \Phi_{\eta|K}(r_1+s_1,\ldots,r_p+s_p)$ for all sufficiently large $(r_1,\ldots,r_p) \in \mathbb{N}^p$. Clearly, these inequalities show that $\deg\Phi_{\zeta|K} = \deg\Phi_{\eta|K}$. Let $d = \deg\Phi_{\eta|K}$. Let us order the terms of total degree d in $\Phi_{\eta|K}$ and $\Phi_{\zeta|K}$ using the lexicographic order $\leq_{p,p-1,\ldots,1}$ and for sufficiently large $r \in \mathbb{N}$, set $x_1 = r$, $x_2 = 2^{x_1}$, $x_3 = 2^{x_2},\ldots,x_p = 2^{x_{p-1}}$, $R = 2^{x_p}$ and $r_i = x_i R$ $(1 \leq i \leq p)$. If $r \to \infty$, then the last two inequalities immediately imply that $\Phi_{\eta|K}$ and $\Phi_{\zeta|K}$ have the same coefficients of the corresponding terms of total degree d. □

The next theorem characterizes one of the invariants of polynomial (6).

Theorem 3.16. *With the notation of the last theorem, $a_{m_1\ldots m_p} = \Delta\text{-tr.deg}_K L$.*

Proof. Let $\Delta\text{-tr.deg}_K L = d$. Then, as it was mentioned in section 2, one can choose a Δ-transcendence basis of L/K from the set η, so we can assume that $\eta_1,\ldots,\eta_d$ form such a basis. Since the family $\{\theta\eta_i \mid \theta \in \Theta,\ 1 \leq i \leq d\}$ is algebraically independent over K, it follows from Remark 3.13 that $\operatorname{tr.deg}_K K(\bigcup_{j=1}^{d}\Theta(r_1,\ldots,r_p)\eta_j) = d\prod_{i=1}^{p}\binom{r_i+m_i}{m_i}$ for all $(r_1,\ldots,r_p) \in \mathbb{N}^p$. Let $F = K\langle\eta_1,\ldots,\eta_d\rangle$. Then every element η_j, $d+1 \leq j \leq n$, is Δ-algebraic over K. It means that there exists a Δ-polynomial $A_j \in F\{y_j\}$ ($F\{y_j\}$ is the ring of Δ-polynomials in one Δ-indeterminate y_j over F) such that $A_j(\eta_j) = 0$. Taking such a polynomial of the smallest possible degree we can assume that $S_{A_j}(\eta_j) \neq 0$. Let $A_j = \sum_{k=0}^{q_j} I_{jk}(u_{A_j}^{(1)})^k$ where all terms of all I_{jk} are less than $u_{A_j}^{(1)}$ with respect to $<_1$.

If $\delta \in \Delta$, then $\delta A_j(\eta_j) = 0$, so
$S_{A_j}(\eta_j)\delta(u_{A_j}^{(1)}(\eta_j)) + \sum_{k=0}^{q_j}\delta(I_{jk}(\eta_j))u_{A_j}^{(1)}(\eta_j) = 0$.
The term $\delta u_{A_j}^{(1)}$ has the form $\theta_j y_j$ for some $\theta_j \in \Theta$ and one can easily see that for any term v in any S_{A_j} or $\delta(I_{jk})$, we have $v <_1 \theta_j y_j$ and $\operatorname{ord}_i v \leq \operatorname{ord}_i u_{A_j}^{(i)} + 1$ $(i = 1,\ldots,p)$. It follows that $\theta_j\eta_j \in F(\{\theta\eta_k \mid \theta \in \Theta, \theta y_k <_1 \theta_j y_j$ and $\operatorname{ord}_i \theta y_k \leq a_{ji}$ for $i = 1,\ldots,p\})$ where $a_{ji} = \operatorname{ord}_i u_{A_j}^{(i)} + 1$ $(1 \leq i \leq p)$.

Since $F = K\left(\bigcup_{k=1}^{d}\bigcup_{(l_1,\ldots,l_p)\in\mathbb{N}^p}\Theta(l_1,\ldots,l_p)\eta_k\right)$, there exist $h_1,\ldots,h_p \in \mathbb{N}$ such that $\theta_j\eta_j \in K(\bigcup_{k=1}^{d}\Theta(h_1,\ldots,h_p)\eta_k \bigcup\{\theta\eta_k \mid \theta \in \Theta, \theta y_k <_1 \theta_j y_j, \operatorname{ord}_i \theta y_k \leq a_{ji}\ (i = 1,\ldots,p)\})$.

Let $\theta' \in \Theta$ and $\theta_j \mid \theta'$. For any $i = 1,\ldots,p$, let $s_i = \operatorname{ord}_i \theta'$ (clearly, $s_i \geq a_{ji}$). Then
$\theta'\eta_j \in K(\bigcup_{k=1}^{d}\Theta(s_1+h_1,\ldots,s_p+h_p)\eta_k \bigcup\{\theta\eta_l \mid \theta \in \Theta, \theta y_j <_1 \theta' y_j, \operatorname{ord}_i\theta \leq \operatorname{ord}_i\theta'\ (1 \leq i \leq p)$ and $\theta_j \nmid \theta\})$.

Therefore, if $r_i \in \mathbb{N}$, $r_i \geq \max_{d+1\leq j\leq n}\{a_{ji}\}$ $(i = 1,\ldots,p)$, then $K(\bigcup_{k=1}^{n}\Theta(r_1,\ldots,r_p)\eta_k) \subseteq K(\bigcup_{k=1}^{d}\Theta(r_1+h_1,\ldots,r_p+h_p)\eta_k \cup \bigcup_{j=d+1}^{n}[\Theta(r_1,\ldots,r_p) \setminus \Theta(r_1-a_{j1},,\ldots,r_p-a_{jp})]\eta_j)$.

It follows that $\Phi_{\eta|K}(r_1,\ldots,r_p) \leq d\prod_{i=1}^{p}\binom{r_i+m_i}{m_i} + \sum_{j=d+1}^{n}\left[\prod_{i=1}^{p}\binom{r_i+m_i}{m_i} - \prod_{i=1}^{p}\binom{r_i-a_{ji}+m_i}{m_i}\right]$ for all sufficiently large $(r_1,\ldots,r_p) \in \mathbb{N}^p$. Since the total degree of the polynomial $\sum_{j=d+1}^{n}\left[\prod_{i=1}^{p}\binom{t_i+m_i}{m_i} - \prod_{i=1}^{p}\binom{t_i-a_{ji}+m_i}{m_i}\right]$ is less than m, we obtain that $a_{m_1\ldots m_p} \leq d$.

On the other hand, for all sufficiently large $(r_1,\ldots,r_p) \in \mathbb{N}^p$, $\Phi_{\eta|K}(r_1,\ldots,r_p) = \operatorname{tr.deg}_K K(\bigcup_{k=1}^{n}\Theta(r_1,\ldots,r_p)\eta_k) \geq \operatorname{tr.deg}_K K(\bigcup_{k=1}^{d}\Theta(r_1,\ldots,r_p)\eta_k) = d\prod_{i=1}^{p}\binom{r_i+m_i}{m_i}$, hence $a_{m_1\ldots m_p} \geq d$.
Thus, $a_{m_1\ldots m_p} = d = \Delta\text{-tr.deg}_K L$. □

With the notation of Theorem 3.15, let $p \geq 2$, $1 \leq k < p$, $\Delta^{(k)} = \Delta_1 \bigcup\cdots\bigcup \Delta_k$, and $F_{r_{k+1},\ldots,r_p}$ denote the $\Delta^{(k)}$-field extension of K generated by the set $\bigcup_{j=1}^{n}\Theta(0,\ldots,0,r_{k+1},\ldots,r_p)\eta_j$ ($r_i \in \mathbb{N}$), that is, $F_{r_{k+1},\ldots,r_p} = K\langle\bigcup_{j=1}^{n}\Theta_{\Delta\setminus\Delta^{(k)}}(r_{k+1},\ldots,r_p)\eta_j\rangle_{\Delta^{(k)}}$. Since $\Theta(r_1,\ldots,r_p) = \Theta(r_1,\ldots,r_k,0,\ldots,0)\Theta(0,\ldots,0,r_{k+1},\ldots,r_p)$, we can combine the results of Theorems 3.11 and 3.16 to get the following statement.

Corollary 3.17. *With the above notation, and the Δ-dimension polynomial (6) of the extension $K\langle\eta_1,\ldots,\eta_n\rangle/K$, the numerical polynomial in $p-k$ variables $\phi(t_{k+1},\ldots,t_p) =$*

$$\sum_{i_{k+1}=0}^{m_{k+1}}\cdots\sum_{i_p=0}^{m_p} a_{m_1\ldots m_k i_{k+1}\ldots i_p}\binom{t_{k+1}+i_{k+1}}{i_{k+1}}\cdots\binom{t_p+i_p}{i_p}$$

describes the growth of $\Delta^{(k)}\text{-tr.deg}_K F_{r_{k+1},\ldots,r_p}$, that is, $\phi(r_{k+1},\ldots,r_p) = \Delta^{(k)}\text{-tr.deg}_K F_{r_{k+1},\ldots,r_p}$ for all sufficiently large $r_{k+1},\ldots,r_p \in \mathbb{N}^{p-k}$.

This corollary, in particular, shows that if $L = K\langle\eta_1,\ldots,\eta_n\rangle$ is a finitely generated differential field extension of a differential field K with a basic set Δ, $\Delta' \subseteq \Delta$, $\Delta'' = \Delta \setminus \Delta'$, and $m_1 = \operatorname{Card}\Delta'$, $m_2 = \operatorname{Card}\Delta''$ ($m_1 + m_2 = m$ where $m = \operatorname{Card}\Delta$), then there exists

a univariate numerical polynomial $\phi(t) = \sum_{i=0}^{m_2} c_i \binom{t+i}{i}$ ($c_i \in \mathbb{Z}$) such that $\phi(r) = \Delta'\text{-tr. deg}_K K\langle \bigcup_{k=1}^{n} \Theta_{\Delta''}(r)\eta_k\rangle_{\Delta'}$ and $c_{m_2} = \Delta\text{-tr. deg}_K L$. Furthermore, if $\Phi_{\eta|K}(t_1, t_2) = \sum_{i=0}^{m_1}\sum_{i_2=0}^{m_2} a_{ij}\binom{t_1+i}{i}\binom{t_2+j}{j}$ is the bivariate Δ-dimension polynomial of L/K associated with the partition $\Delta = \Delta' \bigcup \Delta''$ and $d = \deg_{t_1} \Phi_{\eta|K} < m_1$, then $\Delta'\text{-t. tr. deg}_K K\langle \bigcup_{k=1}^{n} \Theta_{\Delta''}(r)\eta_k\rangle_{\Delta'} = \sum_{i=0}^{m_2} a_{dj}\binom{r+j}{j}$ (in this case $d = \Delta'\text{-type}_K K\langle \bigcup_{k=1}^{n} \Theta_{\Delta''}(r)\eta_k\rangle_{\Delta'}$).

The following theorem provides necessary and sufficient conditions on generators of a differential field extension of a given differential transcendence degree d under which the corresponding multivariate dimension polynomial has the simplest possible form.

Theorem 3.18. *With the notation of Theorem* 3.15*, the following conditions are equivalent.*

(i) $\Phi_{\eta|K}(t_1, \ldots, t_p) = d \prod_{i=1}^{p} \binom{t_i + m_i}{m_i}$.

(ii) $\Delta\text{-tr. deg}_K K\langle \eta_1, \ldots, \eta_n\rangle = \text{tr. deg}_K(\eta_1, \ldots, \eta_n) = d$.

Proof. (i) $\Rightarrow$ (ii). By Theorem 3.16, $d = \Delta\text{-tr. deg}_K L$ where $L = K\langle \eta_1, \ldots, \eta_n\rangle$. Without loss of generality we can assume that $\eta_1, \ldots, \eta_d$ is a Δ-transcendence basis of L over K. Then for all sufficiently large $(r_1, \ldots, r_p) \in \mathbb{N}^p$,
$\Phi_{\eta|K}(r_1, \ldots, r_p) = \text{tr. deg}_K K(\bigcup_{j=1}^{n} \Theta(r_1, \ldots, r_p)\eta_j) =$
$\text{tr. deg}_K K(\bigcup_{j=1}^{d} \Theta(r_1, \ldots, r_p)\eta_j)$, hence
$\text{tr. deg}_{K(\bigcup_{j=1}^{d} \Theta(r_1,\ldots,r_p)\eta_j)} K(\bigcup_{j=1}^{n} \Theta(r_1, \ldots, r_p)\eta_j) = 0$.
Therefore, every element η_j, $d+1 \le j \le n$, is algebraic over the field $F = K\langle \eta_1, \ldots, \eta_d\rangle$, so if η' denotes the $(n-d)$-tuple $(\eta_{d+1}, \ldots, \eta_n)$, then $\Phi_{\eta'|F}(t_1, \ldots, t_p) = 0$.

Let P be the defining Δ-ideal of η' in the ring of Δ-polynomials $F\{y_1, \ldots, y_{n-d}\}$. Let $\mathcal{A}$ be a characteristic set of P (we use the terminology and term orderings $<_1, \cdots <_p$ introduced in the beginning of this section). For every $j = 1, \ldots, n-d$, let E_j denote the set of all $(k_1, \ldots, k_m) \in \mathbb{N}^m$ such that $\delta_1^{k_1} \ldots \delta_m^{k_m} y_j$ is a 1-leader of an element of $\mathcal{A}$. Since $\Phi_{\eta'|F} = 0$, we also have $\omega_{\eta'|F} = 0$ where $\omega_{\eta'|F}$ is the polynomial in p variables defined in Theorem 3.11(ii). Furthermore, it follows from Theorem 2.3(iv) that $\omega_{\eta'|F} = 0$ if and only if $E_j = \{(0, \ldots, 0)\}$ for $j = 1, \ldots, n-d$.

Since $y_1 <_1 y_j$ for $j = 2. \ldots, n-d$, a Δ-polynomial in $\mathcal{A}$ with leader y_1 is a usual polynomial in y_1 with coefficients in F. Therefore, η_{d+1} and all $\theta\eta_{d+1}$ ($\theta \in \Theta$) are algebraic over F. If $\eta'' = (\eta_{d+2}, \ldots, \eta_n)$, then $\Phi_{\eta''|F}(r_1, \ldots, r_p) \le \Phi_{\eta'|F}(r_1, \ldots, r_p)$ for all $(r_1, \ldots, r_p) \in \mathbb{N}^p$, so $\Phi_{\eta''|F} = 0$ and we can repeat the above arguments and obtain that every $\theta\eta_j$ ($\theta \in \Theta, d+1 \le j \le n$) is algebraic over F.

Since the elements $\eta_{d+1}, \ldots, \eta_n$ are algebraic over the field $F = K\langle \eta_1, \ldots, \eta_d\rangle$, there exist $h_1, \ldots, h_p \in \mathbb{N}$ such that $\eta_{d+1}, \ldots, \eta_n$ are algebraic over $K(\bigcup_{j=1}^{d} \Theta(h_1, \ldots, h_p)\eta_j)$.
It follows that if $(h_1, \ldots, h_p) \le_P (r_1, \ldots, r_p)$, then the field extension $K(\bigcup_{j=1}^{n} \Theta(r_1, \ldots, r_p)\eta_j)/K(\bigcup_{j=1}^{d} \Theta(r_1, \ldots, r_p)\eta_j)$ is algebraic.

Suppose η_{d+1} is not algebraic over $K(\eta_1, \ldots, \eta_d)$. Let $q_1, \ldots, q_p$ be a minimal (with respect to the product order $<_P$) element of $\mathbb{N}^p$ such that η_{d+1} is algebraic over the field $K(\bigcup_{j=1}^{d} \Theta(q_1, \ldots, q_p)\eta_j)$. (By the assumption, $(q_1, \ldots, q_p) \ne (0, \ldots, 0)$.) Without loss of generality we can assume that $q_1 \ge 1$. Then η_{d+1} is transcendental over the field
$K(\bigcup_{j=1}^{d} \Theta(q_1 - 1, \ldots, q_p)\eta_j)$. Then there exists a term v in the ring of Δ-polynomials $K\{y_1, \ldots, y_d\}$ such that $\text{ord}_1 v = q_1$, $\text{ord}_i v \le q_i$ for $i = 2, \ldots, p$, η_{d+1} is transcendental over the field $K' = K(\{\theta\eta_j \mid \theta \in \Theta(q_1, \ldots, q_p), 1 \le j \le d, \theta y_j <_1 v\})$ and algebraic over the field $K'(v(\eta))$. It follows that $v(\eta)$ is algebraic over
$K(\bigcup_{j=1}^{d} \Theta(q_1, \ldots, q_p)\eta_j \setminus \{\eta_{d+1}\} \bigcup \{v(\eta)\})$.

Therefore, if $\theta' \in \Theta(r_1, \ldots, r_p)$ where $(h_1, \ldots, h_p) \le_P (r_1, \ldots, r_p)$, then $\theta' v(\eta)$ is algebraic over $K(\bigcup_{j=1}^{d} \Theta(r_1 + q_1, \ldots, r_p + q_p)\eta_j \setminus \{\theta'\eta_{d+1}\} \bigcup \{\theta' v(\eta)\})$.
Since η_{d+1} is algebraic over $K(\bigcup_{j=1}^{d} \Theta(q_1, \ldots, q_p)\eta_j)$, $\theta'\eta_{d+1}$ is algebraic over $K(\bigcup_{j=1}^{d} \Theta(s_1 + q_1, \ldots, s_p + q_p)\eta_j)$ where $s_i = \text{ord}_i \theta'$, $1 \le i \le p$ (clearly, $s_i \le r_i$ for $i = 1, \ldots, p$). Therefore, $\theta' v(\eta)$ is algebraic over $K(\bigcup_{j=1}^{d} \Theta(s_1 + q_1, \ldots, s_p + q_p)\eta_j \setminus \{\theta' v(\eta)\})$, hence the set $\bigcup_{j=1}^{d} \Theta(r_1 + q_1, \ldots, r_p + q_p)\eta_j$ is algebraically dependent over K that contradicts the fact that $\eta_1, \ldots, \eta_d$ are Δ-algebraically independent over K.

Thus, η_{d+1} is algebraic over $K(\eta_1, \ldots, \eta_d)$ and similarly every η_j, $d+1 \le j \le n$, is algebraic over $K(\eta_1, \ldots, \eta_d)$, so $d = \Delta\text{-tr. deg}_K K\langle \eta_1, \ldots, \eta_n\rangle = \text{tr. deg}_K(\eta_1, \ldots, \eta_n)$.

(ii) $\Rightarrow$ (i). As in the proof of Theorem 3.16, without loss of generality we can assume that $\eta_1, \ldots, \eta_d$ is a Δ-transcendence basis of the Δ-field $L = K\langle \eta_1, \ldots, \eta_n\rangle$ over K. Then the elements $\eta_1, \ldots, \eta_d$ are algebraically independent over K, so $K(\eta_1, \ldots, \eta_n)$ is an algebraic extension of $K(\eta_1, \ldots, \eta_d)$. Thus, $K(\bigcup_{j=1}^{n} \Theta(r_1, \ldots, r_p)\eta_j)$ is an algebraic extension of the field $K(\bigcup_{j=1}^{d} \Theta(r_1, \ldots, r_p)\eta_j)$ for any $(r_1, \ldots, r_p) \in \mathbb{N}^p$.
Since $\Phi_{(\eta_1,\ldots,\eta_d)|K}(t_1, \ldots, t_p) = d \prod_{i=1}^{p} \binom{t_i+m_i}{m_i}$ and
$K(\bigcup_{j=1}^{n} \Theta(r_1, \ldots, r_p)\eta_j)$ and $K(\bigcup_{j=1}^{d} \Theta(r_1, \ldots, r_p)\eta_j)$ have the same transcendence degree over K, we obtain the equality of statement (i). □

Proposition 3.19. *Let $L = K\langle \eta_1, \ldots, \eta_n\rangle$ be a Δ-field extension generated by a finite set $\eta = \{\eta_1, \ldots, \eta_n\}$ and let partition* (3) *of the set Δ be fixed. Suppose that $\Delta\text{-tr. deg}_K L = 0$ or that $\Delta\text{-tr. deg}_K L = d \ge 1$, $\eta_1, \ldots, \eta_d$ form a Δ-transcendence basis of L over K and $\eta' = \{\eta_{d+1}, \ldots, \eta_n\}$. Then*

$$\Phi_{\eta'|K\langle \eta_1,\ldots,\eta_d\rangle}(r_1, \ldots, r_p) \le \Phi_{\eta|K}(r_1, \ldots, r_p) - d \prod_{i=1}^{p} \binom{r_i + m_i}{m_i} \quad (7)$$

for all sufficiently large $(r_1, \ldots, r_p) \in \mathbb{N}^p$.

Proof. If $\Delta\text{-tr. deg}_K L = 0$, the statement is obvious. Let $d = \Delta\text{-tr. deg}_K L \ge 1$ and $\{\eta_1, \ldots, \eta_d\}$ a Δ-transcendence basis of L/K. Let $K' = K\langle \eta_1, \ldots, \eta_d\rangle$ and for any $r = (r_1, \ldots, r_p) \in \mathbb{N}^p$, $\Lambda_1(r) = \bigcup_{k=1}^{d} \Theta(r_1, \ldots, r_p)\eta_k$, $\Lambda_2(r) = \bigcup_{l=d+1}^{n} \Theta(r_1, \ldots, r_p)\eta_l$ and $\Lambda_3(r) = \Lambda_1(r) \bigcup \Lambda_2(r)$.
Then $\Phi_{\eta'|K'}(r_1, \ldots, r_p) = \text{tr. deg}_{K'} K'(\Lambda_2(r)) \le$
$\text{tr. deg}_{K(\Lambda_1(r))} K(\Lambda_3(r))\, \text{tr. deg}_{K(\Lambda_1(r))} K(\Lambda_3(r)) =$
$\text{tr. deg}_K K(\Lambda_3(r)) - \text{tr. deg}_K K(\Lambda_1(r)) = \Phi_{\eta|K}(r_1, \ldots, r_p) -$
$d \prod_{i=1}^{p} \binom{r_i+m_i}{m_i}$ for all sufficiently large $(r_1, \ldots, r_p)$. □

The next example shows that a multivariate dimension polynomial of a Δ-field extension carries essentially more information about the extension than its univariate counterpart.

Example 3.20. Let K be a differential field with a basic set of derivations $\Delta = \{\delta_1, \delta_2, \delta_3\}$ and let L be a Δ-field extension of K generated by a single Δ-generator η with the defining equation

$$\delta_1^a \delta_2^b \delta_3^c \eta + \delta_1^a \eta + \delta_2^b \eta + \delta_3^{b+c} \eta = 0 \tag{8}$$

where a, b and c are some positive integers. In other words, $L = K\langle\eta\rangle$ is Δ-isomorphic to the quotient field of the factor ring $K\{y\}/P$ where P is the linear (and therefore prime) Δ-ideal of the ring of differential (Δ-) polynomials $K\{y\}$ generated by the Δ-polynomial $f = \delta_1^a \delta_2^b \delta_3^c y + \delta_1^a y + \delta_2^b y + \delta_3^{b+c} y$. ($P$ is the defining ideal of η over K.)

By [6, Chapter II, Theorem 6], the univariate Kolchin differential dimension polynomial $\omega_{\eta/K}(t)$ of L/K is equal to the univariate dimension polynomial of the subset $\{(a, b, c)\}$ of $\mathbb{N}^3$. Using formula (2) for $p = 1$, we obtain that

$$\omega_{\eta/K}(t) = \binom{t+3}{3} - \binom{t+3-(a+b+c)}{3} =$$

$$(\frac{a+b+c}{2})t^2 + (\frac{(a+b+c)(4-a-b-c)}{2})t +$$

$$\frac{(a+b+c)[(a+b+c)^2 - 6(a+b+c) + 11]}{6}. \tag{9}$$

Now, let us fix a partition $\Delta = \Delta_1 \cup \Delta_2$ with $\Delta_1 = \{\delta_1, \delta_2\}$. Let $\Delta_2 = \{\delta_3\}$, and $\Phi_\eta(t_1, t_2)$ denote the Δ-dimension polynomial of L/K associated with this partition and the Δ-generator η. With the notation of the first part of this section, we obtain that $u_f^{(1)} = \delta_1^a \delta_2^b \delta_3^c y$ and $u_f^{(2)} = \delta_3^{b+c} y$. Using the notation of Theorem 3.11 and formula (2) we obtain that for all sufficiently large $(r_1, r_2) \in \mathbb{N}^2$, $\operatorname{Card} U'_{r_1,r_2} = \binom{r_1+2}{2}(r_2+1) - \binom{r_1+2-(a+b)}{2}(r_2+1-c)$. Expanding the last expression and using symbols t_1 and t_2 for the variables representing r_1 and r_2, respectively, we obtain the polynomial $\omega_{\eta|K}(t_1, t_2)$ (see Theorem 3.11) that describes the size of $\operatorname{Card} U'_{r_1,r_2}$: $\omega_{\eta|K}(t_1, t_2) = \frac{c}{2}t_1^2 + (a+b)t_1t_2 + \frac{2a+2b+3c-2ac-2bc}{2}t_1 + \frac{(a+b)(3-a-b)}{2}t_2 + \frac{1}{2}[(a+b-2)(a+b-1)(c-1)+2]$.

For all sufficiently large $(r_1, r_2) \in \mathbb{N}^2$, $\operatorname{Card} U''_{r_1,r_2} = \operatorname{Card}\{\delta_1^{a+k_1}\delta_2^{a+k_2}\delta_3^{a+k_3} \mid k_1, k_2, k_3 \in \mathbb{N}, k_1 + k_2 \le r_1 - (a+b), r_3 - (b+c) < k_3 \le r_3 - c\} = \binom{r_1+2-(a+b)}{2}b$. Thus, with the notation of Theorem 3.11, $\phi_{\eta|K}(t_1, t_2) = \frac{b}{2}t_1^2 + \frac{b(3-2a-2b)}{2}t_1 + \frac{b(a+b-2)(a+b-1)}{2}$. It follows that the bivariate differential dimension polynomial of the extension L/K corresponding to the partition $\Delta = \{\delta_1, \delta_2\} \bigcup \{\delta_3\}$ is

$$\Phi_{\eta|K}(t_1, t_2) = (\frac{b+c}{2})t_1^2 + (a+b)t_1t_2 + \frac{1}{2}[2a + 5b + 3c - 2ab - 2ac$$

$$-2bc - 2b^2]t_1 + \frac{(a+b)(3-a-b)}{2}t_2 +$$

$$\frac{1}{2}[(a+b-2)(a+b-1)(b+c-1)+2]. \tag{10}$$

Finally, let us fix a partition $\Delta = \Delta_1 \cup \Delta_2 \cup \Delta_3$ with $\Delta_i = \{\delta_i\}$ ($i = 1, 2, 3$). Proceeding as before (with the notation of Theorem 3.11), we obtain that $\omega_{\eta|K}(t_1, t_2, t_3) = ct_1t_2 + bt_1t_3 + at_2t_3 + (b+c-bc)t_1 + (a+c-ac)t_2 + (a+b-ab)t_3 + a + b + c - ab - ac - bc + abc$ and $\phi_{\eta|K}(t_1, t_2) = bt_1t_2 + (b-b^2)t_1 + (b-ab)t_2 + (b-ab-b^2+ab^2$, so

$$\Phi_{\eta|K}(t_1, t_2, t_3) = (b+c)t_1t_2 + bt_1t_3 + at_2t_3$$

$$+(2b + +c - bc - b^2)t_1 + (a+b+c-ab-ac)t_2 + (a+b-ab)t_3$$

$$+ (a + 2b + c - 2ab - ac - bc - b^2 + ab^2 + abc). \tag{11}$$

It follows from Theorem 3.15 that the dimension Δ-polynomial in three variables given by (11) carries four invariants of the extension L/K: the total degree 2 and the coefficients $b+c$, b and a of the terms t_1t_2, t_1t_3 and t_2t_3, respectively. The dimensional polynomial (10) carries three invariants, the total degree 2 and the coefficients $b+c$ and $a+b$, while the univariate Kolchin polynomial (9) carries only two invariants of the extension, the total degree 2 and the sum of the parameters $a+b+c$. Therefore, the Δ-dimension polynomial (11) corresponding to the partition of Δ into the union of three disjoint subsets determines all three parameters a, b and c of the defining differential equation (8) while the univariate dimension polynomial gives just the sum of the parameters.

Also, in accordance with the above considerations, the dimension polynomial (10) (with $\Delta_1 = \{\delta_1, \delta_2\}$ and $\Delta_2 = \{\delta_3\}$) shows that Δ_2-$\operatorname{tr.deg}_K K\langle\{\delta_1^{k_1}\delta_2^{k_2}\eta \mid k_1 + k_2 \le r\}\rangle_{\Delta_2} = (a+b)r + \frac{(a+b)(3-a-b)}{2}$ for all sufficiently large $r \in \mathbb{N}$.

4 ACKNOWLEDGES

This research was supported by the NSF grant CCF-1714425.

REFERENCES

[1] A. Einstein. *The Meaning of Relativity. Appendix II (Generalization of gravitation theory)*, 153–165. Princeton University Press, Princeton, NJ, 1953.

[2] J. Freitag, O. L. Sanchez, O. L.; W. Li. Effective definability of Kolchin polynomials. *Proc.Amer. Math. Soc.*, 148 (2020), 1455–1466.

[3] Joseph L. Johnson A notion on Krull dimension for differential rings. *Comment. Math. Helv.*, 44 (1969), 207–216.

[4] Joseph L. Johnson. Kähler differentials and differential algebra. *Ann. of Math.* (2), 89 (1969), 92–98.

[5] E. R. Kolchin. The notion of dimension in the theory of algebraic differential equations. *Bull. Amer. Math. Soc.*, 70 (1964), 570–573.

[6] E. R. Kolchin. *Differential Algebra and Algebraic Groups.* Academic Press, 1973.

[7] M. V. Kondrateva, A. B. Levin, A. V. Mikhalev, and E. V. Pankratev. *Differential and Difference Dimension Polynomials.* Kluwer Acad. Publ., 1999.

[8] M. Lange-Hegermann. The Differential Dimension Polynomial for Characterizable Differential Ideals. *Algorithmic and Experimental Methods in Algebra, Geometry, and Number Theory* (2018), 443–453.

[9] M. Lange-Hegermann. M. The Differential Counting Polynomial. *Foundations of Computational Mathematics*, 18, no. 2, (2018), 291–308.

[10] A. B. Levin. Multivariable dimension polynomials and new invariants of differential field extensions. *Internat. J. Math. and Math. Sci.*, 27 (2001), no. 4, 201–214.

[11] A. B. Levin. Gröbner bases with respect to several orderings and multivariable dimension polynomials. *J. Symb. Comput.*, 42 (2007), no. 5, 561–578.

[12] A. B. Levin. Dimension polynomials of intermediate fields and Krull-type dimension of finitely generated differential field extensions. *Mathematics in Computer Science*, 4 (2010), no. 2–3, 143–150.

[13] A. V. Mikhalev and E. V. Pankratev. Differential dimension polynomial of a system of differential equations. *Algebra. Collection of papers.*, 57–67. Moscow State University Press, 1980.

[14] O. L. Sanchez. Estimates for the coefficients of differential dimension polynomials. *Mathematics of Computation*, 88 (2019), 2959–2985.

[15] W. Sit. On the differential transcendence polynomials of finitely generated differential field extensions. *Amer. J. Math.*, 101 (1979), no. 6, 1249–1263.

Further Results on the Factorization and Equivalence for Multivariate Polynomial Matrices

Dong Lu
[1]Beijing Advanced Innovation Center for Big Data and Brain Computing, Beihang University
Beijing 100191, China
[2]School of Mathematical Sciences, Beihang University
Beijing 100191, China
donglu@buaa.edu.cn

Dingkang Wang
[1]KLMM, Academy of Mathematics and Systems Science, Chinese Academy of Sciences,
Beijing 100190, China
[2]School of Mathematical Sciences, University of Chinese Academy of Sciences, Beijing 100049, China
dwang@mmrc.iss.ac.cn

Fanghui Xiao
[1]KLMM, Academy of Mathematics and Systems Science, Chinese Academy of Sciences,
Beijing 100190, China
[2]School of Mathematical Sciences, University of Chinese Academy of Sciences, Beijing 100049, China
xiaofanghui@amss.ac.cn

ABSTRACT

This paper is concerned with the factorization and equivalence problems of multivariate polynomial matrices. We present a new criterion for the existence of matrix factorizations for a class of multivariate polynomial matrices, and prove that these matrix factorizations are unique. Based on this new criterion and the constructive proof process, we give an algorithm to compute a matrix factorization of a multivariate polynomial matrix. After that, we put forward a sufficient and necessary condition for the equivalence of square polynomial matrices: a square polynomial matrix is equivalent to a diagonal triangle if it satisfies the condition. An illustrative example is given to show the effectiveness of the matrix equivalence theorem.

CCS CONCEPTS

• **Computing methodologies** → **Symbolic and algebraic algorithms**; **Algebraic algorithms**.

KEYWORDS

Polynomial matrices, Matrix factorization, Matrix equivalence, Minors, Gröbner basis

ACM Reference Format:
Dong Lu, Dingkang Wang, and Fanghui Xiao. 2020. Further Results on the Factorization and Equivalence for Multivariate Polynomial Matrices. In *International Symposium on Symbolic and Algebraic Computation (ISSAC '20), July 20–23, 2020, Kalamata, Greece.* ACM, New York, NY, USA, 8 pages. https://doi.org/10.1145/3373207.3404020

1 INTRODUCTION

Multidimensional systems have wide applications in image, signal processing, and other areas (see, e.g., [1, 2]). A multidimensional system may be represented by a multivariate polynomial matrix, and we can obtain some important properties of the system by studying the matrix. Therefore, the factorization problem and the equivalence problem related to multivariate polynomial matrices have attracted much attention over the past decades.

Up to now, the factorization problem for univariate and bivariate polynomial matrices has been completely solved by [23, 41, 46], but the case of more than two variables is still open. In [60], Youla and Gnavi first introduced three important concepts according to different properties of polynomial matrices, namely zero prime matrix factorization, minor prime matrix factorization and factor prime matrix factorization. Based on the work of [60] on basic structures of multivariate polynomial matrices, the factorization problem for multivariate (more than two variables) polynomial matrices has made great progress.

When multivariate polynomial matrices satisfy several special properties, there are some results about the existence problem of zero prime matrix factorizations for the polynomial matrices (see, e.g., [8, 31, 33]). After that, Lin and Bose in [34] proposed the famous Lin-Bose conjecture: a multivariate polynomial matrix admits a zero prime matrix factorization if all its maximal reduced minors generate a unit ideal. This conjecture was proved by Liu et al. [39], Pommaret [48], Wang and Feng [58], respectively. Wang and Kwong in [59] gave a sufficient and necessary condition for a multivariate polynomial with full row (column) rank to have a minor prime matrix factorization. They extracted an algorithm from Pommaret's proof of the Lin-Bose conjecture, and examples showed the effectiveness of the algorithm. Guan et al. in [22] generalized the main results in [59] to the case of polynomial matrices without full row (column) rank. For the existence problem of factor prime matrix factorizations for multivariate polynomial matrices with full row (column) rank, Wang and Liu have achieved some important results (see, e.g., [40, 56]). Then Guan et al. in [21] gave an algorithm to decide whether a class of polynomial matrices has a factor prime matrix factorization. However, the existence problem of factor prime matrix factorizations for multivariate polynomial matrices remains a challenging open problem so far.

Comparing to the factorization problem of multivariate polynomial matrices which has been widely investigated during the past years, less attention has been paid to the equivalence problem of multivariate polynomial matrices. For any given multidimensional system, our goal is to simplify it into a simpler equivalent form.

ISSAC '20, July 20–23, 2020, Kalamata, Greece

ACM ISBN 978-1-4503-7100-1/20/07…$15.00
https://doi.org/10.1145/3373207.3404020

Since a univariate polynomial ring is a principal ideal domain, a univariate polynomial matrix is always equivalent to its Smith form. This implies that the equivalence problem has been solved [24, 51]. For any given bivariate polynomial matrix, conditions under which it is equivalent to its Smith form have been investigated in [18, 19, 26]. Note that the equivalence problem of two multivariate polynomial matrices is equivalent to the isomorphism problem for two finitely presented modules, Boudellioua and Quadrat [6] and Cluzeau and Quadrat [9–11] obtained some important results by using module theory and homological algebra. According to the previous works in [6], Boudellioua in [3, 5] designed some algorithms based on Maple to compute Smith forms for some classes of multivariate polynomial matrices. For the case of multivariate polynomial matrices with more than one variable, however, the equivalence problem is not yet fully solved due to the lack of a mature polynomial matrix theory (see, e.g., [25, 46, 49]).

From our personal viewpoint, new ideas need to be injected into these areas to obtain new theoretical results and effective algorithms. Therefore, it would be significant to provide some new criteria to study the factorization problem and the equivalence problem for some classes of multivariate polynomial matrices.

From the 1990s to the present, there is a class of multivariate polynomial matrices that has always attracted attention. That is,

$$\mathcal{M} = \{\mathbf{F} \in k[\mathbf{z}]^{l\times m} : (z_1 - f(\mathbf{z}_2)) \mid d_l(\mathbf{F}) \text{ with } f(\mathbf{z}_2) \in k[\mathbf{z}_2]\},$$

where $\mathbf{z}_2 = \{z_2, \ldots, z_n\}$ and $d_l(\mathbf{F})$ is the GCD of all the $l\times l$ minors of $\mathbf{F}$. Many people tried to solve the factorization problem and the equivalence problem of multivariate polynomial matrices in $\mathcal{M}$.

Let $\mathbf{F} \in \mathcal{M}$ and $h = z_1 - f(\mathbf{z}_2)$. Lin and coauthors presented some criteria on the existence problem of a matrix factorization for $\mathbf{F}$ w.r.t. h (see, e.g., [29, 30, 36, 37]). Moreover, Lin et al. in [37] proposed a constructive algorithm to factorize $\mathbf{F}$ w.r.t. h. When $d_l(\mathbf{F}) = h$, Wang [57] gave a new result for $\mathbf{F}$ to have a minor prime matrix factorization using methods from computer algebra. Based on the pioneering work of Lin et al., Liu et al. [38] and Lu et al. [44, 45] obtained some new criteria for factorizing $\mathbf{F}$ w.r.t. h. When $l = m$ and $\det(\mathbf{F}) = h$, Lin et al. [35] proved that $\mathbf{F}$ is equivalent to the diagonal triangle $\mathrm{diag}(1, \ldots, 1, h)$. After that, Li et al. [27] generalized the main results in [35] to the case of $\det(\mathbf{F}) = h^q$.

Through research, we find that there are still many multivariate polynomial matrices in $\mathcal{M}$ which do not satisfy previous results and can be factorized or are equivalent to some diagonal triangles. As a consequence, we continue to study the factorization problem and the equivalence problem of multivariate polynomial matrices in $\mathcal{M}$ in this paper.

The rest of the paper is organized as follows. After a brief introduction to matrix factorization and matrix equivalence in Section 2, we use two examples to propose two problems that we shall consider. We present in Section 3 a new criterion for factorizing $\mathbf{F}$ w.r.t. h, then we study the uniqueness of the matrix factorization and construct an algorithm to factorize $\mathbf{F}$. A sufficient and necessary condition for a square multivariate polynomial matrix being equivalent to a diagonal triangle is described in Section 4, and we use an example to illustrate the effectiveness of the new matrix equivalence theorem. The paper contains a summary of contributions and some remarks in Section 5.

2 PRELIMINARIES AND PROBLEMS

In this section we first recall some basic notions which will be used in the following sections. For those notions which are not formally introduced in the paper, the reader may consult the references [27, 37, 38, 45]. And then, we use two examples to put forward two problems that we are considering.

2.1 Basic Notions

We denote by k an algebraically closed field, $\mathbf{z}$ the n variables $z_1, z_2, \ldots, z_n$, $\mathbf{z}_2$ the $(n-1)$ variables $z_2, \ldots, z_n$, where $n \geq 3$. Let $k[\mathbf{z}]$ and $k[\mathbf{z}_2]$ be the ring of polynomials in variables $\mathbf{z}$ and $\mathbf{z}_2$ with coefficients in k, respectively. Let $k[\mathbf{z}]^{l\times m}$ be the set of $l \times m$ matrices with entries in $k[\mathbf{z}]$. Without loss of generality, we assume that $l \leq m$, and for convenience we use uppercase bold letters to denote polynomial matrices. In addition, "w.r.t." and "GCD" stand for "with respect to" and "greatest common divisor", respectively.

Let $\mathbf{F} \in k[\mathbf{z}]^{l\times m}$ and $f \in k[\mathbf{z}_2]$, then $\mathbf{F}(f, \mathbf{z}_2)$ denotes a polynomial matrix in $k[\mathbf{z}_2]^{l\times m}$ which is formed by transforming z_1 in $\mathbf{F}$ into f. Moreover, $\mathbf{F}^{\mathrm{T}}$ represents the transposed matrix of $\mathbf{F}$. Throughout the paper, we use $d_i(\mathbf{F})$ to denote the GCD of all the $i \times i$ minors of $\mathbf{F}$ with the convention that $d_0(\mathbf{F}) = 1$, where $i = 1, \ldots, l$. Assume that $f_1, \ldots, f_s \in k[\mathbf{z}]$, we use $\langle f_1, \ldots, f_s\rangle$ to denote the ideal generated by $f_1, \ldots, f_s$ in $k[\mathbf{z}]$. Let $g, h \in k[\mathbf{z}]$, then $g \mid h$ means that g is a divisor of h.

The following concepts are from multidimensional systems theory.

Definition 2.1 ([28, 54]). Let $\mathbf{F} \in k[\mathbf{z}]^{l\times m}$ be of full row rank. For any given integer i with $1 \leq i \leq l$, let $a_1, \ldots, a_\beta$ denote all the $i\times i$ minors of $\mathbf{F}$, where $\beta = \binom{l}{i} \cdot \binom{m}{i}$. Extracting $d_i(\mathbf{F})$ from $a_1, \ldots, a_\beta$ yields

$$a_j = d_i(\mathbf{F}) \cdot b_j,\ j = 1, \ldots, \beta,$$

where $b_1, \ldots, b_\beta$ are called all the $i \times i$ reduced minors of $\mathbf{F}$.

Definition 2.2 ([60]). Let $\mathbf{F} \in k[\mathbf{z}]^{l\times m}$ be of full row rank.

(1) If all the $l \times l$ minors of $\mathbf{F}$ generate $k[\mathbf{z}]$, then $\mathbf{F}$ is said to be a zero left prime (ZLP) matrix.
(2) If all the $l \times l$ minors of $\mathbf{F}$ are relatively prime, i.e., $d_l(\mathbf{F})$ is a nonzero constant in k, then $\mathbf{F}$ is said to be a minor left prime (MLP) matrix.
(3) If for any polynomial matrix factorization $\mathbf{F} = \mathbf{F}_1\mathbf{F}_2$ with $\mathbf{F}_1 \in k[\mathbf{z}]^{l\times l}$, $\mathbf{F}_1$ is necessarily a unimodular matrix, i.e., $\det(\mathbf{F}_1)$ is a nonzero constant in k, then $\mathbf{F}$ is said to be a factor left prime (FLP) matrix.

Zero right prime (ZRP) matrices, minor right prime (MRP) matrices and factor right prime (FRP) matrices can be similarly defined for matrices $\mathbf{F} \in k[\mathbf{z}]^{m\times l}$ with $m \geq l$. We refer to [60] for more details about the relationship among ZLP matrices, MLP matrices and FLP matrices.

For any given ZLP matrix $\mathbf{F} \in k[\mathbf{z}]^{l\times m}$, Quillen [50] and Suslin [55] proved that an $m \times m$ unimodular matrix can be constructed such that $\mathbf{F}$ is its first l rows, respectively. This result is called Quillen-Suslin theorem, and it solved the question raised by Serre in [52].

Lemma 2.3 ([50, 55]). *If $\mathbf{F} \in k[\mathbf{z}]^{l\times m}$ is a ZLP matrix, then a unimodular matrix $\mathbf{U} \in k[\mathbf{z}]^{m\times m}$ can be constructed such that $\mathbf{F}$ is its first l rows.*

There are many algorithms for the Quillen-Suslin theorem, we refer to [43, 47, 61] for more details. In [16], Fabiańska and Quadrat first designed a Maple package, which is called QUILLENSUSLIN [17], to implement the Quillen-Suslin theorem.

Let W be a $k[\mathbf{z}]$-module generated by $\vec{u}_1, \ldots, \vec{u}_l \in k[\mathbf{z}]^{1\times m}$. The set of all $(b_1, \ldots, b_l) \in k[\mathbf{z}]^{1\times l}$ such that $b_1\vec{u}_1 + \cdots + b_l\vec{u}_l = \vec{0}$ is a $k[\mathbf{z}]$-module of $k[\mathbf{z}]^{1\times l}$, is called the (first) syzygy module of W, and denoted by $\mathrm{Syz}(W)$. Lin in [32] proposed several interesting structural properties of syzygy modules. Let $\mathbf{F} = \left[\vec{u}_1^{\mathrm{T}}, \ldots, \vec{u}_l^{\mathrm{T}}\right]^{\mathrm{T}}$. The rank of W is defined as the rank of $\mathbf{F}$ that is denoted by $\mathrm{rank}(\mathbf{F})$. Guan et al. in [21] proved that the rank of W does not depend on the choice of generators of W.

Lemma 2.4. *With above notations. If $\mathrm{rank}(W) = r$ with $1 \le r \le l$, then the rank of $\mathrm{Syz}(W)$ is $l - r$.*

Proof. Let $k(\mathbf{z})$ be the fraction field of $k[\mathbf{z}]$, and $\mathrm{Syz}^*(W) = \{\vec{v} \in k(\mathbf{z})^{1\times l} : \vec{v}\cdot\mathbf{F} = \vec{0}\}$. Then, $\mathrm{Syz}^*(W)$ is a $k(\mathbf{z})$-vector space of dimension $l - r$. For any given $l - r + 1$ different vectors $\vec{v}_1, \ldots, \vec{v}_{l-r+1} \in k[\mathbf{z}]^{1\times l}$ in $\mathrm{Syz}(W)$, it is obvious that $\vec{v}_i \in \mathrm{Syz}^*(W)$ for each i, and they are $k(\mathbf{z})$-linearly dependent. This implies that $\vec{v}_1, \ldots, \vec{v}_{l-r+1}$ are $k[\mathbf{z}]$-linearly dependent. Thus $\mathrm{rank}(\mathrm{Syz}(W)) \le l - r$.

Assume that $\vec{p}_1, \ldots, \vec{p}_{l-r} \in k(\mathbf{z})^{1\times l}$ are $l-r$ vectors in $\mathrm{Syz}^*(W)$, and they are $k(\mathbf{z})$-linearly independent. For each j, we have

$$p_{j1}\vec{u}_1 + \cdots + p_{jl}\vec{u}_l = \vec{0}, \tag{1}$$

where $\vec{p}_j = (p_{j1}, \ldots, p_{jl})$. Multiplying both sides of Equation (1) by the least common multiple of the denominators of $p_{j1}, \ldots, p_{jl}$, we obtain $\bar{p}_j = (\bar{p}_{j1}, \ldots, \bar{p}_{jl}) \in k[\mathbf{z}]$ such that $\bar{p}_{j1}\vec{u}_1 + \cdots + \bar{p}_{jl}\vec{u}_l = \vec{0}$. Then, $\bar{p}_j \in \mathrm{Syz}(W)$, where $j = 1, \ldots, l - r$. Moreover, $\bar{p}_1, \ldots, \bar{p}_{l-r}$ are $k[\mathbf{z}]$-linearly independent. Thus, $\mathrm{rank}(\mathrm{Syz}(W)) \ge l - r$.

As a consequence, the rank of $\mathrm{Syz}(W)$ is $l - r$ and the proof is completed. □

Remark 1. *Assume that $\mathrm{Syz}(W)$ is generated by $\vec{v}_1, \ldots, \vec{v}_t \in k[\mathbf{z}]^{1\times l}$, and $\mathbf{H} = \left[\vec{v}_1^{\mathrm{T}}, \ldots, \vec{v}_t^{\mathrm{T}}\right]^{\mathrm{T}}$. It follows from $\mathrm{rank}(\mathbf{H}) = l - r$ that $t \ge l - r$. That is, the number of vectors in any given generators of $\mathrm{Syz}(W)$ is greater than or equal to $l - r$.*

Definition 2.5 ([7]). Let $\mathbf{F} \in k[\mathbf{z}]^{l\times m}$. For each $1 \le i \le l$, the ideal generated by all the $i\times i$ minors of $\mathbf{F}$ is called the i-th determinantal ideal of $\mathbf{F}$, and denoted by $I_i(\mathbf{F})$. For convenience, let $I_0(\mathbf{F}) = k[\mathbf{z}]$.

Definition 2.6 ([15]). Let W be a finitely generated $k[\mathbf{z}]$-module, and $k[\mathbf{z}]^{1\times l} \xrightarrow{\phi} k[\mathbf{z}]^{1\times m} \to W \to 0$ be a presentation of W, where ϕ acts on the right on row vectors, i.e., $\phi(\vec{u}) = \vec{u}\cdot\mathbf{F}$ for $\vec{u} \in k[\mathbf{z}]^{1\times l}$ with $\mathbf{F}$ being a presentation matrix corresponding to the linear mapping ϕ. Then the ideal $Fitt_j(W) = I_{m-j}(\mathbf{F})$ is called the j-th Fitting ideal of W. Here, we make the convention that $Fitt_j(W) = k[\mathbf{z}]$ for $j \ge m$, and that $Fitt_j(W) = 0$ for $j < \max\{m - l, 0\}$.

We remark that $Fitt_j(W)$ only depend on W (see, e.g., [15, 20]). In addition, the chain

$$0 = Fitt_{-1}(W) \subseteq Fitt_0(W) \subseteq \ldots \subseteq Fitt_m(W) = k[\mathbf{z}]$$

of Fitting ideals is increasing. We can use SINGULAR procedures to compute Fitting ideals of modules [13, 14]. Cox et al. in [12] showed that one obtains the presentation matrix $\mathbf{F}$ for W by arranging the generators of $\mathrm{Syz}(W)$ as rows. We denote the submodule of $k[\mathbf{z}]^{1\times m}$ generated by all the row vectors of $\mathbf{F}$ by $\mathrm{Im}(\mathbf{F})$, then $\mathrm{Im}(\mathbf{F}) = \mathrm{Syz}(W)$.

2.2 Matrix Factorization Problem

A matrix factorization of a multivariate polynomial matrix is formulated as follows.

Definition 2.7. Let $\mathbf{F} \in k[\mathbf{z}]^{l\times m}$ and $h_0 \mid d_l(\mathbf{F})$. $\mathbf{F}$ is said to admit a matrix factorization w.r.t. h_0 if $\mathbf{F}$ can be factorized as

$$\mathbf{F} = \mathbf{G}_1\mathbf{F}_1 \tag{2}$$

such that $\mathbf{G}_1 \in k[\mathbf{z}]^{l\times l}$, $\mathbf{F}_1 \in k[\mathbf{z}]^{l\times m}$, and $\det(\mathbf{G}_1) = h_0$. In particular, Equation (2) is said to be a ZLP (MLP, FLP) matrix factorization if $\mathbf{F}_1$ is a ZLP (MLP, FLP) matrix.

Throughout the paper, let $h = z_1 - f(\mathbf{z}_2)$ with $f(\mathbf{z}_2) \in k[\mathbf{z}_2]$. Combining Definition 2.7 and the type of polynomial matrices we mentioned in Section 1, this paper will address the following specific matrix factorization problem.

Problem 1. *Let $\mathbf{F} \in \mathcal{M}$. Under what condition does $\mathbf{F}$ have a matrix factorization w.r.t. h.*

So far, some results have been made on Problem 1, and the latest progress on this problem was obtained by Lu et al. [45].

Lemma 2.8 ([45]). *Let $\mathbf{F} \in \mathcal{M}$. If $h \nmid d_{l-1}(\mathbf{F})$ and the ideal generated by h and all the $(l-1)\times(l-1)$ reduced minors of $\mathbf{F}$ is $k[\mathbf{z}]$, then $\mathbf{F}$ admits a matrix factorization w.r.t. h.*

Although Lemma 2.8 gives a criterion to determine whether $\mathbf{F}$ has a matrix factorization w.r.t. h, we found that there exist some polynomial matrices in $\mathcal{M}$ which do not satisfy the conditions of Lemma 2.8, but still admit matrix factorizations w.r.t. h. Now, we use an example to illustrate this situation.

Example 2.9. Let

$$\mathbf{F} = \begin{bmatrix} \mathbf{F}[1,1] & z_1^3 - z_2^3 - z_1^2z_3 + z_2z_3^2 & z_1z_2 - z_2z_3 & z_2^2 \\ -z_1z_2 + z_3^2 & -z_2^2 + z_1z_3 & 0 & z_2 \end{bmatrix}$$

be a polynomial matrix in $\mathbb{C}[z_1, z_2, z_3]^{2\times 4}$, where $\mathbf{F}[1,1] = -2z_1z_2^2 + z_1^2z_3 + z_2^2z_3 - z_1z_3^2 + z_2z_3^2$ and $\mathbb{C}$ is the complex field.

It is easy to compute that $d_2(\mathbf{F}) = z_2(z_1 - z_3)$ and $d_1(\mathbf{F}) = 1$. Let $h = z_1 - z_3$, then $h \mid d_2(\mathbf{F})$ implies that $\mathbf{F} \in \mathcal{M}$. Obviously, $h \nmid d_1(\mathbf{F})$. Since $d_1(\mathbf{F}) = 1$, the entries in $\mathbf{F}$ are all the 1×1 reduced minors of $\mathbf{F}$. Let $\prec_{\mathbf{z}}$ be the degree reverse lexicographic order, then the reduced Gröbner basis G of the ideal generated by h and all the 1×1 reduced minors of $\mathbf{F}$ w.r.t. $\prec_{\mathbf{z}}$ is $\{z_1 - z_3, z_2, z_3^2\}$. It follows from $G \ne \{1\}$ that Lemma 2.8 cannot be applied. However, $\mathbf{F}$ has a matrix factorization w.r.t. h, i.e., there exist polynomial matrices $\mathbf{G}_1 \in \mathbb{C}[z_1, z_2, z_3]^{2\times 2}$ and $\mathbf{F}_1 \in \mathbb{C}[z_1, z_2, z_3]^{2\times 4}$ such that

$$\mathbf{F} = \mathbf{G}_1\mathbf{F}_1 = \begin{bmatrix} h & z_2 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} z_1z_3 - z_2^2 & z_1^2 - z_2z_3 & z_2 & 0 \\ -z_1z_2 + z_3^2 & -z_2^2 + z_1z_3 & 0 & z_2 \end{bmatrix},$$

where $\det(\mathbf{G}_1) = h$.

From the above example we see that Problem 1 is far from being resolved. So, in the next section we make a detailed analysis on this problem.

2.3 Matrix Equivalence Problem

Now we introduce the concept of the equivalence of two multivariate polynomial matrices.

Definition 2.10. Two polynomial matrices $\mathbf{F}_1 \in k[\mathbf{z}]^{l\times m}$ and $\mathbf{F}_2 \in k[\mathbf{z}]^{l\times m}$ are said to be equivalent if there exist two unimodular matrices $\mathbf{U} \in k[\mathbf{z}]^{l\times l}$ and $\mathbf{V} \in k[\mathbf{z}]^{m\times m}$ such that

$$\mathbf{F}_1 = \mathbf{U}\mathbf{F}_2\mathbf{V}. \tag{3}$$

In fact, a univariate polynomial matrix is equivalent to its Smith form. However, this result is not valid for the case of more than one variable, and there are many counter-examples (see, e.g., [4, 26]). Hence, people began to consider under what conditions multivariate polynomial matrices in $k[\mathbf{z}]$ with $n \geq 2$ are equivalent to simpler forms. In [27], Li et al. investigated the equivalence problem of a class of multivariate polynomial matrices and obtained the following result.

Lemma 2.11 ([27]). *Let $\mathbf{F} \in k[\mathbf{z}]^{l\times l}$ with $\det(\mathbf{F}) = h^q$, where $h = z_1 - f(\mathbf{z}_2)$ and q is a positive integer. Then $\mathbf{F}$ is equivalent to $\mathrm{diag}(1,\ldots,1,h^q)$ if and only if h^q and all the $(l-1)\times(l-1)$ minors of $\mathbf{F}$ generate $k[\mathbf{z}]$.*

For a given matrix that does not satisfy the condition of Lemma 2.11, we use the following example to illustrate that it can be equivalent to another diagonal triangle.

Example 2.12. Let $\mathbf{F} \in \mathbb{C}[z_1,z_2,z_3]^{3\times3}$ with $\mathbb{C}$ being the complex field, where

$$\begin{cases}
\mathbf{F}[1,1] = z_1z_2 - z_2^2 + z_2z_3 + z_2 - z_3 - 1,\\
\mathbf{F}[1,2] = z_1z_2z_3 - z_2^2z_3 + z_1z_2 - z_2^2 + z_2z_3 - z_3,\\
\mathbf{F}[1,3] = z_1z_2z_3 - z_2^2z_3,\\
\mathbf{F}[2,1] = z_1z_2 - z_2^2 + z_1 - z_2 + z_3 + 1,\\
\mathbf{F}[2,2] = (z_1 - z_2)(z_2z_3 + 2z_2 + z_3 + 1) + z_3,\\
\mathbf{F}[2,3] = z_1z_2z_3 - z_2^2z_3 + z_1z_2 - z_2^2 + z_1z_3 - z_2z_3,\\
\mathbf{F}[3,1] = z_1 - z_2,\\
\mathbf{F}[3,2] = z_1z_3 - z_2z_3 + 2z_1 - 2z_2,\\
\mathbf{F}[3,3] = z_1z_3 - z_2z_3 + z_1 - z_2.
\end{cases}$$

It is easy to compute that $\det(\mathbf{F}) = (z_1 - z_2)^2$. Let $h = z_1 - z_2$ and $\prec_{\mathbf{z}}$ be the degree reverse lexicographic order, then the reduced Gröbner basis G of the ideal generated by h^2 and all the 2×2 minors of $\mathbf{F}$ w.r.t. $\prec_{\mathbf{z}}$ is $\{z_1 - z_2\}$. It follows from $G \neq \{1\}$ that Lemma 2.11 cannot be applied. However, $\mathbf{F}$ is equivalent to $\mathrm{diag}(1,h,h)$, i.e., there exist two unimodular polynomial matrices $\mathbf{U} \in \mathbb{C}[z_1,z_2,z_3]^{3\times3}$ and $\mathbf{V} \in \mathbb{C}[z_1,z_2,z_3]^{3\times3}$ such that $\mathbf{F} = \mathbf{U}\cdot\mathrm{diag}(1,h,h)\cdot\mathbf{V} =$

$$\begin{bmatrix} z_2-1 & z_2 & 0\\ 1 & z_2+1 & z_2\\ 0 & 1 & 1\end{bmatrix}\begin{bmatrix}1&0&0\\0&h&0\\0&0&h\end{bmatrix}\begin{bmatrix} z_3+1 & z_3 & 0\\ 1 & z_3+1 & z_3\\ 0 & 1 & 1\end{bmatrix}.$$

Based on the phenomenon of Example 2.12, we consider the following matrix equivalence problem in this paper.

Problem 2. *Let $\mathbf{F} \in k[\mathbf{z}]^{l\times l}$ with $\det(\mathbf{F}) = h^r$, where $h = z_1 - f(\mathbf{z}_2)$ and $1 \leq r \leq l$. What is the sufficient and necessary condition for the equivalence of $\mathbf{F}$ and $\mathrm{diag}(\underbrace{1,\ldots,1}_{l-r},\underbrace{h,\ldots,h}_{r})$?*

3 FACTORIZATION FOR POLYNOMIAL MATRICES

In this section, we first propose a new criterion to judge whether $\mathbf{F} \in \mathcal{M}$ has a matrix factorization w.r.t. h, and then we study the uniqueness of this matrix factorization. Based on the constructive algorithm proposed by Lin et al. [37] and the new criterion, we finally present a polynomial matrix factorization algorithm and use a non-trivial example to demonstrate the detailed process of the algorithm.

3.1 Matrix Factorization Theorem

We first introduce an important result, which is an answer to the generalized Serre problem proposed by Lin and Bose [31, 34].

Lemma 3.1 ([58]). *Let $\mathbf{F} \in k[\mathbf{z}]^{l\times m}$ with $\mathrm{rank}(\mathbf{F}) = r$, and all the $r\times r$ reduced minors of $\mathbf{F}$ generate $k[\mathbf{z}]$. Then there exist $\mathbf{G}_1 \in k[\mathbf{z}]^{l\times r}$ and $\mathbf{F}_1 \in k[\mathbf{z}]^{r\times m}$ such that $\mathbf{F} = \mathbf{G}_1\mathbf{F}_1$ with $\mathbf{F}_1$ being a ZLP matrix.*

Remark 2. *Since $\mathrm{rank}(\mathbf{F}) \leq \min\{\mathrm{rank}(\mathbf{G}_1), \mathrm{rank}(\mathbf{F}_1)\}$, we have $\mathrm{rank}(\mathbf{G}_1) = r$ in Lemma 3.1. This implies that $\mathbf{G}_1$ is a polynomial matrix with full column rank.*

Lemma 3.2 ([37]). *Let $p \in k[\mathbf{z}]$ and $f(\mathbf{z}_2) \in k[\mathbf{z}_2]$. If $p(f,\mathbf{z}_2)$ is a zero polynomial in $k[\mathbf{z}_2]$, then $(z_1 - f(\mathbf{z}_2))$ is a divisor of p.*

Now, we propose a new criterion to solve Problem 1.

Theorem 3.3. *Let $\mathbf{F} \in \mathcal{M}$ and $W = \mathrm{Im}(\mathbf{F}(f,\mathbf{z}_2))$. If $Fitt_{l-2}(W) = 0$ and $Fitt_{l-1}(W) = \langle d\rangle$ with $d \in k[\mathbf{z}_2]\setminus\{0\}$, then $\mathbf{F}$ admits a matrix factorization w.r.t. h.*

Proof. Let $k[\mathbf{z}_2]^{1\times s} \xrightarrow{\phi} k[\mathbf{z}_2]^{1\times l} \to W \to 0$ be a presentation of W, and $\mathbf{H} \in k[\mathbf{z}_2]^{s\times l}$ be a matrix corresponding to the linear mapping ϕ. Then $\mathrm{Syz}(W) = \mathrm{Im}(\mathbf{H})$.

It follows from $Fitt_{l-2}(W) = 0$ that all the 2×2 minors of $\mathbf{H}$ are zero polynomials. Then, $\mathrm{rank}(\mathbf{H}) \leq 1$. Moreover, $Fitt_{l-1}(W) = \langle d\rangle$ with $d \in k[\mathbf{z}_2]\setminus\{0\}$ implies that $\mathrm{rank}(\mathbf{H}) \geq 1$. As a consequence, we have $\mathrm{rank}(\mathbf{H}) = 1$.

Let $a_1,\ldots,a_\beta \in k[\mathbf{z}_2]$ and $b_1,\ldots,b_\beta \in k[\mathbf{z}_2]$ be all the 1×1 minors and reduced minors of $\mathbf{H}$, respectively. Then, $a_i = d_1(\mathbf{H})\cdot b_i$ for $i = 1,\ldots,\beta$. Since $\langle a_1,\ldots,a_\beta\rangle = \langle d\rangle$, it is obvious that $d \mid d_1(\mathbf{H})$. Moreover, we have $d = \sum_{i=1}^{\beta} c_ia_i$ for some $c_i \in k[\mathbf{z}_2]$. Thus $d = d_1(\mathbf{H})\cdot(\sum_{i=1}^{\beta} c_ib_i)$. This implies that $d_1(\mathbf{H}) \mid d$. Hence $d = \delta\cdot d_1(\mathbf{H})$, where δ is a nonzero constant. Therefore, $\langle b_1,\ldots,b_\beta\rangle = k[\mathbf{z}_2]$.

According to Lemma 3.1, there exist $\mathbf{G} \in k[\mathbf{z}_2]^{s\times1}$ and $\mathbf{H}_1 \in k[\mathbf{z}_2]^{1\times l}$ such that $\mathbf{H} = \mathbf{G}\mathbf{H}_1$ with $\mathbf{H}_1$ being a ZLP matrix. It follows from $\mathrm{Syz}(W) = \mathrm{Im}(\mathbf{H})$ that $\mathbf{G}\mathbf{H}_1\mathbf{F}(f,\mathbf{z}_2) = \mathbf{0}_{s\times m}$. Since $\mathbf{G}$ is a matrix with full column rank, we have $\mathbf{H}_1\mathbf{F}(f,\mathbf{z}_2) = \mathbf{0}_{1\times m}$.

Using the Quillen-Suslin theorem, we can construct a unimodular matrix $\mathbf{U} \in k[\mathbf{z}_2]^{l\times l}$ such that $\mathbf{H}_1$ is its first row. Let $\mathbf{F}_0 = \mathbf{U}\mathbf{F}$, then the first row of $\mathbf{F}_0(f,\mathbf{z}_2) = \mathbf{U}\mathbf{F}(f,\mathbf{z}_2)$ is zero vector. By

Lemma 3.2, h is a common divisor of the polynomials in the first row of $\mathbf{F}_0$, thus

$$\mathbf{F}_0 = \mathbf{U}\mathbf{F} = \mathbf{D}\mathbf{F}_1 = \mathrm{diag}(h, \underbrace{1, \ldots, 1}_{l-1}) \cdot \begin{bmatrix} \bar{f}_{11} & \bar{f}_{12} & \cdots & \bar{f}_{1m} \\ \vdots & \vdots & \vdots & \vdots \\ \bar{f}_{l1} & \bar{f}_{l2} & \cdots & \bar{f}_{lm} \end{bmatrix}.$$

Consequently, we can now derive the matrix factorization of $\mathbf{F}$ w.r.t. h:

$$\mathbf{F} = \mathbf{G}_1\mathbf{F}_1,$$

where $\mathbf{G}_1 = \mathbf{U}^{-1}\mathbf{D} \in k[\mathbf{z}]^{l\times l}$, $\mathbf{F}_1 \in k[\mathbf{z}]^{l\times m}$ and $\det(\mathbf{G}_1) = h$. □

According to the proof of Theorem 3.3, it is easy to get a more general result below.

Theorem 3.4. *Let $\mathbf{F} \in \mathcal{M}$ and $W = \mathrm{Im}(\mathbf{F}(f, \mathbf{z}_2))$. If $Fitt_{r-1}(W) = 0$ and $Fitt_r(W) = \langle d\rangle$ with $d \in k[\mathbf{z}_2] \setminus \{0\}$ and $0 \le r \le l-1$, then $\mathbf{F}$ admits a matrix factorization w.r.t. h^{l-r}.*

Remark 3. *In Theorem 3.4, it follows from $Fitt_{r-1}(W) = 0$ and $Fitt_r(W) = \langle d\rangle$ that $\mathrm{rank}(\mathbf{H}) = l - r$, where $\mathrm{Syz}(W) = \mathrm{Im}(\mathbf{H})$. Based on Lemma 2.4, we have $\mathrm{rank}(\mathbf{F}(f, \mathbf{z}_2)) = \mathrm{rank}(W) = r$. $\mathbf{F} \in \mathcal{M}$ implies that $h = z_1 - f(\mathbf{z}_2)$ is a divisor of $d_l(\mathbf{F})$, and it is easy to show that $\mathrm{rank}(\mathbf{F}(f, \mathbf{z}_2)) \le l - 1$. Thus, we have $r \le l - 1$. When $r = 0$, $\mathrm{rank}(\mathbf{F}(f, \mathbf{z}_2)) = 0$ implies that $h \mid d_1(\mathbf{F})$. Then, we can extract h from each row of $\mathbf{F}$ and obtain a matrix factorization of $\mathbf{F}$ w.r.t. h^l.*

Let $k[\bar{\mathbf{z}}_j] = k[z_1, \ldots, z_{j-1}, z_{j+1}, \ldots, z_n]$, where $1 \le j \le n$. We construct a new set of polynomial matrices: $\mathcal{M}_j = \{\mathbf{F} \in k[\mathbf{z}]^{l\times m} : h_j \mid d_l(\mathbf{F})\}$, where $h_j = z_j - f(\bar{\mathbf{z}}_j)$ with $f(\bar{\mathbf{z}}_j)$ being a polynomial in $k[\bar{\mathbf{z}}_j]$. Then, we can get the following corollary.

Corollary 3.5. *Let $\mathbf{F} \in \mathcal{M}_j$ and $W = \mathrm{Im}(\mathbf{F}(z_1, \ldots, z_{j-1}, f, z_{j+1}, \ldots, z_n))$. If $Fitt_{r-1}(W) = 0$ and $Fitt_r(W) = \langle d\rangle$ with $d \in k[\bar{\mathbf{z}}_j] \setminus \{0\}$ and $0 \le r \le l-1$, then $\mathbf{F}$ admits a matrix factorization w.r.t. h_j^{l-r}.*

3.2 Uniqueness of Polynomial Matrix Factorizations

In [42], Liu and Wang studied the uniqueness problem of polynomial matrix factorizations. They pointed out that for a non-regular factor h_0 of $\mathbf{F} \in k[\mathbf{z}]^{l\times m}$, under the condition that there exists a matrix factorization $\mathbf{F} = \mathbf{G}_1\mathbf{F}_1$ with $\det(\mathbf{G}_1) = h_0$, $\mathrm{Im}(\mathbf{F}_1)$ is not uniquely determined. In other words, when $\mathbf{F} = \mathbf{G}_1\mathbf{F}_1 = \mathbf{G}_2\mathbf{F}_2$ with $\det(\mathbf{G}_1) = \det(\mathbf{G}_2) = h_0$, $\mathrm{Im}(\mathbf{F}_1)$ and $\mathrm{Im}(\mathbf{F}_2)$ might not be the same.

Let $\mathbf{F} \in \mathcal{M}$ satisfy the conditions of Theorem 3.4. According to the proof of Theorem 3.3, we can select different generators of $\mathrm{Syz}(W)$ and obtain different presentation matrices of W. Then, we can construct different unimodular matrices and get different matrix factorizations of $\mathbf{F}$ w.r.t. h^{l-r}. Hence, in the following we study the uniqueness of matrix factorizations of $\mathbf{F}$ w.r.t. h^{l-r}.

Theorem 3.6. *Let $\mathbf{F} \in \mathcal{M}$ satisfy $\mathbf{F} = \mathbf{U}_1^{-1}\mathbf{D}\mathbf{F}_1 = \mathbf{U}_2^{-1}\mathbf{D}\mathbf{F}_2$, where $\mathbf{U}_1$, $\mathbf{U}_2$ are two unimodular matrices in $k[\mathbf{z}_2]^{l\times l}$, and $\mathbf{D} = \mathrm{diag}(\underbrace{h, \ldots, h}_{l-r}, \underbrace{1, \ldots, 1}_{r})$. Then, $\mathrm{Im}(\mathbf{F}_1) = \mathrm{Im}(\mathbf{F}_2)$.*

Proof. Let $\mathbf{F}_1 = \left[\vec{u}_1^{\mathrm{T}}, \ldots, \vec{u}_l^{\mathrm{T}}\right]^{\mathrm{T}}$ and $\mathbf{F}_2 = \left[\vec{v}_1^{\mathrm{T}}, \ldots, \vec{v}_l^{\mathrm{T}}\right]^{\mathrm{T}}$, where $\vec{u}_1, \ldots, \vec{u}_l, \vec{v}_1, \ldots, \vec{v}_l \in k[\mathbf{z}]^{1\times m}$. So, $\mathrm{Im}(\mathbf{F}_1) = \langle \vec{u}_1, \ldots, \vec{u}_l\rangle$ and $\mathrm{Im}(\mathbf{F}_2) = \langle \vec{v}_1, \ldots, \vec{v}_l\rangle$.

Let $\mathbf{F}_{01} = \mathbf{U}_1\mathbf{F}$ and $\mathbf{F}_{02} = \mathbf{U}_2\mathbf{F}$. Then $\mathbf{F}_{01} = \mathbf{D}\mathbf{F}_1$ and $\mathbf{F}_{02} = \mathbf{D}\mathbf{F}_2$. It follows that $\mathbf{F}_{01} = \left[h\vec{u}_1^{\mathrm{T}}, \ldots, h\vec{u}_{l-r}^{\mathrm{T}}, \vec{u}_{l-r+1}^{\mathrm{T}}, \ldots, \vec{u}_l^{\mathrm{T}}\right]^{\mathrm{T}}$ and $\mathbf{F}_{02} = \left[h\vec{v}_1^{\mathrm{T}}, \ldots, h\vec{v}_{l-r}^{\mathrm{T}}, \vec{v}_{l-r+1}^{\mathrm{T}}, \ldots, \vec{v}_l^{\mathrm{T}}\right]^{\mathrm{T}}$. Since $\mathbf{U}_1$ and $\mathbf{U}_2$ are two unimodular matrices in $k[\mathbf{z}_2]^{l\times l}$, we have $\mathbf{F}_{01} = \mathbf{U}_1\mathbf{U}_2^{-1}\mathbf{F}_{02}$. This implies that there exist polynomials $a_{i1}, \ldots, a_{il} \in k[\mathbf{z}_2]$ such that

$$h\vec{u}_i = h \cdot \Big(\sum_{j=1}^{l-r} a_{ij}\vec{v}_j\Big) + \sum_{j=l-r+1}^{l} a_{ij}\vec{v}_j,$$

where $i = 1, \ldots, l-r$. Then, for each i setting z_1 of the above equation to $f(\mathbf{z}_2)$, we have

$$a_{i(l-r+1)}\vec{v}_{l-r+1}(f, \mathbf{z}_2) + \cdots + a_{il}\vec{v}_l(f, \mathbf{z}_2) = \vec{0}.$$

As $\mathrm{rank}(\mathbf{F}(f, \mathbf{z}_2)) = r$ and $\mathrm{rank}(\mathbf{F}_{02}(f, \mathbf{z}_2)) = \mathrm{rank}(\mathbf{F}(f, \mathbf{z}_2))$, we have that $\vec{v}_{l-r+1}(f, \mathbf{z}_2), \ldots, \vec{v}_l(f, \mathbf{z}_2)$ are $k[\mathbf{z}_2]$-linearly independent. This implies that $a_{i(l-r+1)} = \cdots = a_{il} = 0$. Hence,

$$\vec{u}_i = a_{i1}\vec{v}_1 + \cdots + a_{i(l-r)}\vec{v}_{l-r},$$

where $i = 1, \ldots, l-r$. Obviously, $\vec{u}_j$ is a $k[\mathbf{z}]$-linear combination of $\vec{v}_1, \ldots, \vec{v}_l$, where $j = l-r+1, \ldots, l$. As a consequence, $\langle \vec{u}_1, \ldots, \vec{u}_l\rangle \subset \langle \vec{v}_1, \ldots, \vec{v}_l\rangle$. We can use the same method to prove that $\langle \vec{v}_1, \ldots, \vec{v}_l\rangle \subset \langle \vec{u}_1, \ldots, \vec{u}_l\rangle$.

Therefore, we have $\mathrm{Im}(\mathbf{F}_1) = \mathrm{Im}(\mathbf{F}_2)$. □

Based on Theorem 3.4 and Theorem 3.6, we can now derive the conclusion: if $\mathbf{F} \in \mathcal{M}$ satisfies the conditions of Theorem 3.4, then we have $\mathbf{F} = \mathbf{G}_1\mathbf{F}_1$ with $\det(\mathbf{G}_1) = h^{l-r}$ and $\mathrm{Im}(\mathbf{F}_1)$ uniquely determined, where $\mathbf{G}_1 = \mathbf{U}^{-1}\mathbf{D}$ with $\mathbf{U} \in k[\mathbf{z}_2]^{l\times l}$ a unimodular matrix and $\mathbf{D} = \mathrm{diag}(h, \ldots, h, 1, \ldots, 1)$.

3.3 Algorithm

Combining the algorithm proposed in [37] and the matrix factorization conditions of Theorem 3.4, we get the following algorithm for factoring matrices in $\mathcal{M}$.

Before proceeding further, let us remark on Algorithm 1.

- Step 2 implies that $\mathrm{rank}(W) = r$.
- In Step 7, $\mathbf{H}$ is a presentation matrix of W. By Lemma 2.4, we have $\mathrm{rank}(\mathbf{H}) = l - r$. Thus, $Fitt_{r-1}(W) = I_{l-r+1}(\mathbf{H}) = 0$.
- In Step 9, $\#(\mathcal{G})$ stands for the number of generators in $\mathcal{G}$, $\#(\mathcal{G}) = 1$ implies that $Fitt_r(W)$ is a principal ideal in $k[\mathbf{z}_2]$.
- From Step 10 to Step 12, we refer to [44, 45] for more details.
- In Step 15, we need to find another new criterion to judge whether $\mathbf{F}$ has a matrix factorization w.r.t. h^{l-r}.

Now, we use an example to illustrate the calculation process of Algorithm 1.

Example 3.7. Let

$$\mathbf{F} = \begin{bmatrix} z_1^2 - z_1z_2 & z_2z_3 + z_3^2 + z_2 + z_3 & -z_2z_3 - z_2 \\ z_1z_2 - z_2^2 & -z_1z_3 + z_2z_3 & z_1^3 - z_1^2z_2 + z_1z_2 - z_2^2 \\ 0 & z_2 + z_3 & -z_2 \end{bmatrix}$$

be a multivariate polynomial matrix in $\mathbb{C}[z_1, z_2, z_3]^{3\times 3}$, where $z_1 > z_2 > z_3$ and $\mathbb{C}$ is the complex field.

Algorithm 1: polynomial matrix factorization algorithm

Input : $\mathbf{F} \in \mathcal{M}$, $h = z_1 - f(\mathbf{z}_2)$ and a monomial order $\prec_{\mathbf{z}_2}$ in $k[\mathbf{z}_2]$.

Output: a matrix factorization of $\mathbf{F}$ w.r.t. h^{l-r}, where r is the rank of $\mathbf{F}(f, \mathbf{z}_2)$.

begin
- compute the rank r of $\mathbf{F}(f, \mathbf{z}_2)$;
- **if** $r = 0$ **then**
  - extract h from each row of $\mathbf{F}$ and obtain $\mathbf{F}_1$, i.e., $\mathbf{F} = \mathrm{diag}(h, \ldots, h) \cdot \mathbf{F}_1$;
  - **return** $\mathrm{diag}(h, \ldots, h)$ and $\mathbf{F}_1$.
- compute a Gröbner basis $\{\vec{h}_1, \ldots, \vec{h}_s\}$ of the syzygy module of $W = \mathrm{Im}(\mathbf{F}(f, \mathbf{z}_2))$;
- let $\mathbf{H}$ be a matrix in $k[\mathbf{z}_2]^{s\times l}$ composed of $\vec{h}_1, \ldots, \vec{h}_s$;
- compute a reduced Gröbner basis $\mathcal{G}$ of the $(l-r)$-th determinantal ideal of $\mathbf{H}$ w.r.t. $\prec_{\mathbf{z}_2}$;
- **if** $\#(\mathcal{G}) = 1$ **then**
  - compute a ZLP matrix factorization of $\mathbf{H}$ and obtain a ZLP matrix $\mathbf{H}_1 \in k[\mathbf{z}_2]^{(l-r)\times l}$;
  - construct a unimodular matrix $\mathbf{U} \in k[\mathbf{z}_2]^{l\times l}$ such that $\mathbf{H}_1$ is its first $l-r$ rows;
  - extract h from the first $l-r$ rows of $\mathbf{UF}$ and obtain $\mathbf{F}_1$, i.e., $\mathbf{UF} = \mathrm{diag}(h, \ldots, h, 1, \ldots, 1) \cdot \mathbf{F}_1$;
  - **return** $\mathbf{U}^{-1} \cdot \mathrm{diag}(h, \ldots, h, 1, \ldots, 1)$ and $\mathbf{F}_1$.
- **else**
  - **return** unable to judge.

It is easy to compute that $d_3(\mathbf{F}) = -z_1(z_1-z_2)^2(z_1^2z_2+z_1^2z_3+z_2^2)$, $d_2(\mathbf{F}) = z_1 - z_2$ and $d_1(\mathbf{F}) = 1$. Let $h = z_1 - z_2$ and $\prec_{z_2,z_3}$ be the degree reverse lexicographic order. Then, the input of Algorithm 1 are $\mathbf{F}$, $h = z_1 - z_2$ and $\prec_{z_2,z_3}$.

Note that

$$\mathbf{F}(z_2, z_2, z_3) = \begin{bmatrix} 0 & (z_2+z_3)(z_3+1) & -z_2(z_3+1) \\ 0 & 0 & 0 \\ 0 & z_2+z_3 & -z_2 \end{bmatrix},$$

the rank of $\mathbf{F}(z_2, z_2, z_3)$ is $r = 1$. Let $W = \mathrm{Im}(\mathbf{F}(z_2, z_2, z_3))$. Then, we use *Singular* command "syz" to compute a Gröbner basis of the syzygy module of W, and obtain

$$\mathbf{H} = \begin{bmatrix} 1 & 0 & -z_3-1 \\ 0 & 1 & 0 \end{bmatrix}.$$

It is easy to check that the reduced Gröbner basis of all the 2×2 minors of $\mathbf{H}$ w.r.t. $\prec_{z_2,z_3}$ is $\mathcal{G} = \{1\}$. Then, $Fitt_1(W) = I_2(\mathbf{H}) = \langle 1\rangle$ and $\mathbf{H}$ is a ZLP matrix. This implies that $\mathbf{H}_1 = \mathbf{H}$. $\mathbf{H}_1$ can be easily extended as the first 2 rows of a unimodular matrix

$$\mathbf{U} = \begin{bmatrix} 1 & 0 & -z_3-1 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}.$$

We can extract h from the first 2 rows of $\mathbf{UF}$, and get

$$\mathbf{UF} = \mathbf{DF}_1 = \begin{bmatrix} z_1-z_2 & 0 & 0 \\ 0 & z_1-z_2 & 0 \\ 0 & 0 & 1 \end{bmatrix}\begin{bmatrix} z_1 & 0 & 0 \\ z_2 & -z_3 & z_1^2+z_2 \\ 0 & z_2+z_3 & -z_2 \end{bmatrix}.$$

Then, we obtain a matrix factorization of $\mathbf{F}$ w.r.t. h^2: $\mathbf{F} = \mathbf{G}_1\mathbf{F}_1 = (\mathbf{U}^{-1}\mathbf{D})\mathbf{F}_1 =$

$$\begin{bmatrix} z_1-z_2 & 0 & z_3+1 \\ 0 & z_1-z_2 & 0 \\ 0 & 0 & 1 \end{bmatrix}\begin{bmatrix} z_1 & 0 & 0 \\ z_2 & -z_3 & z_1^2+z_2 \\ 0 & z_2+z_3 & -z_2 \end{bmatrix},$$

where $\det(\mathbf{G}_1) = \det(\mathbf{U}^{-1}\mathbf{D}) = h^2$.

At this moment, $d_3(\mathbf{F}_1) = -z_1(z_1^2z_2 + z_1^2z_3 + z_2^2)$. We reuse Algorithm 1 to judge whether $\mathbf{F}_1$ has a matrix factorization w.r.t. z_1. Similarly, we obtain

$$\mathbf{F}_1 = \mathbf{G}_2\mathbf{F}_2 = \begin{bmatrix} z_1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}\begin{bmatrix} 1 & 0 & 0 \\ z_2 & -z_3 & z_1^2+z_2 \\ 0 & z_2+z_3 & -z_2 \end{bmatrix},$$

where $\det(\mathbf{G}_2) = z_1$.

Therefore, we obtain a matrix factorization of $\mathbf{F}$ w.r.t. $z_1(z_1-z_2)^2$, i.e., $\mathbf{F} = \mathbf{GF}_2 = (\mathbf{G}_1\mathbf{G}_2)\mathbf{F}_2 =$

$$\begin{bmatrix} z_1(z_1-z_2) & 0 & z_3+1 \\ 0 & z_1-z_2 & 0 \\ 0 & 0 & 1 \end{bmatrix}\begin{bmatrix} 1 & 0 & 0 \\ z_2 & -z_3 & z_1^2+z_2 \\ 0 & z_2+z_3 & -z_2 \end{bmatrix},$$

where $\det(\mathbf{G}) = z_1(z_1-z_2)^2$.

Remark 4. *In Example 3.7, we can first judge whether $\mathbf{F}$ has a matrix factorization w.r.t. z_1. Note that*

$$\mathbf{F}(0, z_2, z_3) = \begin{bmatrix} 0 & (z_2+z_3)(z_3+1) & -z_2(z_3+1) \\ -z_2^2 & z_2z_3 & -z_2^2 \\ 0 & z_2+z_3 & -z_2 \end{bmatrix},$$

the rank of $\mathbf{F}(0, z_2, z_3)$ is $r = 2$. We compute a Gröbner basis of the syzygy module of $\mathrm{Im}(\mathbf{F}(0, z_2, z_3))$ and get $\mathbf{H} = \begin{bmatrix} -1 & 0 & z_3+1 \end{bmatrix}$. Since the reduced Gröbner basis of all the 1×1 minors of $\mathbf{H}$ w.r.t. $\prec_{z_2,z_3}$ is $\mathcal{G} = \{1\}$. Then, $Fitt_2(W) = I_1(\mathbf{H}) = \langle 1\rangle$. This implies that $\mathbf{F}$ has a matrix factorization w.r.t. z_1. According to the above calculations, we have the following conclusions: $\mathbf{F}$ has matrix factorizations w.r.t. z_1, $z_1 - z_2$, $z_1(z_1-z_2)$, $(z_1-z_2)^2$ and $z_1(z_1-z_2)^2$, respectively.

4 EQUIVALENCE FOR POLYNOMIAL MATRICES

In this section, we first put forward a sufficient and necessary condition to solve Problem 2, and then we use an example to illustrate the effectiveness of the new matrix equivalence theorem.

4.1 Matrix Equivalence Theorem

We first introduce a lemma, which is a generalization of Binet-Cauchy formula in [53].

Lemma 4.1 ([53]). *Let $\mathbf{F} = \mathbf{G}_1\mathbf{F}_1$, where $\mathbf{G}_1 \in k[\mathbf{z}]^{l\times l}$ and $\mathbf{F}_1 \in k[\mathbf{z}]^{l\times m}$. Then an $i\times i$ $(1 \le i \le l)$ minor of $\mathbf{F}$ is*

$$\det\left(\mathbf{F}\begin{pmatrix} r_1 \cdots r_i \\ j_1 \cdots j_i \end{pmatrix}\right) = \sum_{1\le s_1<\cdots<s_i\le l} \det\left(\mathbf{G}_1\begin{pmatrix} r_1 \cdots r_i \\ s_1 \cdots s_i \end{pmatrix}\right)\cdot\det\left(\mathbf{F}_1\begin{pmatrix} s_1 \cdots s_i \\ j_1 \cdots j_i \end{pmatrix}\right).$$

In Lemma 4.1, $\mathbf{F}\begin{pmatrix} r_1 \cdots r_i \\ j_1 \cdots j_i \end{pmatrix}$ denotes an $i\times i$ sub-matrix consisting of the $r_1, \ldots, r_i$ rows and $j_1, \ldots, j_i$ columns of $\mathbf{F}$. Based on this lemma, we can obtain the following two results.

Lemma 4.2. *Let $\mathbf{F} \in k[\mathbf{z}]^{l\times m}$ be of full row rank with $\mathbf{F} = \mathbf{G}_1\mathbf{F}_1$, where $\mathbf{G}_1 \in k[\mathbf{z}]^{l\times l}$ and $\mathbf{F}_1 \in k[\mathbf{z}]^{l\times m}$. Then $d_i(\mathbf{F}_1) \mid d_i(\mathbf{F})$ and $d_i(\mathbf{G}_1) \mid d_i(\mathbf{F})$ for each $i \in \{1,\ldots,l\}$.*

Proof. We only prove $d_i(\mathbf{F}_1) \mid d_i(\mathbf{F})$, since the proof of $d_i(\mathbf{G}_1) \mid d_i(\mathbf{F})$ follows in a similar manner. For any given $i \in \{1,\ldots,l\}$, let $a_{i,1},\ldots,a_{i,t_i}$ and $\bar{a}_{i,1},\ldots,\bar{a}_{i,t_i}$ be all the $i\times i$ minors of $\mathbf{F}$ and $\mathbf{F}_1$ respectively, where $t_i = \binom{l}{i}\binom{m}{i}$. For each $a_{i,j}$, it is a $k[\mathbf{z}]$-linear combination of $\bar{a}_{i,1},\ldots,\bar{a}_{i,t_i}$ by using Lemma 4.1, where $j = 1,\ldots,t_i$. Since $d_i(\mathbf{F}_1) = \mathrm{GCD}(\bar{a}_{i,1},\ldots,\bar{a}_{i,t_i})$, for each j we have $d_i(\mathbf{F}_1) \mid a_{i,j}$. Then, $d_i(\mathbf{F}_1) \mid d_i(\mathbf{F})$. □

Lemma 4.3. *Let $\mathbf{F}_1, \mathbf{F}_2 \in k[\mathbf{z}]^{l\times m}$ be of full row rank. If $\mathbf{F}_1$ and $\mathbf{F}_2$ are equivalent, then $d_i(\mathbf{F}_1) = d_i(\mathbf{F}_2)$ for each $i \in \{1,\ldots,l\}$.*

Proof. Since $\mathbf{F}_1$ and $\mathbf{F}_2$ are equivalent, then there exist two unimodular matrices $\mathbf{U} \in k[\mathbf{z}]^{l\times l}$ and $\mathbf{V} \in k[\mathbf{z}]^{m\times m}$ such that $\mathbf{F}_1 = \mathbf{U}\mathbf{F}_2\mathbf{V}$. For each $i \in \{1,\ldots,l\}$, it follows from Lemma 4.2 that $d_i(\mathbf{F}_2) \mid d_i(\mathbf{U}\mathbf{F}_2) \mid d_i(\mathbf{F}_1)$. Furthermore, we have $\mathbf{F}_2 = \mathbf{U}^{-1}\mathbf{F}_1\mathbf{V}^{-1}$ since $\mathbf{U}$ and $\mathbf{V}$ are two unimodular matrices. Similarly, we obtain $d_i(\mathbf{F}_1) \mid d_i(\mathbf{U}^{-1}\mathbf{F}_1) \mid d_i(\mathbf{F}_2)$. Therefore, $d_i(\mathbf{F}_1) = d_i(\mathbf{F}_2)$. □

Before presenting the matrix equivalence theorem, we introduce a lemma which plays an important role in our proof.

Lemma 4.4 ([44]). *Let $\mathbf{F} \in k[\mathbf{z}]^{l\times m}$ with $\mathrm{rank}(\mathbf{F}) = r$. If all the $r\times r$ minors of $\mathbf{F}$ generate $k[\mathbf{z}]$, then there exists a ZLP matrix $\mathbf{H} \in k[\mathbf{z}]^{(l-r)\times l}$ such that $\mathbf{H}\mathbf{F} = \mathbf{0}_{(l-r)\times m}$.*

Combining Lemma 4.4 and the Quillen-Suslin theorem, we can now solve Problem 2.

Theorem 4.5. *Let $\mathbf{F} \in k[\mathbf{z}]^{l\times l}$ with $\det(\mathbf{F}) = h^r$, where $h = z_1 - f(\mathbf{z}_2)$ and $1 \le r \le l$. Then $\mathbf{F}$ and $\mathrm{diag}(1,\ldots,1,h,\ldots,h)$ are equivalent if and only if $h \mid d_{l-r+1}(\mathbf{F})$ and the ideal generated by h and all the $(l-r)\times(l-r)$ minors of $\mathbf{F}$ is $k[\mathbf{z}]$.*

Proof. For convenience, let $\mathbf{D} = \mathrm{diag}(1,\ldots,1,h,\ldots,h)$ and $\bar{\mathbf{F}} = \mathbf{F}(f,\mathbf{z}_2)$. Let $a_1,\ldots,a_\beta$ be all the $(l-r)\times(l-r)$ minors of $\mathbf{F}$. It is obvious that $a_1(f,\mathbf{z}_2),\ldots,a_\beta(f,\mathbf{z}_2)$ are all the $(l-r)\times(l-r)$ minors of $\bar{\mathbf{F}}$.

Sufficiency. It follows from $h \mid d_{l-r+1}(\mathbf{F})$ that $\mathrm{rank}(\bar{\mathbf{F}}) \le l-r$. Assume that there exists a point $(\varepsilon_2,\ldots,\varepsilon_n) \in k^{1\times(n-1)}$ such that

$$a_i(f(\varepsilon_2,\ldots,\varepsilon_n),\varepsilon_2,\ldots,\varepsilon_n) = 0,\ i = 1,\ldots,\beta. \tag{4}$$

Let $\varepsilon_1 = f(\varepsilon_2,\ldots,\varepsilon_n)$, then Equation (4) implies that $(\varepsilon_1,\varepsilon_2,\ldots,\varepsilon_n) \in k^{1\times n}$ is a common zero of the polynomial system $\{h = 0, a_1 = 0,\ldots,a_\beta = 0\}$. This contradicts the fact that h and all the $(l-r)\times(l-r)$ minors of $\mathbf{F}$ generate $k[\mathbf{z}]$. Then, all the $(l-r)\times(l-r)$ minors of $\bar{\mathbf{F}}$ generate $k[\mathbf{z}_2]$. According to Lemma 4.4, there exists a ZLP matrix $\mathbf{H} \in k[\mathbf{z}_2]^{r\times l}$ such that $\mathbf{H}\bar{\mathbf{F}} = \mathbf{0}_{r\times l}$. Based on the Quillen-Suslin theorem, we can construct a unimodular matrix $\mathbf{U} \in k[\mathbf{z}_2]^{l\times l}$ such that $\mathbf{H}$ is its last r rows. Then, there is a polynomial matrix $\mathbf{V} \in k[\mathbf{z}]^{l\times l}$ such that $\mathbf{U}\mathbf{F} = \mathbf{D}\mathbf{V}$. Since $\det(\mathbf{F}) = h^r$ and $\mathbf{U}$ is a unimodular matrix, we have $\mathbf{F} = \mathbf{U}^{-1}\mathbf{D}\mathbf{V}$ and $\mathbf{V}$ is a unimodular matrix. Therefore, $\mathbf{F}$ and $\mathbf{D}$ are equivalent.

Necessity. If $\mathbf{F}$ and $\mathbf{D}$ are equivalent, then there exist two unimodular matrices $\mathbf{U} \in k[\mathbf{z}]^{l\times l}$ and $\mathbf{V} \in k[\mathbf{z}]^{l\times l}$ such that $\mathbf{F} = \mathbf{U}\mathbf{D}\mathbf{V}$. It follows from Lemma 4.3 that $d_{l-r+1}(\mathbf{F}) = d_{l-r+1}(\mathbf{D}) = h$. If $\langle h, a_1,\ldots,a_\beta\rangle \ne k[\mathbf{z}]$, then there exists a point $\vec{\varepsilon} \in k^{1\times n}$ such that $h(\vec{\varepsilon}) = 0$ and $\mathrm{rank}(\mathbf{F}(\vec{\varepsilon})) < l-r$. Obviously, $\mathrm{rank}(\mathbf{D}(\vec{\varepsilon})) = l-r$ and $\mathrm{rank}(\mathbf{U}^{-1}(\vec{\varepsilon})) = \mathrm{rank}(\mathbf{V}^{-1}(\vec{\varepsilon})) = l$. Since $\mathbf{D} = \mathbf{U}^{-1}\mathbf{F}\mathbf{V}^{-1}$, we have

$$\mathrm{rank}(\mathbf{D}(\vec{\varepsilon})) \le \min\{\mathrm{rank}(\mathbf{U}^{-1}(\vec{\varepsilon})), \mathrm{rank}(\mathbf{F}(\vec{\varepsilon})), \mathrm{rank}(\mathbf{V}^{-1}(\vec{\varepsilon}))\},$$

which leads to a contradiction. Therefore, $\langle h, a_1,\ldots,a_\beta\rangle = k[\mathbf{z}]$ and the proof is completed. □

Remark 5. *When $r = l$ in Theorem 4.5, we just need to check whether h is a divisor of $d_1(\mathbf{F})$.*

4.2 Example

Now, we use Example 2.12 to illustrate a constructive method which follows Lin et al. in [35] and explain how to obtain the two unimodular matrices associated with equivalent matrices in Theorem 4.5.

Example 4.6. Let $\mathbf{F}$ be the same polynomial matrix as in Example 2.12. It is easy to compute that $\det(\mathbf{F}) = (z_1 - z_2)^2$ and $d_2(\mathbf{F}) = z_1 - z_2$. Let $h = z_1 - z_2$, it is obvious that $h \mid d_2(\mathbf{F})$. The reduced Gröbner basis of the ideal generated by h and all the 1×1 minors of $\mathbf{F}$ w.r.t. $\prec_{\mathbf{z}}$ is $\{1\}$. Then, $\mathbf{F}$ is equivalent to $\mathrm{diag}(1,h,h)$.

Note that

$$\mathbf{F}(z_2,z_2,z_3) = \begin{bmatrix} (z_3+1)(z_2-1) & z_3(z_2-1) & 0 \\ z_3+1 & z_3 & 0 \\ 0 & 0 & 0 \end{bmatrix},$$

the rank of $\mathbf{F}(z_2,z_2,z_3)$ is $r = 1$. According to the calculation process of Example 3.7, we can get a ZLP matrix

$$\mathbf{H} = \begin{bmatrix} 1 & -z_2+1 & z_2^2 - z_2 \\ -1 & z_2 - 1 & -z_2^2 + z_2 + 1 \end{bmatrix}$$

such that $\mathbf{H}\cdot\mathbf{F}(z_2,z_2,z_3) = \mathbf{0}_{2\times 3}$. Then, a unimodular matrix $\mathbf{U} \in k[\mathbf{z}_2]^{3\times 3}$ can be constructed such that $\mathbf{H}$ is its the last 2 rows, where

$$\mathbf{U} = \begin{bmatrix} -1 & z_2 & -z_2^2 \\ 1 & -z_2+1 & z_2^2 - z_2 \\ -1 & z_2 - 1 & -z_2^2 + z_2 + 1 \end{bmatrix}.$$

Now we can extract h from the last 2 rows of $\mathbf{U}\mathbf{F}$, and get $\mathbf{F} = \mathbf{U}^{-1}\cdot\mathrm{diag}(1,h,h)\cdot\mathbf{V} =$

$$\begin{bmatrix} z_2-1 & z_2 & 0 \\ 1 & z_2+1 & z_2 \\ 0 & 1 & 1 \end{bmatrix}\begin{bmatrix} 1 & 0 & 0 \\ 0 & h & 0 \\ 0 & 0 & h \end{bmatrix}\begin{bmatrix} z_3+1 & z_3 & 0 \\ 1 & z_3+1 & z_3 \\ 0 & 1 & 1 \end{bmatrix}.$$

5 CONCLUDING REMARKS

In this paper, we point out two directions of research in which multivariate polynomial matrices have been explored. The first is concerned with the factorization problem of multivariate polynomial matrices in $\mathcal{M}$, and the second direction is devoted to the investigation of the equivalence problem of square matrices in $\mathcal{M}$.

The main contributions of this paper include: (1) a new criterion (Theorem 3.4) and an algorithm (Algorithm 1) are given to factorize $\mathbf{F} \in \mathcal{M}$ w.r.t. h^{l-r}, as a consequence, the application range of the constructive algorithm in [37] has been greatly extended; (2) Theorem 3.6 shows that the output of Algorithm 1 is unique if $\mathbf{F}$ satisfies the new criterion; (3) a sufficient and necessary condition (Theorem 4.5) is proposed to judge whether a square polynomial matrix $\mathbf{F}$ with $\det(\mathbf{F}) = h^r$ is equivalent to $\mathrm{diag}(1,\ldots,1,h,\ldots,h)$; (4) a generalization about the type of polynomial matrices has been

presented (Corollary 3.5) and the implementation of two main theorems (Theorem 3.4 and Theorem 4.5) has been illustrated by two non-trivial examples.

If $\#(\mathcal{G}) \neq 1$, then Algorithm 1 returns "unable to judge". At this moment, how to establish a necessary and sufficient condition for $\mathbf{F} \in \mathcal{M}$ admitting a matrix factorization w.r.t. h^{l-r} is the question that remains for further investigation.

ACKNOWLEDGMENTS

This research was supported in part by the CAS Key Project QYZDJ-SSW-SYS022.

REFERENCES

[1] N. Bose. 1982. *Applied Multidimensional Systems Theory.* Van Nostrand Reinhold Co., New York.

[2] N. Bose, B. Buchberger, and J. Guiver. 2003. *Multidimensional Systems Theory and Applications.* Kluwer Academic Publishers, Dordrecht, The Netherlands.

[3] M. Boudellioua. 2012. Computation of the Smith form for multivariate polynomial matrices using Maple. *American Journal of Computational Mathematics* 2, 1 (2012), 21–26.

[4] M. Boudellioua. 2013. Further results on the equivalence to Smith form of multivariate polynomial matrices. *Control and Cybernetics* 42, 2 (2013), 543–551.

[5] M. Boudellioua. 2014. Computation of a canonical form for linear 2-D systems. *International Journal of Computational Mathematics* 2014, 487465 (2014), 1–6.

[6] M. Boudellioua and A. Quadrat. 2010. Serre's reduction of linear function systems. *Mathematics in Computer Science* 4, 2-3 (2010), 289–312.

[7] W. Brown. 1992. *Matrices over Commutative Rings.* Taylor and Francis.

[8] C. Charoenlarpnopparut and N. Bose. 1999. Multidimensional FIR filter bank design using Gröbner bases. *IEEE Transactions on Circuits and Systems II: Analog and Digital Signal Processing* 46, 12 (1999), 1475–1486.

[9] T. Cluzeau and A. Quadrat. 2008. Factoring and decomposing a class of linear functional systems. *Linear Algebra and Its Applications* 428 (2008), 324–381.

[10] T. Cluzeau and A. Quadrat. 2013. Isomorphisms and Serre's reduction of linear systems. In *Proceedings of the 8th International Workshop on Multidimensional Systems*. VDE, Erlangen, Germany, 1–6.

[11] T. Cluzeau and A. Quadrat. 2015. A new insight into Serre's reduction problem. *Linear Algebra Appl.* 483 (2015), 40–100.

[12] D. Cox, J. Little, and D. O'shea. 2005. *Using Algebraic Geometry.* Springer, New York.

[13] W. Decker, G. Greuel, G. Pfister, and H. Schoenemann. 2016. SINGULAR 4.0.3. a computer algebra system for polynomial computations. https://www.singular.uni-kl.de/

[14] W. Decker and C. Lossen. 2006. *Computing in Algebraic Geometry: a quick start using SINGULAR.* Springer-Verlag.

[15] D. Eisenbud. 2013. *Commutative Algebra: with a view toward algebraic geometry.* New York: Springer.

[16] A. Fabiańska and A. Quadrat. 2007. *Applications of the Quillen-Suslin theorem to multidimensional systems theory.* In: Park, H., Regensburger, G. (Eds.), Gröbner Bases in Control Theory and Signal Processing, Radon Series on Computational and Applied Mathematics, Vol. 3. Walter de Gruyter. 23–106 pages.

[17] A. Fabiańska and A. Quadrat. 2007. A Maple implementation of a constructive version of the Quillen-Suslin theorem. https://wwwb.math.rwth-aachen.de/QuillenSuslin/

[18] M.G. Frost and M.S. Boudellioua. 1986. Some further results concerning matrices with elements in a polynomial ring. *Internat. J. Control* 43, 5 (1986), 1543–1555.

[19] M. Frost and C. Storey. 1978. Equivalence of a matrix over $R[s, z]$ with its Smith form. *Internat. J. Control* 28, 5 (1978), 665–671.

[20] G. Greuel and G. Pfister. 2002. *A SINGULAR Introduction to Commutative Algebra.* Springer-Verlag.

[21] J. Guan, W. Li, and B. Ouyang. 2018. On rank factorizations and factor prime factorizations for multivariate polynomial matrices. *Journal of Systems Science and Complexity* 31, 6 (2018), 1647–1658.

[22] J. Guan, W. Li, and B. Ouyang. 2019. On minor prime factorizations for multivariate polynomial matrices. *Multidimensional Systems and Signal Processing* 30 (2019), 493–502.

[23] J. Guiver and N. Bose. 1982. Polynomial matrix primitive factorization over arbitrary coefficient field and related results. *IEEE Transactions on Circuits and Systems* 29, 10 (1982), 649–657.

[24] T. Kailath. 1993. *Linear Systems.* Englewood Cliffs, NJ: Prentice Hall.

[25] S. Kung, B. Levy, M. Morf, and T. Kailath. 1977. New results in 2-D systems theory, part II: 2-D state-space models–realization and the notions of controllability, observability, and minimality. In *Proceedings of the IEEE*, Vol. 65. 945–961.

[26] E. Lee and S. Zak. 1983. Smith forms over $R[z_1, z_2]$. *IEEE Trans. Automat. Control* 28, 1 (1983), 115–118.

[27] D. Li, J. Liu, and L. Zheng. 2017. On the equivalence of multivariate polynomial matrices. *Multidimensional Systems and Signal Processing* 28 (2017), 225–235.

[28] Z. Lin. 1988. On matrix fraction descriptions of multivariable linear n-D systems. *IEEE Transactions on Circuits and Systems* 35, 10 (1988), 1317–1322.

[29] Z. Lin. 1992. On primitive factorizations for 3-D polynomial matrices. *IEEE Transactions on Circuits and Systems I: Fundamental Theory and Applications* 39, 12 (1992), 1024–1027.

[30] Z. Lin. 1993. On primitive factorizations for n-D polynomial matrices. In *IEEE International Symposium on Circuits and Systems*. 601–618.

[31] Z. Lin. 1999. Notes on n-D polynomial matrix factorizations. *Multidimensional Systems and Signal Processing* 10, 4 (1999), 379–393.

[32] Z. Lin. 1999. On syzygy modules for polynomial matrices. *Linear Algebra and Its Applications* 298, 1-3 (1999), 73–86.

[33] Z. Lin. 2001. Further results on n-D polynomial matrix factorizations. *Multidimensional Systems and Signal Processing* 12, 2 (2001), 199–208.

[34] Z. Lin and N. Bose. 2001. A generalization of Serre's conjecture and some related issues. *Linear Algebra and Its Applications* 338 (2001), 125–138.

[35] Z. Lin, M. Boudellioua, and L. Xu. 2006. On the equivalence and factorization of multivariate polynomial matrices. In *Proceeding of ISCAS*. Greece, 4911–4914.

[36] Z. Lin, X. Li, and H. Fan. 2005. On minor prime factorizations for n-D polynomial matrices. *IEEE Transactions on Circuits and Systems II: Express Briefs* 52, 9 (2005), 568–571.

[37] Z. Lin, J. Ying, and L. Xu. 2001. Factorizations for n-D polynomial matrices. *Circuits, Systems, and Signal Processing* 20, 6 (2001), 601–618.

[38] J. Liu, D. Li, and M. Wang. 2011. On general factorizations for n-D polynomial matrices. *Circuits Systems and Signal Processing* 30, 3 (2011), 553–566.

[39] J. Liu, D. Li, and L. Zheng. 2014. The Lin-Bose problem. *IEEE Transactions on Circuits and Systems II: Express Briefs* 61, 1 (2014), 41–43.

[40] J. Liu and M. Wang. 2010. Notes on factor prime factorizations for n-D polynomial matrices. *Multidimensional Systems and Signal Processing* 21, 1 (2010), 87–97.

[41] J. Liu and M. Wang. 2013. New results on multivariate polynomial matrix factorizations. *Linear Algebra and Its Applications* 438, 1 (2013), 87–95.

[42] J. Liu and M. Wang. 2015. Further remarks on multivariate polynomial matrix factorizations. *Linear Algebra and Its Applications* 465 (2015), 204–213.

[43] A. Logar and B. Sturmfels. 1992. Algorithms for the Quillen-Suslin theorem. *Journal of Algebra* 145, 1 (1992), 231–239.

[44] D. Lu, X. Ma, and D. Wang. 2017. A new algorithm for general factorizations of multivariate polynomial matrices. In *proceedings of 42nd ISSAC*. 277–284.

[45] D. Lu, D. Wang, and F. Xiao. 2020. Factorizations for a class of multivariate polynomial matrices. *Multidimensional Systems and Signal Processing* 31 (2020), 989–1004.

[46] M. Morf, B. Levy, and S. Kung. 1977. New results in 2-D systems theory, part I: 2-D polynomial matrices, factorization, and coprimeness. In *Proceedings of the IEEE*, Vol. 65. 861–872.

[47] H. Park. 1995. *A computational theory of Laurent polynomial rings and multidimensional FIR systems.* Ph.D. Dissertation. University of California at Berkeley.

[48] J. Pommaret. 2001. Solving Bose conjecture on linear multidimensional systems. In *Proceeding of European Control Conference*. IEEE, 1653–1655.

[49] A. Pugh, S. Mcinerney, M. Boudellioua, D. Johnson, and G. Hayton. 1998. A transformaition for 2-D linear systems and a generalization of a theorem of rosenbrock. *Internat. J. Control* 71, 3 (1998), 491–503.

[50] D. Quillen. 1976. Projective modules over polynomial rings. *Inventiones mathematicae* 36 (1976), 167–171.

[51] H. Rosenbrock. 1970. *State-space and Multivariable Theory.* London: Nelson.

[52] J. Serre. 1955. Faisceaux algébriques cohérents. *Annals of Mathematics (Second Series)* 61, 2 (1955), 197–278.

[53] G. Strang. 1980. *Linear Algebra and Its Applications (Second Edition).* Academic Press.

[54] V. Sule. 1994. Feedback stabilization over commutative rings: the matrix case. *SIAM Journal on Control and Optimization* 32, 6 (1994), 1675–1695.

[55] A. Suslin. 1976. Projective modules over polynomial rings are free. *Soviet mathematics - Doklady* 17 (1976), 1160–1165.

[56] M. Wang. 2007. On factor prime factorization for n-D polynomial matrices. *IEEE Transactions on Circuits and Systems I: Regular Papers* 54, 6 (2007), 1398–1405.

[57] M. Wang. 2008. Remarks on n-D polynomial matrix factorization problems. *IEEE Transactions on Circuits and Systems II: Express Briefs* 55, 1 (2008), 61–64.

[58] M. Wang and D. Feng. 2004. On Lin-Bose problem. *Linear Algebra and Its Applications* 390 (2004), 279–285.

[59] M. Wang and C.P. Kwong. 2005. On multivariate polynomial matrix factorization problems. *Mathematics of Control, Signals and Systems* 17, 4 (2005), 297–311.

[60] D. Youla and G. Gnavi. 1979. Notes on n-dimensional system theory. *IEEE Transactions on Circuits and Systems* 26, 2 (1979), 105–111.

[61] D. Youla and P. Pickel. 1984. The Quillen-Suslin theorem and the structure of n-dimensional elementary polynomial matrices. *IEEE Transactions on Circuits and Systems* 31, 6 (1984), 513–518.

Punctual Hilbert Scheme and Certified Approximate Singularities

Angelos Mantzaflaris, Bernard Mourrain
angelos.mantzaflaris@inria.fr,bernard.mourrain@inria.fr
INRIA Sophia Antipolis, Université Côte d'Azur
Sophia Antipolis, France

Agnes Szanto
aszanto@ncsu.edu
Dept. of Mathematics, North Carolina State University
Raleigh, NC, USA

ABSTRACT

In this paper we provide a new method to certify that a nearby polynomial system has a singular isolated root and we compute its multiplicity structure. More precisely, given a polynomial system $\mathbf{f} = (f_1, \ldots, f_N) \in \mathbb{C}[x_1, \ldots, x_n]^N$, we present a Newton iteration on an extended deflated system that locally converges, under regularity conditions, to a small deformation of f such that this deformed system has an exact singular root. The iteration simultaneously converges to the coordinates of the singular root and the coefficients of the so-called inverse system that describes the multiplicity structure at the root. We use α-theory test to certify the quadratic convergence, and to give bounds on the size of the deformation and on the approximation error. The approach relies on an analysis of the punctual Hilbert scheme, for which we provide a new description. We show in particular that some of its strata can be rationally parametrized and exploit these parametrizations in the certification. We show in numerical experimentation how the approximate inverse system can be computed as a starting point of the Newton iterations and the fast numerical convergence to the singular root with its multiplicity structure, certified by our criteria.

CCS CONCEPTS

• **Mathematics of Computing → Roots of Nonlinear Equations**.

KEYWORDS

certification, singularity, multiplicity structure, Newton's method, inverse system, multiplication matrix

ACM Reference Format:
Angelos Mantzaflaris, Bernard Mourrain and Agnes Szanto. 2020. Punctual Hilbert Scheme and Certified Approximate Singularities. In *International Symposium on Symbolic and Algebraic Computation (ISSAC '20), July 20–23, 2020, Kalamata, Greece.* ACM, New York, NY, USA, 8 pages. https://doi.org/10.1145/3373207.3404024

1 INTRODUCTION

Local numerical methods such as Newton iterations have proved their efficiency to approximate and certify the existence of simple roots. However for multiple roots they dramatically fail to provide fast numerical convergence and certification. The motivation for this work is to find a method with fast convergence to an exact singular point and its multiplicity structure for a small perturbation of the input polynomials, and to give numerical tests that can certify it. The knowledge of the multiplicity structure together with a high precision numerical approximation of a singular solution can be valuable information in many problems.

In [27] a method called later *integration method* is devised to compute the so-called *inverse system* or multiplicity structure at a multiple root. It is used in [25] to compute an approximation of the inverse system, given an approximation of that root and to obtain a perturbed system that satisfies the duality property. However, this method did not give a way to improve the accuracy of the initial approximation of the root and the corresponding inverse system. In [16] a new one-step deflation method is presented that gives an overdetermined polynomial system in the coordinates of the roots and the corresponding inverse system, serving as a starting point for the present paper. However, for certification, [16] refers to the symbolic-numeric method in [1] that only works if the input system is given exactly with rational coefficients and have a multiple root with the prescribed multiplicity structure.

In the present paper we give a solution for the following problem:

Problem 1.1. Given a polynomial system $\mathbf{f} = (f_1, \ldots, f_N) \in \mathbb{C}[\mathbf{x}]^N$ and a point $\xi \in \mathbb{C}^n$, deduce an iterative method that converges quadratically to the triple $(\xi^*, \mu^*, \epsilon^*)$ such that $\xi^* \in \mathbb{C}^n$, μ^* defines the coefficients of a basis $\mathbf{\Lambda}^* = \{\Lambda_1^*, \ldots, \Lambda_r^*\} \subset \mathbb{C}[\mathbf{d}_{\xi^*}]$ dual to the set $B_{\xi^*} = \{(\mathbf{x} - \xi^*)^{\beta_1}, \ldots, (\mathbf{x} - \xi^*)^{\beta_r}\} \subset \mathbb{C}[\mathbf{x}]$ and ϵ^* defines a perturbed polynomial system $\mathbf{f}_{\epsilon^*} := \mathbf{f} + \epsilon^* B_{\xi^*}$ with the property that ξ^* is an exact multiple root of $\mathbf{f}_{\epsilon^*}$ with inverse system $\mathbf{\Lambda}^*$. Furthermore, certify this property and give an upper bound on the size of the perturbation $\|\epsilon^*\|$.

The difficulty in solving Problem 1.1 is that known polynomial systems defining the coordinates of the roots and the inverse system are overdetermined, and we need a square subsystem of it in the Newton iterations to guarantee the existence of a root together with the quadratic convergence. Thus, roots of this square subsystem may not be exact roots of the complete polynomial system, and we cannot certify numerically that they are approximations of a root of the complete system. This is the reason why we introduce the variables ϵ that allow perturbation of the input system. One of the goals of the present paper is to understand what kind of perturbations are needed and to bound their magnitude.

Certifying the correctness of the multiplicity structure that the numerical iterations converge to poses a more significant challenge: the set of parameter values describing an affine point with multiplicity r forms a projective variety called the *punctual Hilbert scheme*. The goal is to certify that we converge to a point on this variety. We study an affine subset of the punctual Hilbert scheme and give a new description using multilinear quadratic equations that have a triangular structure. These equations appear in our deflated polynomial system, have integer coefficients, and have to be satisfied exactly without perturbation, otherwise the solution does not define a proper inverse system, closed under derivation. Fortunately, the structure allowed us to define a rational parametrization of a strata of the punctual Hilbert scheme, called the *regular* strata. In turn, this rational parametrization allows certification when converging to a point on this regular strata.

Our method comprises three parts: first, we apply the Integration Method (Algorithm 1) with input $\mathbf{f}$ and ξ to compute an approximation of the multiplicity structure, second, an analysis and certification part (see Section 6 and Algorithm 2), and third, a numerical iteration part converging to the exact multiple root with its multiplicity structure for an explicit perturbation of the input system (see Section 5). The missing proofs are available at hal.inria.fr/hal-02478768.

Related Work. There are many works in the literature studying the certification of isolated singular roots of polynomial systems. One approach

ISSAC '20, July 20–23, 2020, Kalamata, Greece

ACM ISBN 978-1-4503-7100-1/20/07…$15.00
https://doi.org/10.1145/3373207.3404024

is to give *separation bounds* for isolated roots, i.e. a bound that guarantees that there is exactly one root within a neighborhood of a given point. Worst case separation bounds for square polynomial systems with support in given polytopes and rational coefficients are presented in [10]. In the presence of singular roots, turned into root clusters after perturbations, these separation bounds separate the clusters from each other and bound the cluster size. [11, 32, 33] give separation bounds and numerical algorithms to compute clusters of zeroes of univariate polynomials. [8] extends α-theory and gives separation bounds for simple double zeroes of polynomial systems, [12] extend these results to zeroes of embedding dimension one.

Another approach, called deflation, comprises of transforming the singular root into a regular root of a new system and to apply certification techniques on the new system. [18] uses a square deflated system to prove the existence of singular solutions. [20] devises a deflation technique that adds new variables to the systems for isolated singular roots that accelerates Newton's method and [21] modifies this to compute the multiplicity structure. [28] computes error bounds that guarantee the existence of a simple double root within that error bound from the input, [22, 23] generalizes [28] to the breadth one case and give an algorithm to compute such error bound. [24] gives verified error bounds for isolated and some non-isolated singular roots using higher order deflations. [6, 7, 15, 30, 31, 34] give deflation techniques based on numerical linear algebra on the Macaulay matrices that compute the coefficients of the inverse system, with improvements using the closedness property of the dual space. [13, 14] give a new deflation method that does not introduce new variables and extends α-theory to general isolated multiple roots for the certification to a simple root of a subsystem of the overdetermined deflated system. In [16] a new deflated system is presented, its simple roots correspond to the isolated singular points with their multiplicity structure. A somewhat different approach is given in [1], where they use a symbolic-numeric certification techniques that certify that polynomial systems with rational coefficients have exact isolated singular roots. More recently, [19] design a square Newton iteration and provide separation bounds for roots when the deflation method of [20] terminates in one iteration, and give bounds for the size of the clusters.

The certification approach that we propose is based on an algebraic analysis of some strata of the punctual Hilbert scheme. Some of its geometric properties have been investigated long time ago, for instance in [4, 5, 17] or more recently in the plane [2]. However, as far as we know, the effective description that we use and the rational parametrization of the regular strata that we compute have not been developed previously.

2 PRELIMINARIES

Let $\mathbf{f} := (f_1, \ldots, f_N) \in \mathbb{C}[\mathbf{x}]^N$ with $\mathbf{x} = (x_1, \ldots, x_n)$. Let $\xi = (\xi_1, \ldots, \xi_n) \in \mathbb{C}^n$ be an isolated multiple root of $\mathbf{f}$. Let $I = \langle f_1, \ldots, f_N \rangle$, $\mathfrak{m}_\xi$ be the maximal ideal at ξ and Q be the primary component of I at ξ so that $\sqrt{Q} = \mathfrak{m}_\xi$. The shifted monomials at ξ will be denoted for $\alpha = (\alpha_1, \ldots, \alpha_n) \in \mathbb{N}^n$ by

$$\mathbf{x}_\xi^\alpha := (x_1 - \xi_1)^{\alpha_1} \cdots (x_1 - \xi_n)^{\alpha_n}.$$

Consider the ring of power series $\mathbb{C}[[\mathbf{d}_\xi]] := \mathbb{C}[[d_{1,\xi}, \ldots, d_{n,\xi}]]$ and we denote $\mathbf{d}_\xi^\beta := d_{1,\xi}^{\beta_1} \cdots d_{n,\xi}^{\beta_n}$, with $\beta = (\beta_1, \ldots, \beta_n) \in \mathbb{N}^n$. We identify $\mathbb{C}[[\mathbf{d}_\xi]]$ with the dual space $\mathbb{C}[\mathbf{x}]^*$ by considering the action of $\mathbf{d}_\xi^\beta$ on polynomials as derivations and evaluations at ξ, defined as

$$\mathbf{d}_\xi^\beta(p) := \boldsymbol{\partial}^\beta(p)\Big|_\xi = \frac{\partial^{|\beta|} p}{\partial x_1^{\beta_1} \cdots \partial x_n^{\beta_n}}(\xi) \quad \text{for } p \in \mathbb{C}[\mathbf{x}]. \tag{1}$$

Hereafter, we reserve the notation $\mathbf{d}$ and d_i for the dual variables while $\boldsymbol{\partial}$ and ∂_{x_i} for derivation. We indicate the evaluation at $\xi \in \mathbb{C}^n$ by writing $d_{i,\xi}$ and $\mathbf{d}_\xi$, and for $\xi = 0$ it will be denoted by $\mathbf{d}$. The derivation with respect to the variable $d_{i,\xi}$ in $\mathbb{C}[[\mathbf{d}_\xi]]$ is denoted $\partial_{d_{i,\xi}}$ $(i = 1, \ldots, n)$.

Observe that

$$\frac{1}{\beta!}\mathbf{d}_\xi^\beta((\mathbf{x} - \xi)^\alpha) = \begin{cases} 1 & \text{if } \alpha = \beta, \\ 0 & \text{otherwise,} \end{cases}$$

where $\beta! = \beta_1! \cdots \beta_n!$.

For $p \in \mathbb{C}[\mathbf{x}]$ and $\Lambda \in \mathbb{C}[[\mathbf{d}_\xi]] = \mathbb{C}[\mathbf{x}]^*$, let $p \cdot \Lambda : q \mapsto \Lambda(p\,q)$. We check that $p = (x_i - \xi_i)$ acts as a derivation on $\mathbb{C}[[\mathbf{d}_\xi]]$: $(x_i - \xi_i) \cdot \mathbf{d}_\xi^\beta = \partial_{d_{i,\xi}}(\mathbf{d}_\xi^\beta) = \beta_i \mathbf{d}_\xi^{\beta - \mathbf{e}_i}$. Throughout the paper we use the notation $\mathbf{e}_1, \ldots, \mathbf{e}_n$ for the standard basis of $\mathbb{C}^n$ or for a canonical basis of any vector space V of dimension n. We will also use integrals of polynomials in $\mathbb{C}[[\mathbf{d}_\xi]]$ as follows: for $\Lambda \in \mathbb{C}[[\mathbf{d}_\xi]]$ and $k = 1, \ldots, n$, $\int_k \Lambda$ denotes the polynomial $\Lambda^* \in \mathbb{C}[[\mathbf{d}_\xi]]$ such that $\partial_{d_{k,\xi}}(\Lambda^*) = \Lambda$ and Λ^* has no constant term. We introduce the following shorthand notation

$$\overline{\int_k} \Lambda := \int_k \Lambda(d_{1,\xi}, \ldots, d_{k,\xi}, 0, \ldots, 0). \tag{2}$$

For an ideal $I \subset \mathbb{C}[\mathbf{x}]$, let $I^\perp = \{\Lambda \in \mathbb{C}[[\mathbf{d}_\xi]] \mid \forall p \in I, \Lambda(p) = 0\}$. The vector space $I^\perp$ is naturally identified with the dual space of $\mathbb{C}[\mathbf{x}]/I$. We check that $I^\perp$ is a vector subspace of $\mathbb{C}[[\mathbf{d}_\xi]]$ which is closed under the derivations $\partial_{d_{i,\xi}}$ for $i = 1, \ldots, n$.

Lemma 2.1. *If Q is a $\mathfrak{m}_\xi$-primary isolated component of I, then $Q^\perp = I^\perp \cap \mathbb{C}[\mathbf{d}_\xi]$.*

This lemma shows that to compute $Q^\perp$, it suffices to compute all polynomials of $\mathbb{C}[\mathbf{d}_\xi]$ which are in $I^\perp$. Let us denote this set $\mathcal{D} = I^\perp \cap \mathbb{C}[\mathbf{d}_\xi]$. It is a vector space stable under the derivations $\partial_{d_{i,\xi}}$. Its dimension is the dimension of $Q^\perp$ or $\mathbb{C}[\mathbf{x}]/Q$, that is the *multiplicity* of ξ, denoted $r_\xi(I)$, or simply r if ξ and I is clear from the context.

For an element $\Lambda(\mathbf{d}_\xi) \in \mathbb{C}[\mathbf{d}_\xi]$ we define the degree or *order* $\operatorname{ord}(\Lambda)$ to be the maximal $|\beta|$ s.t. $\mathbf{d}_\xi^\beta$ appears in $\Lambda(\mathbf{d}_\xi)$ with non-zero coefficient.

For $t \in \mathbb{N}$, let $\mathcal{D}_t$ be the elements of $\mathcal{D}$ of order $\leq t$. As $\mathcal{D}$ is of dimension r, there exists a smallest $t \geq 0$ s.t. $\mathcal{D}_{t+1} = \mathcal{D}_t$. Let us call this smallest t, the *nil-index* of $\mathcal{D}$ and denote it by $\delta_\xi(I)$, or simply by δ. As $\mathcal{D}$ is stable by the derivations $\partial_{d_{i,\xi}}$, we easily check that for $t \geq \delta_\xi(I)$, $\mathcal{D}_t = \mathcal{D}$ and that $\delta_\xi(I)$ is the maximal degree of elements of $\mathcal{D}$.

Let $B = \{\mathbf{x}_\xi^{\beta_1}, \ldots, \mathbf{x}_\xi^{\beta_r}\}$ be a basis of $\mathbb{C}[\mathbf{x}]/Q$. We can identify the elements of $\mathbb{C}[\mathbf{x}]/Q$ with the elements of the vector space $\operatorname{span}_{\mathbb{C}}(B)$. We define the normal form $N(p)$ of a polynomial p in $\mathbb{C}[\mathbf{x}]$ as the unique element b of $\operatorname{span}_{\mathbb{C}}(B)$ such that $p - b \in Q$. Hereafter, we are going to identify the elements of $\mathbb{C}[\mathbf{x}]/Q$ with their normal form in $\operatorname{span}_{\mathbb{C}}(B)$. For $\alpha \in \mathbb{N}^n$, we will write the normal form of $\mathbf{x}_\xi^\alpha$ as

$$N(\mathbf{x}_\xi^\alpha) = \sum_{i=1}^r \mu_{\beta_i,\alpha}\, \mathbf{x}_\xi^{\beta_i}. \tag{3}$$

2.1 The multiplicity structure

We start this subsection by recalling the definition of graded primal-dual pairs of bases for the space $\mathbb{C}[\mathbf{x}]/Q$ and its dual. The following lemma defines the same dual space as in e.g. [6, 7, 23], but we emphasize on a primal-dual basis pair to obtain a concrete isomorphism between the coordinate ring and the dual space.

Lemma 2.2 (Graded primal-dual basis pair)**.** *Let $\mathbf{f}$, ξ, Q, $\mathcal{D}$, $r = r_\xi(\mathbf{f})$ and $\delta = \delta_\xi(\mathbf{f})$ be as above. Then there exists a primal-dual basis pair $(B, \boldsymbol{\Lambda})$ of the local ring $\mathbb{C}[\mathbf{x}]/Q$ with the following properties:*

(1) The primal basis *of the local ring $\mathbb{C}[\mathbf{x}]/Q$ has the form*

$$B := \left\{\mathbf{x}_\xi^{\beta_1}, \mathbf{x}_\xi^{\beta_2}, \ldots, \mathbf{x}_\xi^{\beta_r}\right\}. \tag{4}$$

We can assume that $\beta_1 = 0$ and that the ordering of the elements in B by increasing degree. Define the set of exponents in B as $E := \{\beta_1, \ldots, \beta_r\} \subset \mathbb{N}^n$.

(2) *The unique* dual basis $\mathbf{\Lambda} = \{\Lambda_1, \Lambda_2, \ldots, \Lambda_r\}$ *of* $\mathcal{D} \subset \mathbb{C}[\mathbf{d}_\xi]$ *dual to* B *has the form* $\Lambda_i = \frac{1}{\beta_i!}\mathbf{d}_\xi^{\beta_i} + \sum_{\substack{|\alpha|\le|\beta_i| \\ \alpha\notin E}} \mu_{\beta_i,\alpha}\ \frac{1}{\beta!}\mathbf{d}_\xi^{\alpha}$.

(3) *We have* $0 = \mathrm{ord}(\Lambda_1) \le \cdots \le \mathrm{ord}(\Lambda_r)$, *and for all* $0 \le t \le \delta$ *we have* $\mathcal{D}_t = \mathrm{span}\left\{\Lambda_j \ :\ \mathrm{ord}(\Lambda_j) \le t\right\}$, *where* $\mathcal{D}_t$ *denotes the elements of* $\mathcal{D}$ *of order* $\le t$, *as above.*

A graded primal-dual basis pair $(B, \mathbf{\Lambda})$ of $\mathcal{D}$ as described in Lemma 2.2 can be obtained from any basis $\tilde{\mathbf{\Lambda}}$ of $\mathcal{D}$ by first choosing pivot elements that are the leading monomials with respect to a graded monomial ordering on $\mathbb{C}[\mathbf{d}]$, these leading monomials define B, then transforming the coefficient matrix of $\tilde{\mathbf{\Lambda}}$ into row echelon form using the pivot leading coefficients, defining $\mathbf{\Lambda}$.

A monomial set B is called a *graded primal basis* of $\mathbf{f}$ at ξ if there exists $\mathbf{\Lambda} \subset \mathbb{C}[\mathbf{d}_\xi]$ such that $(B, \mathbf{\Lambda})$ is a graded primal-dual basis pair and $\mathbf{\Lambda}$ is complete for $\mathbf{f}$ at ξ.

Next we describe the so-called *integration method* introduced in [25, 27] that computes a graded pair of primal-dual bases as in Lemma 2.2 if the root ξ is given. The integration method performs the computation of a basis order by order. We need the following proposition, a new version of [27, Theorem 4.2]:

Proposition 2.3. *Let* $\Lambda_1, \ldots, \Lambda_s \in \mathbb{C}[\mathbf{d}_\xi]$ *and assume that* $\mathrm{ord}(\Lambda_i) \le t$ *for some* $t \in \mathbb{N}$. *Suppose that the subspace* $\mathcal{D} := \mathrm{span}(\Lambda_1, \ldots, \Lambda_s) \subset \mathbb{C}[\mathbf{d}_\xi]$ *is closed under derivation. Then* $\Delta \in \mathbb{C}[\mathbf{d}_\xi]$ *with no constant term satisfies* $\partial_{d_k}(\Delta) \in \mathcal{D}$ *for all* $k = 1, \ldots, n$ *if and only if* Δ *is of the form*

$$\Delta = \sum_{i=1}^{s} \sum_{k=1}^{n} v_i^k \overline{\int_k} \Lambda_i \tag{5}$$

for some $v_i^k \in \mathbb{C}$ *satisfying*

$$\sum_{i=1}^{s} v_i^k \partial_{d_l}(\Lambda_i) - v_i^l \partial_{d_k}(\Lambda_i) = 0 \ \text{ for } \ 1 \le k < l \le n. \tag{6}$$

Furthermore, (5) and (6) implies that

$$\partial_{d_k}(\Delta) = \sum_{i=1}^{s} v_i^k \Lambda_i \quad \text{for } k = 1, \ldots, n. \tag{7}$$

Let Q be a $\mathfrak{m}_\xi$-primary ideal. Proposition 2.3 implies that if $\mathbf{\Lambda} = \{\Lambda_1, \ldots, \Lambda_r\} \subset \mathbb{C}[\mathbf{d}_\xi]$ with $\Lambda_1 = 1_\xi$ is a basis of $Q^\perp$, dual to the basis $B = \{\mathbf{x}_\xi^{\beta_1}, \ldots, \mathbf{x}_\xi^{\beta_r}\} \subset \mathbb{C}[\mathbf{x}]$ of $\mathbb{C}[\mathbf{x}]/Q$ with $\mathrm{ord}(\Lambda_i) = |\beta_i|$, then there exist $v_{i,j}^k \in \mathbb{C}$ such that

$$\partial_{d_k}(\Lambda_i) = \sum_{|\beta_j|<|\beta_i|} v_{i,j}^k \Lambda_j.$$

Therefore, the matrix $\mathtt{M}_k$ of the multiplication map M_k by $x_k - \xi_k$ in the basis B of $\mathbb{C}[\mathbf{x}]/Q$ is

$$\mathtt{M}_k = [v_{j,i}^k]_{1\le i,j\le r}^T = [\mu_{\beta_i,\beta_j+\mathbf{e}_k}]_{1\le i,j\le r}$$

using the notation (3) and the convention that $v_{i,j}^k = \mu_{\beta_i,\beta_j+\mathbf{e}_k} = 0$ if $|\beta_i| \ge |\beta_j|$. Consequently,

$$v_{i,j}^k = \mu_{\beta_i,\beta_j+e_k} \quad i,j,= 1,\ldots,r, k = 1,\ldots,n,$$

and we have

$$\Lambda_i = \sum_{|\beta_j|<|\beta_i|} \sum_{k=1}^{n} \mu_{\beta_i,\beta_j+\mathbf{e}_k} \overline{\int_k} \Lambda_j$$

where $\mu_{\beta_i,\beta_j+\mathbf{e}_k}$ is the coefficient of $\mathbf{x}_\xi^{\beta_i}$ in the normal form of $\mathbf{x}_\xi^{\beta_j+\mathbf{e}_k}$ in the basis B of $\mathbb{C}[\mathbf{x}]/Q$.

Next we give a result that allows to simplify the linear systems involved in the integration method. We first need a definition:

Definition 2.4. Let $E \subset \mathbb{N}^n$ be a set of exponents. We say that E is *closed under division* if $\beta = (\beta_1, \ldots, \beta_n) \in E$ implies that $\beta - \mathbf{e}_k \in E$ as long as $\beta_k > 0$ for all $k = 1, \ldots, n$. We also call the corresponding primal basis $B = \{\mathbf{x}_\xi^{\beta_1}, \ldots, \mathbf{x}_\xi^{\beta_r}\}$ closed under division.

The following lemma provides a simple characterization of dual bases of inverse systems closed under derivation, that we will use in the integration algorithm.

Lemma 2.5. *Let* $B = \{\mathbf{x}_\xi^{\beta_1}, \ldots, \mathbf{x}_\xi^{\beta_r}\} \subset \mathbb{C}[\mathbf{x}]$ *be closed under division and ordered by degree. Let* $\mathbf{\Lambda} = \{\Lambda_1, \ldots, \Lambda_r\} \subset \mathbb{C}[\mathbf{d}_\xi]$ *be a linearly independent set such that*

$$\Lambda_i = \sum_{|\beta_j|<|\beta_i|} \sum_{k=1}^{n} \mu_{\beta_i,\beta_j+\mathbf{e}_k} \overline{\int_k} \Lambda_j. \tag{8}$$

Then $\mathcal{D} = \mathrm{span}\{\Lambda_1, \ldots, \Lambda_r\}$ *is closed under derivation iff for all* $i, s = 1, \ldots, r$, $|\beta_s| < |\beta_i|$ *and* $k \ne l \in \{1, \ldots, n\}$ *we have*

$$\sum_{j:|\beta_s|<|\beta_j|<|\beta_i|} \mu_{\beta_i,\beta_j+\mathbf{e}_k}\mu_{\beta_j,\beta_s+\mathbf{e}_l} - \mu_{\beta_i,\beta_j+\mathbf{e}_l}\mu_{\beta_j,\beta_s+\mathbf{e}_k} = 0. \tag{9}$$

Furthermore, $(B, \mathbf{\Lambda})$ *is a graded primal-dual basis pair iff they satisfy* (9) *and*

$$\mu_{\beta_i,\beta_j+\mathbf{e}_k} = \begin{cases} 1 & \text{for } \beta_i = \beta_j + \mathbf{e}_k \\ 0 & \text{for } \beta_j + \mathbf{e}_k \in E,\ \beta_i \ne \beta_j + \mathbf{e}_k, \end{cases} \tag{10}$$

To compute the inverse system $\mathcal{D}$ of $\mathbf{f}$ at a point ξ, we will consider the additional systems of equations in ξ and $\mu = \{\mu_{\beta_i,\alpha}\}$:

$$\Lambda_i(f_j) = 0 \text{ for } 1 \le i \le r, 1 \le j \le N. \tag{11}$$

Throughout the paper we use the following notation:

Notation 2.6. Let $f_1, \ldots, f_N \in \mathbb{C}[\mathbf{x}]$, $\xi \in \mathbb{C}^n$ and fix $t \in \mathbb{N}$. Let $B_{t-1} = \{\mathbf{x}_\xi^{\beta_1}, \ldots, \mathbf{x}_\xi^{\beta_{r_{t-1}}}\} \subset \mathbb{C}[\mathbf{x}_\xi]_{t-1}$ be closed under division and $\mathbf{\Lambda}_{t-1} = \{\Lambda_1, \ldots, \Lambda_{r_{t-1}}\} \subset \mathbb{C}[\mathbf{d}_\xi]_{t-1}$ dual to B_{t-1} with

$$\partial_{d_k}(\Lambda_j) = \sum_{|\beta_s|<|\beta_j|} \mu_{\beta_j,\beta_s+\mathbf{e}_k}\Lambda_s \quad j = 1,\ldots,r_{t-1}, k = 1,\ldots,n.$$

Consider the following homogeneous linear system of equations in the variables $\{v_j^k \ :\ j = 1,\ldots,r_{t-1},\ k = 1,\ldots,n\}$:

$$\sum_{j:|\beta_s|<|\beta_j|<t} v_j^k\ \mu_{\beta_j,\beta_s+\mathbf{e}_l} - v_j^l\ \mu_{\beta_j,\beta_s+\mathbf{e}_k} = 0, \quad 1 \le k < l \le n \tag{12}$$

$$v_j^k = 0 \quad \text{if } \beta_j + \mathbf{e}_k = \beta_l \text{ for some } 1 \le l \le r_{t-1} \tag{13}$$

$$\left(\sum_{j=1}^{r_{t-1}} \sum_{k=1}^{n} v_j^k \overline{\int_k} \Lambda_j\right)(f_l) = 0 \quad l = 1, \ldots, N. \tag{14}$$

We will denote by H_t the coefficient matrix of the equations in (12) and (13) and by K_t the coefficient matrix of the equations in (12)-(14).

Algorithm 1 produces incrementally a basis of $\mathcal{D}$, similarly to Macaulay's method. The algorithmic advantage is the smaller matrix size in $O(r\,n^2 + N)$ instead of $N\binom{n+\delta-1}{\delta}$, where δ is the maximal degree (depth) in the dual, cf. [16, 25].

The full Integration Method consists of taking $\Lambda_1 := 1_\xi$ for $t = 0$, a basis of $\mathcal{D}_0$ and then iterating algorithm Integration Method - Iteration t until we find a value of t when $\mathcal{D}_t = \mathcal{D}_{t-1}$. This implies that the order $\delta = \delta_\xi(\mathbf{f}) = t - 1$. This leads to the following definition.

Definition 2.7. We say that $\mathbf{\Lambda} \subset \mathbb{C}[\mathbf{d}_\xi]$ is *complete* for $\mathbf{f}$ at ξ if the linear system K_t of the equations (12)-(14) in degree $t = \delta + 1 = \mathrm{ord}(\mathbf{\Lambda}) + 1$ is such that $\ker K_{\delta+1} = \{0\}$.

Notice that the full Integration Method constructs a graded primal-dual basis pair $(B, \mathbf{\Lambda})$. The basis $\mathbf{\Lambda} \subset (\mathbf{f})^\perp$ spans a space stable by derivation and is complete for $\mathbf{f}$, so that we have $\mathrm{span}(\mathbf{\Lambda}) = (\mathbf{f})^\perp \cap \mathbb{C}[\mathbf{d}_\xi] = Q^\perp$ where Q is the primary component of $(\mathbf{f})$ at ξ.

To guarantee that B_t is closed under division, one could choose a graded monomial ordering $\prec$ of $\mathbb{C}[\mathbf{d}_\xi]$ and compute an auto-reduced basis of $\ker K_t$ such that the initial terms for $\prec$ are $\mathbf{d}_\xi^{\beta_i}$. The set B_t constructed in this way would be closed under division, since $\mathcal{D}_t$ is stable under derivation. In the approach we use in practice, we choose the column pivot taking into account the numerical values of the coefficients

Algorithm 1 Integration Method - Iteration t

Input: $t > 0$, $\mathbf{f} = (f_1, \ldots, f_N) \in \mathbb{C}[\mathbf{x}]^N$, $\xi \in \mathbb{C}^n$, $B_{t-1} = \{\mathbf{x}_\xi^{\beta_1}, \ldots, \mathbf{x}_\xi^{\beta_{r_{t-1}}}\} \subset \mathbb{C}[\mathbf{x}]$ closed under division and $\mathbf{\Lambda}_{t-1} = \{\Lambda_1, \ldots, \Lambda_{r_{t-1}}\} \subset \mathbb{C}[\mathbf{d}_\xi]$ a basis for $\mathcal{D}_{t-1}$ dual to B_{t-1}, of the form (8).

Output: Either "$\mathcal{D}_t = \mathcal{D}_{t-1}$" or $B_t = \{\mathbf{x}_\xi^{\beta_1}, \ldots, \mathbf{x}_\xi^{\beta_{r_t}}\}$ for some $r_t > r_{t-1}$ closed under division and $\mathbf{\Lambda}_t = \{\Lambda_1, \ldots, \Lambda_{r_t}\}$ with Λ_i of the form (8), satisfying (9), (10) and (11).

(1) Set up the coefficient matrix K_t of the homogeneous linear system (12)-(14) in Notation 2.6 in the variables $\{v_j^k\}_{j=1,\ldots,r_{t-1},\ k=1,\ldots,n}$ associated to an element of the form $\Lambda = \sum_{j=1}^{r_{t-1}} \sum_{k=1}^{n} v_j^k \overline{\partial}_k \Lambda_j$. Let $h_t := \dim \ker K_t$.

(2) If $h_t = 0$ then return "$\mathcal{D}_t = \mathcal{D}_{t-1}$". If $h_t > 0$ define $r_t := r_{t-1} + h_t$. Perform a triangulation of K_t by row reductions with row permutations and column pivoting so that the non-pivoting columns correspond to exponents $\beta_{r_{t-1}+1}, \ldots, \beta_{r_t}$ with strict divisors in B_{t-1}. Let $B_t = B_{t-1} \cup \{\mathbf{x}_\xi^{\beta_{r_{t-1}+1}}, \ldots, \mathbf{x}_\xi^{\beta_{r_t}}\}$.

(3) Compute a basis $\Lambda_{r_{t-1}+1}, \ldots, \Lambda_{r_t} \in \mathbb{C}[\mathbf{d}_\xi]$ of $\ker K_t$ from the triangular reduction of K_t by setting the coefficients of the non-pivoting columns to 0 or 1. This yields a basis $\mathbf{\Lambda}_t = \mathbf{\Lambda}_{t-1} \cup \{\Lambda_{r_{t-1}+1}, \ldots, \Lambda_{r_t}\}$ dual to B_t. The coefficients $v_{i,j}^k$ of Λ_i are $\mu_{\beta_i, \beta_j + \mathbf{e}_k}$ in (8) so that Eq. (11) are satisfied. Eq. (10) are satisfied, since $\mathbf{\Lambda}_t$ is dual to B_t.

and not according to a monomial ordering and we check a posteriori that the set of exponents is closed under division (See Example 7.1).

The main property that we will use for the certification of multiplicities is given in the next theorem.

Theorem 2.8. *If ξ^* is an isolated solution of the system $\mathbf{f}(\mathbf{x}) = 0$ and B is a graded primal basis at ξ^* closed under division, then the system $F(\xi, \mu) = 0$ of all equations* (9), (10) *and* (11) *admits (ξ^*, μ^*) as an isolated simple root, where μ^* defines the basis $\mathbf{\Lambda}^*$ of the inverse system of $(\mathbf{f})$ at ξ dual to B, due to* (8).

3 PUNCTUAL HILBERT SCHEME

The results in Sections 3 and 4 do not depend on the point $\xi \in \mathbb{C}^n$, so to simplify the notation, we assume in these sections that $\xi = \mathbf{0}$. Let $\mathfrak{m} = (x_1, \ldots, x_n)$ be the maximal ideal defining $\xi = \mathbf{0} \in \mathbb{C}^n$. Let $\mathbb{C}[\mathbf{d}]$ be the space of polynomials in the variables $\mathbf{d} = (d_1, \ldots, d_n)$ and $\mathbb{C}[\mathbf{d}]_t \subset \mathbb{C}[\mathbf{d}]$ the subspace of polynomials in $\mathbf{d}$ of degree $\leq t$.

For a vector space V, let $\mathscr{G}_r(V)$ be the projective variety of the r dimensional linear subspaces of V, also known as the *Grassmannian* of r-spaces of V. The points in $\mathscr{G}_r(V)$ are the projective points of $\mathbb{P}(\wedge^r V)$ of the form $\mathbf{v} = v_1 \wedge \cdots \wedge v_r$ for $v_i \in V$. Fixing a basis $\mathbf{e}_1, \ldots, \mathbf{e}_s$ of V, the Plücker coordinates of $\mathbf{v}$ are the coefficients of $\Delta_{i_1,\ldots,i_r}(\mathbf{v})$ of $\mathbf{v} = \sum_{i_1 < \cdots < i_r} \Delta_{i_1,\ldots,i_r}(\mathbf{v})\, \mathbf{e}_{i_1} \wedge \cdots \wedge \mathbf{e}_{i_r}$. When $V = \mathbb{C}[\mathbf{d}]_{r-1}$, a natural basis is the dual monomial basis $(\frac{\mathbf{d}^\alpha}{\alpha!})_{|\alpha|<r}$. The Plücker coordinates of an element $\mathbf{v} \in \mathscr{G}_r(\mathbb{C}[\mathbf{d}]_{r-1})$ for this basis are denoted $\Delta_{\alpha_1,\ldots,\alpha_r}(\mathbf{v})$ where $\alpha_i \in \mathbb{N}^n$, $|\alpha_i| < r$.

If $\mathbf{\Lambda} = \{\Lambda_1, \ldots, \Lambda_r\}$ is a basis of a r-dimensional space $\mathcal{D}$ in $\mathbb{C}[\mathbf{d}]_{r-1}$ with $\Lambda_i = \sum_{|\alpha|<r} \mu_{i,\alpha} \frac{\mathbf{d}^\alpha}{\alpha!}$, the Plücker coordinates of $\mathcal{D}$ are, up to a scalar, of the form $\Delta_{\alpha_1,\ldots,\alpha_r} = \det\left[\mu_{i,\alpha_j}\right]_{1 \leq i,j \leq r}$. In particular, a monomial set $B = \{\mathbf{x}^{\beta_1}, \ldots, \mathbf{x}^{\beta_r}\} \subset \mathbb{C}[\mathbf{x}]_{r-1}$ has a dual basis in $\mathcal{D}$ iff $\Delta_{\beta_1,\ldots,\beta_r}(\mathcal{D}) \neq 0$. If $(B = \{\mathbf{x}^{\beta_i}\}_{i=1}^r, \mathbf{\Lambda} = \{\Lambda_i\}_{i=1}^r)$ is a graded primal-dual basis pair, then $\mu_{i,\beta_j} = \delta_{i,j}$. To keep our notation consistent with the previous sections, the coordinates of $\Lambda_i \in \mathbf{\Lambda}$ when $\mathbf{\Lambda}$ is dual to B will be denoted by $\mu_{\beta_i,\alpha}$ instead of $\mu_{i,\alpha}$. By properties of the determinant, the Plücker coordinates of $\mathcal{D}$ are such that

$$\mu_{\beta_i,\alpha} = \frac{\Delta_{\beta_1,\ldots,\beta_{i-1},\alpha,\beta_{i+1},\ldots,\beta_r}}{\Delta_{\beta_1,\ldots,\beta_r}} \quad i = 1, \ldots, r. \tag{15}$$

If $\mathcal{D}$ is the dual of an ideal $Q = \mathcal{D}^\perp \subset \mathbb{C}[\mathbf{x}]$ and $B = \{\mathbf{x}^{\beta_1}, \ldots, \mathbf{x}^{\beta_r}\}$ is a basis of $\mathbb{C}[\mathbf{x}]/Q$ so that $\Delta_{\beta_1,\ldots,\beta_r}(\mathcal{D}) \neq 0$, the normal form of $\mathbf{x}^\alpha \in \mathbb{C}[\mathbf{x}]_{r-1}$ modulo $Q = \mathcal{D}^\perp$ in the basis B is

$$N(\mathbf{x}^\alpha) = \sum_{j=1}^{r} \mu_{\beta_j,\alpha}\, \mathbf{x}^{\beta_j} = \sum_{j=1}^{r} \frac{\Delta_{\beta_1,\ldots,\beta_{j-1},\alpha,\beta_{j+1},\ldots,\beta_r}}{\Delta_{\beta_1,\ldots,\beta_r}} \mathbf{x}^{\beta_j}.$$

(if $\deg(\mathbf{x}^\alpha) \geq r$, then $N(\mathbf{x}^\alpha) = 0$).

Definition 3.1. Let $\mathscr{H}_r \subset \mathscr{G}_r(\mathbb{C}[\mathbf{d}]_{r-1})$ be the set of linear spaces $\mathcal{D}$ of dimension r in $\mathbb{C}[\mathbf{d}]_{r-1}$ which are stable by the derivations ∂_{d_i} with respect to the variables $\mathbf{d}$ (i.e. $\partial_{d_i}\mathcal{D} \subset \mathcal{D}$ for $i = 1, \ldots, n$). We called $\mathscr{H}_r$ the *punctual Hilbert scheme* of points of multiplicity r.

If $\mathcal{D} \subset \mathbb{C}[\mathbf{d}]$ is stable by the derivations ∂_{d_i}, then by duality $I = \mathcal{D}^\perp \subset \mathbb{C}[\mathbf{x}]$ is a vector space of $\mathbb{C}[\mathbf{x}]$ stable by multiplication by x_i, i.e. an ideal of $\mathbb{C}[\mathbf{x}]$.

Proposition 3.2. *$\mathcal{D} \in \mathscr{H}_r$ iff $\mathcal{D}^\perp = Q$ is an $\mathfrak{m}$-primary ideal such that* $\dim \mathbb{C}[\mathbf{x}]/Q = r$.

Proof. Let $\mathcal{D} \in \mathscr{H}_r$. We prove that $\mathcal{D}^\perp = Q$ is an $\mathfrak{m}$-primary ideal. As $\mathcal{D}$ is stable by derivation, $Q = \mathcal{D}^\perp$ is an ideal of $\mathbb{C}[\mathbf{x}]$. This also implies that $1 \in \mathcal{D}$, so that $Q \subset \mathfrak{m}$. As $\dim \mathcal{D} = \dim \mathbb{C}[\mathbf{x}]/Q = r$, $\delta = \mathrm{ord}(\mathcal{D})$ is finite and $\mathfrak{m}^{\delta+1} \subset \mathcal{D}^\perp = Q$. Therefore, Q is $\mathfrak{m}$-primary, which shows the first implication.

Conversely, let Q be a $\mathfrak{m}$-primary ideal such that $\dim \mathbb{C}[\mathbf{x}]/Q = r$. Then by Lemma 2.1, $\mathcal{D} = Q^\perp \subset \mathbb{C}[\mathbf{d}]_t$ is stable by derivation and of dimension $r = \dim \mathbb{C}[\mathbf{x}]/Q$. Thus $\mathcal{D} \in \mathscr{H}_r$. This concludes the proof of the proposition. □

For $\mathcal{D} \in \mathscr{H}_r$, for $t \geq 0$ we denote by $\mathcal{D}_t$ the vector space of elements of $\mathcal{D}$ of order $\leq t$. We verify that $\mathcal{D}_t^\perp = \mathcal{D}^\perp + \mathfrak{m}^{t+1}$. The next theorem follows from Proposition 2.3 and Lemma 2.5.

Theorem 3.3. *For $B \subset \mathbb{C}[\mathbf{x}]$ closed under division such that $|B| = r$ and $\delta = \deg(B)$, the following points are equivalent:*

1. *$\mathcal{D} \in \mathscr{H}_r$ and B_t is a basis of $\mathbb{C}[\mathbf{x}]/(\mathcal{D}^\perp + \mathfrak{m}^{t+1})$ for $t = 1, \ldots, \delta$.*
2. *The dual basis $\mathbf{\Lambda} = \{\Lambda_1, \ldots, \Lambda_r\}$ of B satisfies $\Lambda_1 = 1$ and the equations* (8), (9) *and* (10).

For a sequence $\mathbf{h} = (h_0, h_1, \ldots, h_\delta) \in \mathbb{N}_+^{\delta+1}$ and $0 \leq t \leq \delta$, let $\mathbf{h}_t = (h_0, \ldots, h_t)$, $r_t = \sum_{i=0}^t h_i$. For $r \geq 1$ we denote by $\mathbf{S}^r$ the set of sequences $\mathbf{h}$ of some length $\delta < r$ with $h_i \neq 0$, $h_0 = 1$ and $r_\delta = r$. For $\mathbf{h} \in \mathbf{S}^r$, we consider the following subvarieties of $\mathscr{H}_{r_t}$:

$$\mathscr{H}_{\mathbf{h}_t} = \{\mathcal{D} \in \mathscr{H}_{r_t} \mid \dim \mathcal{D}_i = \dim \mathcal{D} \cap \mathbb{C}[\mathbf{d}]_i \leq r_i, i = 0, \ldots, t\}.$$

These are projective varieties in $\mathscr{H}_{r_t}$ defined by rank conditions on the linear spaces $\mathcal{D} \cap \mathbb{C}[\mathbf{d}]_i$ for $\mathcal{D} \in \mathscr{H}_{r_t}$, that can be expressed in terms of homogeneous polynomials in the Plücker coordinates of $\mathcal{D}$. In particular, the varieties $\mathscr{H}_{\mathbf{h}} := \mathscr{H}_{\mathbf{h}_\delta}$ are projective subvarieties of $\mathscr{H}_r$. They may not be irreducible or irreducible components of $\mathscr{H}_r$, but we have $\mathscr{H}_r = \cup_{\mathbf{h} \in \mathbf{S}^r} \mathscr{H}_{\mathbf{h}}$.

We will study a particular component of $\mathscr{H}_{\mathbf{h}}$, that we call the *regular component of* $\mathscr{H}_{\mathbf{h}}$, denoted $\mathscr{H}_{\mathbf{h}}^{reg}$. It is characterized as follows. Let $\mathscr{H}_{\mathbf{h}_0}^{reg} = \{\langle 1 \rangle\} = \{\mathbb{C}[\mathbf{d}]_0\} = \mathscr{G}_1(\mathbb{C}[\mathbf{d}]_0)$ and assume that $\mathscr{H}_{\mathbf{h}_{t-1}}^{reg}$ has been defined as an irreducible component of $\mathscr{H}_{\mathbf{h}_{t-1}}$. Let

$$W_t = \{(\mathcal{D}_{t-1}, \mathcal{E}_t) \mid \mathcal{D}_{t-1} \in \mathscr{H}_{\mathbf{h}_{t-1}}, \mathcal{E}_t \in \mathscr{G}_{r_t}(\mathbb{C}[\mathbf{d}]_t), \mathcal{D}_{t-1} \subset \mathcal{E}_t, \forall i\ \partial_{d_i}\mathcal{E}_t \subset \mathcal{D}_{t-1}\}$$

The constraints $\mathcal{D}_{t-1} \subset \mathcal{E}_t$ and $\partial_{d_i}\mathcal{E}_t \subset \mathcal{D}_{t-1}$ for $i = 1, \ldots, n$ define a linear system of equations in the Plücker coordinates of $\mathcal{E}_t$ (see e.g. [9]), corresponding to the equations (5), (6). By construction, the projection of

$W_t \subset \mathscr{H}_{\mathbf{h}_{t-1}} \times \mathscr{G}_{r_t}(\mathbb{C}[\mathbf{d}]_t)$ on the second factor $\mathscr{G}_{r_t}(\mathbb{C}[\mathbf{d}]_t)$ is $\pi_2(W_t) = \mathscr{H}_{\mathbf{h}_t}$ and the projection on the first factor is $\pi_1(W_t) = \mathscr{H}_{\mathbf{h}_{t-1}}$.

There exists a dense subset U_{t-1} of the irreducible variety $\mathscr{H}^{reg}_{\mathbf{h}_{t-1}}$ (with $\overline{U_{t-1}} = \mathscr{H}^{reg}_{\mathbf{h}_{t-1}}$) such that the rank of the linear system corresponding to (5) and (6) defining $\mathcal{E}_t$ is maximal. Since $\pi_1^{-1}(\mathcal{D}_{t-1})$ is irreducible (in fact linear) of fixed dimension for $\mathcal{D}_{t-1} \in U_{t-1} \subset \mathscr{H}^{reg}_{\mathbf{h}_{t-1}}$, there is a unique irreducible component $W_{t,reg}$ of W_t such that $\pi_1(W_{t,reg}) = \mathscr{H}^{reg}_{\mathbf{h}_{t-1}}$ (see eg. [29][Theorem 1.26]). We define $\mathscr{H}^{reg}_{\mathbf{h}_t} = \pi_2(W_{t,reg})$. It is an irreducible component of $\mathscr{H}_{\mathbf{h}_t}$, since otherwise $W_{t,reg} = \pi_2^{-1}(\mathscr{H}^{reg}_{\mathbf{h}_t})$ would not be a component of W_t but strictly included in one of the irreducible components of W_t.

Definition 3.4. Let $\pi_t : \mathscr{H}_{\mathbf{h}_t} \to \mathscr{H}_{\mathbf{h}_{t-1}}$, $\mathcal{D} \mapsto \mathcal{D} \cap \mathbb{C}[\mathbf{d}]_{t-1}$ be the projection in degree $t-1$. We define by induction on t, $\mathscr{H}^{reg}_{\mathbf{h}_0} = \{\langle 1\rangle\}$ and $\mathscr{H}^{reg}_{\mathbf{h}_t}$ is the irreducible component $\pi_t^{-1}(\mathscr{H}^{reg}_{\mathbf{h}_{t-1}})$ of $\mathscr{H}_{\mathbf{h}_t}$ for $t = 1,\dots,\delta$.

4 RATIONAL PARAMETRIZATION

Let $B = \{\mathbf{x}^{\beta_1},\dots,\mathbf{x}^{\beta_r}\} \subset \mathbb{C}[\mathbf{x}]_{r-1}$ be a monomial set. In this section we assume that B is closed under division and its monomials are ordered by increasing degree. For $t \in \mathbb{N}$, we denote by $B_t = B \cap \mathbb{C}[\mathbf{x}]_t$, by $B_{[t]}$ the subset of its monomials of degree t. Let $h_t = |B_{[t]}|$, $r_t = \sum_{0\le i\le t} h_t = |B_t|$ and $\delta = \deg(B)$.

Let $\mathscr{H}_B := \{\mathcal{D} \in \mathscr{H}_r \mid B_t \text{ is a basis of } \mathbb{C}[\mathbf{x}]/(\mathcal{D}^\perp + \mathfrak{m}^{t+1}), t = 0,\dots,\delta\}$. By Theorem 3.3, $\mathscr{H}_B$ is the set of linear spaces $\mathcal{D} \in \mathscr{H}_r$ such that $\mathcal{D}_t = \mathcal{D} \cap \mathbb{C}[\mathbf{d}]_t$ satisfy Equations (8) and (9). It is the open subset of $\mathcal{D} \in \mathscr{H}_{\mathbf{h}}$ such that $\Delta_{B_t}(\mathcal{D}_t) \neq 0$ for $t = 1,\dots,\delta$, where $\Delta_{B_t} := \Delta_{\beta_1,\dots,\beta_{r_t}}$ denotes the Plücker coordinate for $\mathscr{G}_{r_t}(\mathbb{C}[\mathbf{d}]_t)$ corresponding to the monomials in B_t.

Since for $\mathcal{D} \in \mathscr{H}_B$ we have $\Delta_B(\mathcal{D}) \neq 0$, we can define the affine coordinates of $\mathscr{H}_B$ using the coordinates of the elements of the basis $\mathbf{\Lambda} = \{\Lambda_1,\dots,\Lambda_r\}$ dual to B:

$$\left\{\mu_{\beta_j,\alpha} = \frac{\Delta_{\beta_1,\dots,\beta_{j-1},\alpha,\beta_{j+1},\dots,\beta_r}}{\Delta_B} \;:\; j = 1,\dots,r,\ |\alpha| < r\right\}.$$

The following lemma shows that the values of the coordinates $\{\mu_{\beta_i,\beta_j+\mathbf{e}_k} : i,j = 1,\dots r, |\beta_j| < |\beta_i|, k = 1,\dots,n\}$ uniquely define $\mathbf{\Lambda}$.

Lemma 4.1. *Let $B = \{\mathbf{x}^{\beta_1},\dots,\mathbf{x}^{\beta_{r_t}}\}$ closed under division, $\mathcal{D} \in \mathscr{H}_B$ and $\mathbf{\Lambda} = \{\Lambda_1,\dots,\Lambda_r\}$ be the unique basis of $\mathcal{D}$ dual to B with $\Lambda_i = \sum_{|\alpha|\le|\beta_i|} \mu_{\beta_i,\alpha} \frac{\mathbf{d}^\alpha}{\alpha!}$ for $i = 1,\dots,r$. Then $\Lambda_1 = 1$ and for $i = 2,\dots,r$*

$$\Lambda_i = \sum_{|\beta_j|<|\beta_i|} \sum_{k=1}^{n} \mu_{\beta_i,\beta_j+\mathbf{e}_k} \overline{\int_k} \Lambda_j.$$

Thus, $\mu_{\beta_i,\alpha}$ is a polynomial function of $\{\mu_{\beta_s,\beta_j+\mathbf{e}_k} : |\beta_s| \le |\beta_i|, |\beta_j| < |\beta_s|, k = 1,\dots,n\}$ for $i = 1,\dots,r$, $|\alpha| < |\beta_i|$.

Proof. Since $\mathcal{D}$ is closed under derivation, by Proposition 2.3 there exist $c_{i,s,k} \in \mathbb{C}$ such that $\partial_{d_k}(\Lambda_i) = \sum_{|\beta_s|<|\beta_i|} c_{i,s,k}\Lambda_s$. Then

$$\mu_{\beta_i,\beta_j+\mathbf{e}_k} = \Lambda_i(\mathbf{x}^{\beta_j+\mathbf{e}_k}) = \partial_{d_k}(\Lambda_i)(\mathbf{x}^{\beta_j}) = \sum_{|\beta_s|<|\beta_i|} c_{i,s,k}\Lambda_s(\mathbf{x}^{\beta_j}) = c_{i,j,k}.$$

The second claim follows from obtaining the coefficients in Λ recursively from $\Lambda_1 = 1$ and $\Lambda_i = \sum_{|\beta_j|<|\beta_i|} \sum_{k=1}^{n} \mu_{\beta_i,\beta_j+\mathbf{e}_k} \overline{\int_k} \Lambda_j$, for $i = 2,\dots,r$. □

We define $\mu := \{\mu_{\beta_i,\beta_j+\mathbf{e}_k}\}_{i,j=1,\dots r,|\beta_j|<|\beta_i|,k=1,\dots,n}$, $\mu_t := \{\mu_{\beta_i,\beta_j+\mathbf{e}_k} \in \mu \;:\; |\beta_i| \le t\} \subset \mu$ and $\mu_{[t]} := \{\mu_{\beta_i,\beta_j+\mathbf{e}_k} \in \mu \;:\; |\beta_j| = t\} \subset \mu_t$. The next definition uses the fact that Equations (12) and (13) are linear in v_j^k with coefficients depending on μ_{t-1}:

Definition 4.2. Given $\mathcal{D}_{t-1} \in \mathscr{H}_{B_{t-1}}$ with a unique basis $\mathbf{\Lambda}_{t-1} = \{\Lambda_1,\dots,\Lambda_{r_{t-1}}\}$ with $\Lambda_i = \sum_{|\alpha|<t} \mu_{\beta_i,\alpha} \frac{\mathbf{d}^\alpha}{\alpha!}$ for $j = 1,\dots,r_{t-1}$ that is dual to B_{t-1}, uniquely determined by $\mu_{t-1} = \{\mu_{\beta_i,\beta_j+\mathbf{e}_k} \;:\; |\beta_i| \le t-1, |\beta_j| < |\beta_i|\}$ as above. Recall from Notation 2.6 that H_t is the coefficient matrix of the homogeneous linear system (12) and (13) in the variables $\{v_j^k \;:\; j = 1,\dots,r_{t-1},\ k = 1,\dots,n\}$. To emphasize the dependence of its coefficients on $\mathcal{D}_{t-1}$ or μ_{t-1} we use the notation $H_t(\mathcal{D}_{t-1})$ or $H_t(\mu_{t-1})$. For $\mathcal{D} \in \mathscr{H}^{reg}_{\mathbf{h}}$ in an open subset, the rank ρ_t of $H_t(\mathcal{D}_{t-1})$ is maximal.

The next definition describes a property of a monomial set B such that it will allow us to give a rational parametrization of $\mathscr{H}_B$.

Definition 4.3. For $t = 1,\dots,\delta = \deg(B)$ we say that $\mathcal{D}_t \in \mathscr{G}_{r_t}(\mathbb{C}[\mathbf{d}]_t)$ is regular for B_t if,

- $\dim(\mathcal{D}_t) = r_t = |B_t|$,
- $\operatorname{rank} H_t(\mathcal{D}_{t-1}) = \rho_t$ the generic rank of H_t on $\mathscr{H}^{reg}_{\mathbf{h}_t}$,
- $\Delta_{B_{[t]}}(\mathcal{D}_{[t]}) \neq 0$ where $\Delta_{B_{[t]}}(\mathcal{D}_{[t]})$ is the Plücker coordinate of $\mathcal{D}_{[t]} \in \mathscr{G}_{h_t}(\mathbb{C}[\mathbf{d}]_r)$ corresponding to the monomials in $B_{[t]}$.

Let $U_t := \{\mathcal{D}_t \in \mathscr{H}^{reg}_{\mathbf{h}_t} \;:\; \mathcal{D}_t \text{ is regular for } B_t\}$. Then U_t is either an open dense subset of the irreducible variety $\mathscr{H}^{reg}_{\mathbf{h}_t}$ or empty if $\Delta_{B_{[t]}}(\mathcal{D}_{[t]}) = 0$ for all $\mathcal{D} \in \mathscr{H}^{reg}_{\mathbf{h}_t}$. We say that B is a *regular basis* if $\overline{U_t} = \mathscr{H}^{reg}_{\mathbf{h}_t}$ (or $U_t \neq \emptyset$) for $t = 1,\dots,\delta$.

We denote by $\gamma_{[t]} = \dim \mathscr{G}_{h_t}(\ker H_t(\mathcal{D}_{t-1}))$ for $\mathcal{D}_{t-1} \in U_{t-1}$ and $\gamma = \sum_{t=0}^{\delta} \gamma_{[t]}$.

If the basis B is regular and closed under division, then $\mathscr{H}^{reg}_{\mathbf{h}}$ can be parametrized by rational functions of free parameters $\overline{\mu}$. We present hereafter Algorithm 2 to compute such a parametrization iteratively.

Algorithm 2 Rational Parametrization - Iteration t

Input: $t > 0$, $B_t = \{\mathbf{x}^{\beta_1},\dots,\mathbf{x}^{\beta_{r_t}}\} \subset \mathbb{C}[\mathbf{x}]_t$ closed under division and regular, $\overline{\mu}_{t-1} \subset \mu_{t-1}$ and $\Phi_{t-1} : \overline{\mu}_{t-1} \mapsto \left(q_{\beta_j,\alpha}(\overline{\mu}_{t-1})\right)_{|\beta_j|\le t-1,|\alpha|<r}$ with $q_{\beta_j,\alpha} \in \mathbb{Q}(\overline{\mu}_{t-1})$ parametrizing a dense subset of $\mathscr{H}^{reg}_{\mathbf{h}_{t-1}}$.
Output: $\overline{\mu}_t \subset \mu_t$ and $\Phi_t : \overline{\mu}_t \mapsto \left(q_{\beta_j,\alpha}\right)_{|\beta_j|\le t,|\alpha|<r}$, $q_{\beta_j,\alpha} \in \mathbb{Q}(\overline{\mu}_t)$ extending Φ_{t-1} and parametrizing a dense subset of $\mathscr{H}^{reg}_{\mathbf{h}_t}$.

(1) Let H_t be as in Notation 2.6, $v = [v_j^k : j = 1,\dots,r_{t-1}, k = 1,\dots,n]^T$. Decompose $H_t(\Phi_{t-1}(\overline{\mu}_{t-1})) \cdot v = 0$ as

$$\left[\begin{array}{c|c|c} A(\overline{\mu}_{t-1}) & B(\overline{\mu}_{t-1}) & C(\overline{\mu}_{t-1}) \end{array}\right] \begin{bmatrix} v' \\ v'' \\ \overline{v} \end{bmatrix} = 0, \tag{16}$$

where v' is associated to a maximal set of independent columns of $H_t(\Phi_{t-1}(\overline{\mu}_{t-1}))$, $v'' = \{v_j^k : \mathbf{x}^{\beta_j+\mathbf{e}_k} \in B_{[t]}\}$ and $\overline{v}$ refers to the rest of the columns. If no such decomposition exists, return "B_t is not regular".
(2) For $v_j^k \in v'$ express $v_j^k = \varphi_j^k(\overline{v}, v'') \in \mathbb{Q}(\overline{\mu}_{t-1})[\overline{v}, v'']_1$ as the generic solution of the system $H_t(\Phi_{t-1}(\overline{\mu}_{t-1})) \cdot v = 0$.
(3) For $i = r_{t-1}+1,\dots,r_t$ do:

(3.1) Define $\overline{\mu}_{[t],i} := \left\{\mu_{\beta_i,\beta_j+\mathbf{e}_k} \;:\; v_{j,k} \in \overline{v}\right\}$, $\mu'_{[t],i} = \{\mu_{\beta_i,\beta_j+\mathbf{e}_k} : v_j^k \in v'\}$, $\mu''_{[t],i} = \{\mu_{\beta_i,\beta_j+\mathbf{e}_k} : v_j^k \in v''\}$, and $\overline{\mu}_t := \overline{\mu}_{t-1} \cup \bigcup_{i=r_{t-1}+1}^{r_t} \overline{\mu}_{[t],i}$.
(3.2) For $\mu_{\beta_i,\beta_j+\mathbf{e}_k} \in \mu''_{[t],i}$ set $q_{\beta_i,\beta_j+\mathbf{e}_k} = \mu_{\beta_i,\beta_j+\mathbf{e}_k} = 1$ if $\beta_i = \beta_j + \mathbf{e}_k$ and 0 otherwise.
(3.3) For $\mu_{\beta_i,\beta_j+\mathbf{e}_k} \in \mu'_{[t],i}$ define

$$q_{\beta_i,\beta_j+\mathbf{e}_k} := \varphi_j^k(\overline{\mu}_{[t],i}, \mu''_{[t],i}) \in \mathbb{Q}(\overline{\mu}_t)$$

(3.4) For $|\alpha| < r$ and $\mu_{\beta_i,\alpha} \notin \mu_t$ find $q_{\beta_i,\alpha}$ using Lemma 4.1.

Proposition 4.4. *Let $B = \{\mathbf{x}^{\beta_1},\dots,\mathbf{x}^{\beta_r}\} \subset \mathbb{C}[\mathbf{x}]_{r-1}$ be closed under division and assume that B is a regular basis. There exist a subset $\overline{\mu} \subset \mu$*

with $|\overline{\mu}| = \gamma$ and rational functions $q_{\beta_j,\alpha}(\overline{\mu}) \in \mathbb{Q}(\overline{\mu})$ for $j = 1,\dots,r$ and $|\alpha| < r$, such that the map $\Phi : \mathbb{C}^\gamma \to \mathscr{H}_B$ defined by

$$\Phi : \overline{\mu} \mapsto \left(q_{\beta_j,\alpha}(\overline{\mu})\right)_{j=1,\dots,r,|\alpha|<r}$$

parametrizes a dense subset of $\mathscr{H}_{\mathbf{h}}^{reg}$.

Definition 4.5. We denote by $\mathsf{H}_t(\mu)$ *a* maximal square submatrix of A in (16) such that $\det(\mathsf{H}_t(\overline{\mu}_{t-1})) \neq 0$.

The size of $\mathsf{H}_t(\mu)$ is the size of ν' in (16), that is the maximal number of independent columns in $H_t(\overline{\mu}_{t-1})$. Given an element $\mathcal{D} = \Lambda_1 \wedge \cdots \wedge \Lambda_r \in \mathscr{G}_r(\mathbb{C}[\mathbf{d}]_{r-1})$, in order to check that $\mathcal{D}$ is regular for B, it is sufficient to check first that $\Delta_B(\mathcal{D}) \neq 0$ and secondly that $|\mathsf{H}_t(\mu)| \neq 0$ for all $t = 0,\dots,\delta$, where $\mu = (\mu_{\beta,\alpha})$ is the ratio of Plücker coordinates of $\mathcal{D}$ defined by the formula (15).

5 NEWTON'S ITERATIONS

In this section we describe the extraction of a square, deflated system that allows for a Newton's method with quadratic convergence. We assume that the sole input is the equations $\mathbf{f} = (f_1,\dots,f_N) \in \mathbb{C}[\mathbf{x}]^N$, an approximate point $\xi \in \mathbb{C}^n$ and a tolerance $\varepsilon > 0$.

Using this input we first compute an approximate primal-dual pair $(B, \mathbf{\Lambda})$ by applying the iterative Algorithm 1. The rank and kernel vectors of the matrices K_t (see Algorithm 1) are computed numerically within tolerance ε, using SVD. Note that here and in Section 6 we do not need to certify the SVD computation but we are only using SVD to certify that some matrices are full rank by checking that the distance to the variety of singular matrices is bigger than the perturbation of the matrix. Thus we need a weaker test, which relies only on a lower bound of the smallest singular value.

The algorithm returns a basis $B = \{\mathbf{x}^{\beta_1}_{\xi},\dots,\mathbf{x}^{\beta_r}_{\xi}\}$ with exponent vectors $E = \{\beta_1,\dots,\beta_r\}$, as well as approximate values for the parameters $\mu = \{\mu_{\beta_i,\beta_j+\mathbf{e}_k} \ : \ |\beta_j| < |\beta_i| \in E,\ k = 1,\dots,n\}$. These parameters will be used as a starting point for Newton's iteration. Note that, by looking at B, we can also deduce the multiplicity r, the maximal order δ of dual differentials, the sequences $r_t = |B_t|$, and $h_t = |B_{[t]}|$ for $t = 0,\dots,\delta$.

Let F be the deflated system with variables $(\mathbf{x},\mu)$ defined by the relations (8) and Equations (9), (10) and (11) i.e.

$$F(\mathbf{x},\mu)=\begin{cases} \sum_{|\beta_s|<|\beta_j|<|\beta_i|} \mu_{\beta_i,\beta_j+\mathbf{e}_k}\mu_{\beta_j,\beta_s+\mathbf{e}_l} - \mu_{\beta_i,\beta_j+\mathbf{e}_l}\mu_{\beta_j,\beta_s+\mathbf{e}_k} = 0 & (a)\\ \quad \text{for all } i = 1,\dots,r,\ |\beta_s| < |\beta_i|,\ k \neq l \in \{1,\dots,n\} & \\ \mu_{\beta_i,\beta_j+\mathbf{e}_k} = \begin{cases} 1 & \text{for } \beta_i = \beta_j + \mathbf{e}_k \\ 0 & \text{for } \beta_j + \mathbf{e}_k \in E,\ \beta_i \neq \beta_j + \mathbf{e}_k, \end{cases} & (b)\\ \Lambda_i(f_j) = 0, \quad i = 1,\dots,r,\ j = 1,\dots,N. & (c) \end{cases}$$

Here $\Lambda_1 = 1_{\mathbf{x}}$ and $\Lambda_i = \sum_{|\beta_j|<|\beta_i|} \sum_{k=1}^n \mu_{\beta_i,\beta_j+\mathbf{e}_k} \overline{\int_k} \Lambda_j \in \mathbb{C}[\mu][\mathbf{d}_{\mathbf{x}}]$ denote dual elements with parametric coefficients defined recursively. Also, if $\Lambda_i = \sum_{|\alpha|\leq|\beta_i|} \mu_{\beta_i,\alpha} \frac{\mathbf{d}_{\mathbf{x}}^\alpha}{\alpha!}$ then

$$\Lambda_i(f_j) = \sum_{|\alpha|\leq|\beta_i|} \mu_{\beta_i,\alpha} \frac{\boldsymbol{\partial}^\alpha(f_j)(\mathbf{x})}{\alpha!}$$

which is in $\mathbb{C}[\mathbf{x},\mu]$ by Lemma 4.1. Note, however, that (a) and (b) are polynomials in $\mathbb{C}[\mu]$, only (c) depends on $\mathbf{x}$ and μ. Equations (b) define a simple substitution into some of the parameters μ. Hereafter, we explicitly substitute them and eliminate this part (b) from the equations we consider and reducing the parameter vector μ.

By Theorem 2.8, if B is a graded primal basis for $\mathbf{f}$ at the root ξ^* then the above overdetermined system has a simple root at a point (ξ^*,μ^*).

To extract a square subsystem defining the simple root (ξ^*,μ^*) in order to certify the convergence, we choose a maximal set of equations whose corresponding rows in the Jacobian are linearly independent. This is done by extracting first a maximal set of equations in (a) with linearly independent rows in the Jacobian. For that purpose, we use the rows associated to the maximal invertible matrix H_t (Definition 4.5) for each new basis element $\Lambda_i \in \mathcal{D}_{[t]}$ and $t = 1,\dots,r$. We denote by G_0 the subsystem of (a) that correspond to rows of H_t.

We complete the system of independent equations G_0 with equations from (c), using a QR decomposition and thresholding on the transposed Jacobian matrix of G_0 and (c) at the approximate root. Let us denote by F_0 the resulting square system, whose Jacobian, denoted by J_0, is invertible.

For the remaining equations F_1 of (c), not used to construct the square system F_0, define $\Omega = \{(i,j) : \Lambda_i(f_j) \in F_1\}$. We introduce new parameters $\epsilon_{i,j}$ for $(i,j) \in \Omega$ and we consider the perturbed system

$$f_{i,\epsilon} = f_i - \sum_{j|(i,j)\in\Omega} \epsilon_{i,j}\, \mathbf{x}^{\beta_j}_{\xi}.$$

The perturbed system is $\mathbf{f}_\epsilon = \mathbf{f} - \epsilon\, B$, where ϵ is the $N \times r$ matrix with $[\epsilon]_{i,j} = \epsilon_{i,j}$ if $(i,j) \in \Omega$ and $[\epsilon]_{i,j} = 0$ otherwise. Denote by $F(\mathbf{x},\mu,\epsilon)$ obtained from $F(\mathbf{x},\mu)$ by replacing $\Lambda_j(f_i)$ by $\Lambda_j(f_{i,\epsilon})$ for $j = 1,\dots,r, i = 1,\dots,N$. Then the equations used to construct the square Jacobian J_0 are unchanged. The remaining equations are of the form

$$\Lambda_j(f_{i,\epsilon}) = \Lambda_j(f_i) - \epsilon_{i,j} = 0 \quad (i,j) \in \Omega.$$

Therefore the Jacobian of the complete system $F(\mathbf{x},\mu,\epsilon)$ is a square invertible matrix of the form

$$J_\epsilon := \begin{pmatrix} J_0 & 0 \\ J_1 & \mathrm{Id} \end{pmatrix}$$

where J_1 is the Jacobian of the system F_1 of polynomials $\Lambda_j(f_i) \in \mathbb{C}[\mathbf{x},\mu]$ with $(i,j) \in \Omega$.

Since J_ϵ is invertible, the square extended system $F(\mathbf{x},\mu,\epsilon)$ has an isolated root (ξ^*,μ^*,ϵ^*) corresponding to the isolated root (ξ^*,μ^*) of the square system F_0. Furthermore, $\Lambda^*_j(f_i) = \epsilon^*_{i,j} = 0$ for $(i,j) \in \Omega$. Here $\Lambda^*_1,\dots,\Lambda^*_r \in \mathbb{C}[\mathbf{d}_{\xi^*}]$ are defined from (ξ^*,μ^*) recursively by

$$\Lambda^*_1 = 1_{\xi^*} \text{ and } \Lambda^*_i = \sum_{|\beta_j|<|\beta_i|} \sum_{k=1}^n \mu^*_{\beta_i,\beta_j+\mathbf{e}_k} \overline{\int_k} \Lambda^*_j. \tag{17}$$

We have the following property:

Theorem 5.1. *If the Newton iteration*

$$(\xi_{k+1},\mu_{k+1}) = (\xi_k,\mu_k) - J_0(\xi_k,\mu_k)^{-1} F_0(\xi_k,\mu_k),$$

starting from a point (ξ_0,μ_0) converges when $k \to \infty$, to a point (ξ^*,μ^*) such that B is a regular basis for the inverse system $\mathcal{D}^*$ associated to (ξ^*,μ^*) and $\mathcal{D}^*$ is complete for $\mathbf{f}$, then there exists a perturbed system $f_{i,\epsilon^*} = f_i - \sum_{j|(i,j)\in\Omega} \epsilon^*_{i,j}\, \mathbf{x}^{\beta_j}_{\xi^*}$ with $\epsilon^*_{i,j} = \Lambda^*_j(f_i)$ such that ξ^* is a multiple root of f_{i,ϵ^*} with the multiplicity structure defined by μ^*.

6 CERTIFICATION

In this section we describe how to certify that the Newton iteration defined in Section 5 quadratically converges to a point that defines an exact root with an exact multiplicity structure of a perturbation of the input polynomial system $\mathbf{f}$. More precisely, we are given $\mathbf{f} = (f_1,\dots,f_N) \in \mathbb{C}[\mathbf{x}]^N$, $B = \{\mathbf{x}^{\beta_1},\dots,\mathbf{x}^{\beta_r}\} \subset \mathbb{C}[\mathbf{x}]$ in increasing order of degrees and closed under division, $\delta := |\beta_r|$. We are also given the deflated systems $F(\mathbf{x},\mu)$, its square subsystem $F_0(\mathbf{x},\mu)$ defined in Section 5 and $F_1(\mathbf{x},\mu)$ the remaining equations in $F(\mathbf{x},\mu)$. Finally, we are given $\xi_0 \in \mathbb{C}^n$ and $\mu_0 = \{\mu^{(0)}_{\beta_i,\beta_j+\mathbf{e}_k} \in \mathbb{C} \ : \ i,j = 1,\dots,r, |\beta_j| < |\beta_i|, k = 1,\dots,n\}$. Our certification will consist of a symbolic and a numeric part:

Regularity certification. We certify that B is regular (see Definition 4.3). This part of the certification is purely symbolic and inductive on t. Suppose for some $t-1 < \delta$ we certified that B_{t-1} is regular and computed the parameters $\overline{\mu}_{t-1}$ and the parametrization

$$\Phi_{t-1} : \overline{\mu}_{t-1} \mapsto \left(q_{\beta_i,\alpha}(\overline{\mu}_{t-1})\right)_{|\beta_i|\leq t-1,|\alpha|\leq t-1}$$

(Algorithm 2). Then to prove that B_t is regular, we consider the coefficient matrix H_t of equations (12) and (13). We substitute the parametrization Φ_{t-1} to get the matrices $H_t(\overline{\mu}_{t-1})$. We symbolically prove that the rows of $\mathsf{H}_t(\overline{\mu}_{t-1})$ (Definition 4.5) are linearly independent and span all

rows of $H_t(\overline{\mu}_{t-1})$ over $\mathbb{Q}(\overline{\mu}_{t-1})$. If that is certified, we compute the parameters $\overline{\mu}_t$ and the parametrization $\Phi_t : \overline{\mu}_t \mapsto \left(q_{\beta_i,\alpha}(\overline{\mu}_t)\right)_{|\beta_i|\le t, |\alpha|\le t}$ as in Algorithm 2 inverting the square submatrix H_t of H_t such that the denominators of $q_{\beta_i,\alpha}$ for $|\beta_i| = t$ divide $\det(\mathsf{H}_t(\overline{\mu}_{t-1})) \neq 0$.

Singularity certification.

(C1) We certify that the Newton iteration for the square system F_0 starting from (ξ_0, μ_0) quadratically converges to some root (ξ^*, μ^*) of F_0, such that $\|(\xi_0, \mu_0) - (\xi^*, \mu^*)\|_2 \le \tilde{\beta}$, using α-theory.

(C2) We certify that $\mathcal{D}^* = \mathrm{span}(\mathbf{\Lambda}^*)$ is regular for B (see Definition 4.3), by checking that $|\mathsf{H}_t(\mu^*)| \neq 0$ for $t = 1, \ldots, \delta$ (See Definition 4.5), using the Singular Value Decomposition of $\mathsf{H}_t(\mu_0)$ and the distance bound $\tilde{\beta}$ between μ^* and μ_0.

(C3) We certify that $\mathbf{\Lambda}^*$ is complete for $\mathbf{f}$ at ξ^* (see Definition 2.7), where $\mathbf{\Lambda}^* \subset \mathbb{C}[\mathbf{d}_{\xi^*}]$ is the dual systems defined from (ξ^*, μ^*) recursively as in (17). This is done by checking that $\ker K_{\delta+1}(\xi^*, \mu^*) = \{0\}$ (See Definition 2.7), using the Singular Value Decomposition of $K_{\delta+1}(\xi_0, \mu_0)$ and the distance bound $\tilde{\beta}$ between (ξ^*, μ^*) and (ξ_0, μ_0).

Let us now consider for a point-multplicity structure pair (ξ_0, μ_0) $\tilde{\gamma} := \sup_{k\ge 2} \|DF_0^{-1}(\xi_0,\mu_0)\frac{D^k F_0(\xi_0,\mu_0)}{k!}\|^{\frac{1}{k-1}}$, $\tilde{\beta} := 2\|DF_0^{-1}(\xi_0,\mu_0)\, F_0(\xi_0,\mu_0)\|$, $\tilde{\alpha} := \tilde{\beta}\,\tilde{\gamma}$ and for a matrix function $A(\xi,\mu)$, let $\mathcal{L}_1(A; \xi_0, \mu_0; b)$ be a bound on its Lipschitz constant in the ball $\mathcal{B}_b(\xi_0, \mu_0)$ of radius b around (ξ_0, μ_0) such that $\|A(\xi,\mu) - A(\xi_0,\mu_0)\| \le \mathcal{L}_1(A; \xi_0, \mu_0; b)\, \|(\xi,\mu) - (\xi_0,\mu_0)\|$ for $(\xi,\mu) \in \mathcal{B}_b(\xi_0,\mu_0)$. For a matrix M, let $\sigma_{min}(M)$ be its smallest singular value. We have the following result:

Theorem 6.1. *Let $B = \{\mathbf{x}^{\beta_1}, \ldots, \mathbf{x}^{\beta_r}\} \subset \mathbb{C}[\mathbf{x}]$ be closed under division and suppose B is regular. Suppose that $\tilde{\alpha} < \tilde{\alpha}_0 := 0.26141$, $\mathcal{L}_1(K_{\delta+1}; \xi_0, \mu_0; \tilde{\beta})\,\tilde{\beta} < \sigma_{\min}(K_{\delta+1}(\xi_0,\mu_0))$ and for $t = 1, \ldots, \delta$ it holds that $\mathcal{L}_1(\mathsf{H}_t; \mu_0; \tilde{\beta})\,\tilde{\beta} < \sigma_{min}(\mathsf{H}_t(\mu_0))$. Then the Newton iteration on the square system F_0 starting from (ξ_0, μ_0) converges quadratically to a point (ξ^*, μ^*) corresponding to a multiple point ξ^* with multiplicity structure μ^* of the perturbed system $\mathbf{f}_{\epsilon^*} = \mathbf{f} - \epsilon^* B_{\xi^*}$ such that $\|\epsilon^*\| \le \|F_1(\xi_0,\mu_0)\| + \mathcal{L}_1(F_1; \xi_0, \mu_0; \tilde{\beta})\,\tilde{\beta}$, where $B_{\xi^*} = \{\mathbf{x}^{\beta_1}_{\xi^*}, \ldots, \mathbf{x}^{\beta_r}_{\xi^*}\}$.*

7 EXPERIMENTATION

In this section we work out some examples with (approximate) singularities. The experiments are carried out using Maple, and our code is publicly available at https://github.com/filiatra/polyonimo.

Example 7.1. We consider the equations

$f_1 = x_1^3 + x_2^2 + x_3^2 - 1$, $f_2 = x_2^3 + x_1^2 + x_3^2 - 1$, $f_3 = x_3^3 + x_1^2 + x_2^2 - 1$,

the approximate root $\boldsymbol{\xi}_0 = (0.002, 1.003, 0.004)$ and threshold $\varepsilon = 0.01$. In the following we use 32-digit arithmetic for all computations.

We shall first compute a primal basis using Algorithm 1. In the first iteration we compute the 3×3 matrix $K_1 = K_1(\boldsymbol{\xi}_0)$. The elements in the kernel of this matrix consists of elements of the form $\Lambda = \nu_1^1 d_1 + \nu_1^2 d_2 + \nu_1^3 d_3$. The singular values of $K_1(\boldsymbol{\xi}_0)$ are $(4.1421, 0.0064, 0.0012)$, which implies a two-dimensional kernel, since two of them are below threshold ε. The (normalized) elements in the kernel are $\tilde{\Lambda}_2 = d_1 - 0.00117 d_2$ and $\tilde{\Lambda}_3 = d_3 - 0.00235 d_2$. Note that d_2 was not chosen as a leading term. This is due to pivoting used in the numeric process, in order to avoid leading terms with coefficients below the tolerance ε. The resulting primal basis $B_1 = \{1, x_1, x_3\}$ turns out to be closed under derivation.

Similarly, in degree 2 we compute one element $\tilde{\Lambda}_4 = d_1 d_3 - 0.00002 d_1^2 - 0.00235 d_1 d_2 + 5.5\cdot 10^{-6} d_2^2 - 0.00117\cdot d_2 d_3 - 0.00002 d_3^2 + 5.9\cdot 10^{-6} d_2$.

In the next step, we have $\ker K_3 = \{0\}$, since the minimum singular value is $\sigma_{\min} = 0.21549$, therefore we stop the process, since the computed dual is approximately complete (cf. Definition 2.7). We derive that the approximate multiple point has multiplicity $r = 4$ and one primal basis is $B = \{1, x_1, x_3, x_1x_3\}$.

The parametric form of a basis of $\mathcal{D}_1$ is $\ker K_1 = \langle \Lambda_2 = d_1 + \mu_{2,1} d_2, \Lambda_3 = d_3 + \mu_{3,1} d_2\rangle$. Here we incorporated (10), thus fixing some of the parameters according to primal monomials x_1 and x_3.

The parametric form of the matrix $K_2(\boldsymbol{\xi}, \boldsymbol{\mu})$ of the integration method at degree 2 is

$$\begin{array}{c} \\ (9)\\ (9)\\ (9)\\ \Lambda(f_1)\\ \Lambda(f_2)\\ \Lambda(f_3) \end{array}\!\!\begin{array}{c} \begin{array}{ccccccccc} \nu_1^1 & \nu_1^2 & \nu_1^3 & \nu_2^1 & \nu_2^2 & \nu_2^3 & \nu_3^1 & \nu_3^2 & \nu_3^3 \end{array}\\ \begin{bmatrix} 0 & 0 & 0 & 0 & \mathbf{0} & -\mu_{2,1} & \mathbf{0} & \mathbf{1} & -\mu_{3,1} \\ 0 & 0 & 0 & 0 & \mathbf{0} & -1 & \mathbf{1} & \mathbf{0} & 0 \\ 0 & 0 & 0 & \mu_{2,1} & \mathbf{-1} & 0 & \boldsymbol{\mu_{3,1}} & \mathbf{0} & 0 \\ 3\xi_1^2 & 2\xi_2 & 2\xi_3 & 3\xi_1 & \mu_{2,1} & 0 & 3\xi_1 & \mu_{3,1} & 1 \\ 2\xi_1 & 3\xi_2^2 & 2\xi_3 & 1 & 3\mu_{2,1}\,\xi_2 & 0 & 0 & 3\mu_{3,1}\,\xi_2 & 1 \\ 2\xi_1 & 2\xi_2 & 3\xi_3^2 & 1 & \mu_{2,1} & 0 & 0 & \mu_{3,1} & 3\xi_3 \end{bmatrix} \end{array},$$

where the columns correspond to the parameters in the expansion (5):

$$\Lambda_4 = \nu_1^1 d_1 + \nu_1^2 d_2 + \nu_1^3 d_3 + \nu_2^1 d_1^2 + \nu_2^2 (d_1 d_2 + \mu_{2,1} d_2^2)$$
$$+\nu_2^3 (d_1 d_3 + \mu_{2,1} d_3 d_2) + \nu_3^1 (\mu_{3,1} d_1 d_2) + \nu_3^2 (\mu_{3,1} d_2^2) + \nu_3^3 (d_3^2 + \mu_{3,1} d_2 d_3)$$

Setting $\Lambda_4(x_1x_3) = 1$ and $\Lambda_4(x_1) = \Lambda_4(x_3) = \Lambda_4(1) = 0$, we obtain $\nu_1^1 = \nu_1^3 = 0$ and $\nu_2^3 = 1$. The dual element of order 2 has the parametric form

$$\Lambda_4 = d_1 d_3 + \mu_{4,1}\, d_2 + \mu_{4,2}\, d_1^2 + \mu_{4,3}\, d_1 d_2 + \mu_{4,6}\, d_3^2 + \qquad (18)$$
$$+ (\mu_{2,1} + \mu_{3,1}\mu_{4,6})\, d_2 d_3 + (\mu_{2,1}\mu_{4,4} + \mu_{3,1}\mu_{4,5})\, d_2^2$$

($\nu_1^2 = \mu_{4,1}, \nu_2^1 = \mu_{4,2}, \nu_2^2 = \mu_{4,3}, \nu_3^1 = \mu_{4,4}, \nu_3^2 = \mu_{4,5}, \nu_3^3 = \mu_{4,6}$). Overall 8 parameters are used in the representation of $\mathcal{D}_2$.

The highlighted entries of $K_2(\boldsymbol{\xi}, \boldsymbol{\mu})$ form the non-singular matrix H_2 in Definition 4.5, therefore $\mathcal{D}_2$ is regular for B (cf. Definition 4.3). We obtain the polynomial parameterization $\mu_{4,3} = \mu_{2,1}\mu_{4,2} + \mu_{3,1}$, $\mu_{4,4} = 1$, $\mu_{4,5} = \mu_{2,1} + \mu_{3,1}\mu_{4,6}$ with the free parameters $\bar{\mu} = (\mu_{2,1}, \mu_{3,1}, \mu_{4,1}, \mu_{4,2}, \mu_{4,6})$. There is no denominator since $\det \mathsf{H}_2 = 1$.

We now setup the numerical scheme. The overdetermined and deflated system $F(\mathbf{x}, \boldsymbol{\mu})$ consists of 15 equations:

$\mu_{2,1}\mu_{4,2} + \mu_{3,1} - \mu_{4,3}$, $-\mu_{4,4} + 1$, $-\mu_{2,1}\mu_{4,4} - \mu_{3,1}\mu_{4,6} + \mu_{4,5}$,
$\Lambda_1(f_1) = f_1, \Lambda_1(f_2) = f_2, \Lambda_1(f_3) = f_3, \Lambda_2(f_1) = 2\mu_{2,1}x_2 + 3x_1^2$,
$\Lambda_2(f_2) = 3\mu_{2,1}x_2^2 + 2x_1, \Lambda_2(f_3) = 2\mu_{2,1}x_2 + 2x_1, \Lambda_3(f_1) = 2\mu_{3,1}x_2 + 2x_3$,
$\Lambda_3(f_2) = 3\mu_{3,1}x_2^2 + 2x_3$, $\Lambda_3(f_3) = 2\mu_{3,1}x_2 + 3x_3^2$,
$\Lambda_4(f_1) = \mu_{2,1}\mu_{4,3} + \mu_{3,1}\mu_{4,5} + 2\mu_{4,1}x_2 + 3\mu_{4,2}x_1 + \mu_{4,6}$,
$\Lambda_4(f_2) = 3\mu_{2,1}\mu_{4,3}x_2 + 3\mu_{3,1}\mu_{4,5}x_2 + 3\mu_{4,1}x_2^2 + \mu_{4,2} + \mu_{4,6}$,
$\Lambda_4(f_3) = \mu_{2,1}\mu_{4,3} + \mu_{3,1}\mu_{4,5} + 2\mu_{4,1}x_2 + 3\mu_{4,6}x_3 + \mu_{4,2}$

We now consider $J_F(\boldsymbol{\xi}_0, \boldsymbol{\mu}_0)$. This Jacobian is of full rank, and we can obtain a maximal minor by removing $\Lambda_1(f_2), \Lambda_1(f_3), \Lambda_2(f_3)$ and $\Lambda_3(f_3)$ from F. We obtain the square 11×11 system denoted by F_0.

The initial point of the Newton iterations is $\boldsymbol{\xi}_0 = (0.002, 1.003, 0.004)$ and the approximation of the variables $\mu_{i,j}$ provided by the numerical integration method: $\boldsymbol{\mu}_0 = (-0.00117, -0.00235, 5.9\cdot 10^{-6}, -0.00002, -0.00235, 1.0, -0.00117, -0.00002)$.

We now use Theorem 6.1 to certify the convergence to a singular system. We can compute for $(\boldsymbol{\xi}_0, \boldsymbol{\mu}_0)$ the value $\tilde{\beta} \approx 0.01302$. Moreover, $\sigma_{\min}(K_{\delta+1}(\boldsymbol{\xi}_0, \boldsymbol{\mu}_0)) = 0.21549$ and the minimum singular value of the highlighted submatrix of $K_2(\boldsymbol{\xi}_0, \boldsymbol{\mu}_0)$ is equal to one. Therefore $\tilde{\beta}$ is at least one order of magnitude less than both of them, which is sufficient, since the involved Lipschitz and $\tilde{\gamma}$ constants are of the order of 1 for the input polynomials. In the first iteration we obtain $\tilde{\beta} \approx 0.00011$ which clearly indicates that we are in the region of convergence. Indeed, the successive residuals for 4 iterations are $0.00603, 4.0\cdot 10^{-5}, 2.07\cdot 10^{-9}, 8.6\cdot 10^{-18}, 3.55\cdot 10^{-35}$. Clearly, the residual shrinks with a quadratic rate[1]. We obtain $\boldsymbol{\xi}_4 = (1.8\cdot 10^{-37}, 1.0, 2.8\cdot 10^{-36})$ and the overdetermined system is satisfied by this point: $\|F(\boldsymbol{\xi}_4, \boldsymbol{\mu}_4)\|_\infty = 8\cdot 10^{-35}$; the resulting dual structure is $\mathcal{D}_2^* = \{1, d_1, d_3, d_1 d_3\}$.

Example 7.2. We demonstrate how our method handles inaccuracies in the input, and recovers a nearby system with a true multiple point. Let

$$f_1 = {x_1}^2 + x_1 - x_2 + 0.003 \quad , \quad f_2 = {x_2}^2 + 1.004 x_1 - x_2.$$

[1]The convergence is seen up to machine error. If we increase the accuracy to 150 digits the rate remains quadratic for 7 iterations: $\ldots 3.55\cdot 10^{-35}, 6.78\cdot 10^{-70}, 4.15\cdot 10^{-140}, 5.1\cdot 10^{-281}$.

There is a cluster of three roots around $\xi_0 = (0.001, -0.002)$. Our goal is to squeeze the cluster down to a three-fold real root. We use 32 digits for the computation. Starting with ξ_0, and a tolerance equal to 10^{-2} Algorithm 1 produces an approximate dual $\mathbf{1}$, $d_1 + 1.00099651 d_2$, $d_1^2 + 1.00099651 d_1 d_2 + 1.00266222 d_2^2 + 0.99933134 d_2$ and identifies the primal basis $B = \{1, x_1, x_1^2\}$ using pivoting on the integration matrix. The sole stability condition reads $\mu_{1,1} - \mu_{2,2} = 0$, and $\Lambda_1 = \mathbf{1}$, $\Lambda_2 = d_1 + \mu_{1,1} d_2$, $\Lambda_3 = d_1^2 + \mu_{1,1} d_1 d_2 + \mu_{2,1} d_2 + \mu_{2,2} \mu_{1,1} d_2^2$.

The nearby system that we shall obtain is deduced by the residue in Newton's method. In particular, starting from ξ_0, we consider the square system given by removing the equations $\Lambda_1(f_1) = 0$ and $\Lambda_2(f_2) = 0$. The rank of the corresponding Jacobian matrix remains maximal, therefore such a choice is valid. Newton's iterations converge quadratically to the point $(\xi_5, \mu_5) = (1.1 \cdot 10^{-33}, 1.2 \cdot 10^{-33}, 1, 1, 1)$. The full residual is now

$$F(\xi_5, \mu_5) = (0, 0.003, -10^{-32}, 10^{-32}, 0.004, 0, 0)\,.$$

This yields a perturbation $\tilde{f}_1 \approx f_1 - 0.003$ and $\tilde{f}_2 \approx f_2 - 0.004(x_1 - \xi_1^*)$ to obtain a system with an exact multiple root at the origin (cf. Th. 6.1). Of course, this choice of the square sub-system is not unique. By selecting to remove equations $\Lambda_1(f_1) = 0$ and $\Lambda_1(f_2) = 0$ instead, we obtain $(\xi_5, \mu_5) = (0.00066578, -0.00133245, 1.001, 1.0, 1.001)$ and the residual $F(\xi_5, \mu_5) = (0, 0.005, 0.002, 0, 0, 0, 0)$, so that the nearby system

$$f_1^* \approx {x_1}^2 + x_1 - x_2 + 0.008, \quad f_2^* \approx {x_2}^2 + 1.004 x_1 - x_2 + 0.002$$

has a singularity at the limit point $\xi^* \approx (0.00066578, -0.00133245)$ described locally by the coefficients $\mu^* \approx (1.001, 1.0, 1.001)$.

Finally, consider the two square sub-systems as above, after changing f_1, f_2 to define an *exact three-fold root* at the origin (i.e. $f_1 = {x_1}^2 + x_1 - x_2$, $f_2 = {x_2}^2 + x_1 - x_2$). Newton's iteration with initial point ξ_0 on either deflated system converges quadratically to $(\xi, \mu) = (\mathbf{0}, \mathbf{1})$. This is a general property of the method: exact multiple roots and their structure are recovered by this process if ξ_0 is a sufficiently good initial approximation (cf. Section 5). We plan to develop this aspect further in the future.

Example 7.3. We show some execution details on a set of benchmark examples in taken from [7], see also [26]. For this benchmark, we are given systems and points with multiplicities. We perturb the given points with a numerical perturbation of order 10^{-2}. We use double precision arithmetic and setup Newton's iteration; with less than 10 iterations, the root was approximated within the chosen accuracy.

In Table 1, "IM" is the maximal size of the (numeric) integration matrix that is computed to obtain the multiplicity, "$\#\mu$" is the number of new parameters that are needed for certified deflation, "SC" is the number of stability constraints that were computed and "OS" stands for the size of the overdetermined system (equations × variables). This is the size of the Jacobian matrix that must be computed and inverted in each Newton's iteration. We can observe that the number of parameters required can grow significantly. Moreover, these parameters induce non-trivial denominators in the rational functions $q_{\beta_j,\alpha}(\overline{\mu})$ of Prop. 4.4. for the instances cmbs1, cmbs2 and KSS.

System	r/n	IM	SC	$\#\mu$	OS
cmbs1	11/3	27×23	75	74	108×77
cmbs2	8/3	21×17	21	33	45×36
mth191	4/3	10×9	3	9	15×12
decker2	4/2	5×5	4	8	12×10
Ojika2	2/3	6×5	0	2	6×5
Ojika3	4/3	12×9	15	14	27×17
KSS	16/5	155×65	510	362	590×367
Capr.	4/4	22×13	6	15	22×19
Cyclic-9	4/9	104×33	36	40	72×49

Table 1: Size of required matrices and parameters for deflation.

Acknowledgments. This research was partly supported by the H2020-MSCA-ITN projects GRAPES (GA 860843) and POEMA (GA 813211) and the NSF grant CCF-1813340.

REFERENCES

[1] Ayyildiz Akoglu, T., Hauenstein, J. D., and Szanto, A. Certifying solutions to overdetermined and singular polynomial systems over Q. *Journal of Symbolic Computation 84* (2018), 147–171.

[2] Bejleri, D., and Stapleton, D. The tangent space of the punctual Hilbert scheme. *The Michigan Mathematical Journal 66*, 3 (Aug. 2017), 595–610.

[3] Blum, L., Cucker, F., Shub, M., and Smale, S. *Complexity and Real Computation.* Springer, NY, 1998.

[4] Briançon, J. Description de $Hilb^n C\{x, y\}$. *Inventiones mathematicae 41* (1977), 45–90.

[5] Briançon, J., and Iarrobino, A. Dimension of the punctual Hilbert scheme. *Journal of Algebra 55*, 2 (Dec. 1978), 536–544.

[6] Dayton, B. H., Li, T.-Y., and Zeng, Z. Multiple zeros of nonlinear systems. *Math. Comput. 80*, 276 (2011), 2143–2168.

[7] Dayton, B. H., and Zeng, Z. Computing the multiplicity structure in solving polynomial systems. In *Proc. of ISSAC '05* (NY, USA, 2005), ACM, pp. 116–123.

[8] Dedieu, J.-P., and Shub, M. On simple double zeros and badly conditioned zeros of analytic functions of n variables. *Math. Comp. 70*, 233 (2001), 319–327.

[9] Doubilet, P., Rota, G.-C., and Stein, J. On the Foundations of Combinatorial Theory. *Studies in Applied Mathematics 53*, 3 (1974), 185–216.

[10] Emiris, I. Z., Mourrain, B., and Tsigaridas, E. The DMM bound: multivariate (aggregate) separation bounds. In *Proceedings of the ISSAC'10* (Munich, Germany, July 2010), S. Watt, Ed., ACM, pp. 243–250.

[11] Giusti, M., Lecerf, G., Salvy, B., and Yakoubsohn, J.-C. On location and approximation of clusters of zeros of analytic functions. *Found. Comput. Math. 5*, 3 (2005), 257–311.

[12] Giusti, M., Lecerf, G., Salvy, B., and Yakoubsohn, J.-C. On location and approximation of clusters of zeros: Case of embedding dimension one. *Foundations of Computational Mathematics 7* (2007), 1–58.

[13] Giusti, M., and Yakoubsohn, J.-C. Multiplicity hunting and approximating multiple roots of polynomial systems. vol. 604 of *Contemp. Math.*, AMS, pp. 105–128.

[14] Giusti, M., and Yakoubsohn, J.-C. Approximation numérique de racines isolées multiples de systèmes analytiques, 2018. arXiv:1809.05446.

[15] Hao, W., Sommese, A. J., and Zeng, Z. Algorithm 931: an algorithm and software for computing multiplicity structures at zeros of nonlinear systems. *ACM Trans. Math. Software 40*, 1 (2013), Art. 5, 16.

[16] Hauenstein, J. D., Mourrain, B., and Szanto, A. On deflation and multiplicity structure. *Journal of Symbolic Computation 83* (2016), 228–253.

[17] Iarrobino, A. A. *Punctual Hilbert Schemes*, vol. 188 of *Memoirs of the American Mathematical Society*. AMS, Providence, 1977.

[18] Kanzawa, Y., and Oishi, S. Approximate singular solutions of nonlinear equations and a numerical method of proving their existence. No. 990. 1997, pp. 216–223.

[19] Lee, K., Li, N., and Zhi, L. On isolation of singular zeros of multivariate analytic systems, 2019. arXiv:1904.0793.

[20] Leykin, A., Verschelde, J., and Zhao, A. Newton's method with deflation for isolated singularities of polynomial systems. *Theor. Computer Science 359*, 1-3 (2006), 111 – 122.

[21] Leykin, A., Verschelde, J., and Zhao, A. Higher-order deflation for polynomial systems with isolated singular solutions. In *Algorithms in Algebraic Geometry*, A. Dickenstein, F.-O. Schreyer, and A. Sommese, Eds., vol. 146 of *The IMA Volumes in Mathematics and its Applications*. Springer, 2008, pp. 79–97.

[22] Li, N., and Zhi, L. Verified error bounds for isolated singular solutions of polynomial systems: case of breadth one. *Theoret. Comput. Sci. 479* (2013), 163–173.

[23] Li, N., and Zhi, L. Verified error bounds for isolated singular solutions of polynomial systems. *SIAM J. Numer. Anal. 52*, 4 (2014), 1623–1640.

[24] Li, Z., and Sang, H. Verified error bounds for singular solutions of nonlinear systems. *Numer. Algorithms 70*, 2 (2015), 309–331.

[25] Mantzaflaris, A., and Mourrain, B. Deflation and certified isolation of singular zeros of polynomial systems. In *Proc. of ISSAC '11* (2011), ACM, pp. 249–256.

[26] Mantzaflaris, A., and Mourrain, B. Singular zeros of polynomial systems. In *Advances in Shapes, Geometry, and Algebra*, T. Dokken and G. Muntingh, Eds., vol. 10 of *Geometry and Computing*. Springer, 2014, pp. 77–103.

[27] Mourrain, B. Isolated points, duality and residues. *Journal of Pure and Applied Algebra 117-118* (1997), 469 – 493.

[28] Rump, S., and Graillat, S. Verified error bounds for multiple roots of systems of nonlinear equations. *Numerical Algorithms 54* (2010), 359–377.

[29] Shafarevich, I. R. *Basic Algebraic Geometry 1: Varieties in Projective Space*, 3rd edition ed. Springer, New York, 2013.

[30] Wu, X., and Zhi, L. Computing the multiplicity structure from geometric involutive form. In *Proceedings of ISSAC'08* (NY, USA, 2008), ACM, pp. 325–332.

[31] Wu, X., and Zhi, L. Determining singular solutions of polynomial systems via symbolic-numeric reduction to geometric involutive form. *J. Symb. Comput. 27* (2008), 104–122.

[32] Yakoubsohn, J.-C. Finding a cluster of zeros of univariate polynomials. vol. 16. 2000, pp. 603–638. Complexity theory, real machines, and homotopy (Oxford, 1999).

[33] Yakoubsohn, J.-C. Simultaneous computation of all the zero-clusters of a univariate polynomial. In *Foundations of computational mathematics (Hong Kong, 2000)*. World Sci. Publ., River Edge, NJ, 2002, pp. 433–455.

[34] Zeng, Z. The closedness subspace method for computing the multiplicity structure of a polynomial system. vol. 496 of *Contemp. Math.* Amer. Math. Soc., Providence, RI, 2009, pp. 347–362.

Fast Multipoint Evaluation and Interpolation of Polynomials in the LCH-basis over $\mathbb{F}_{p^r}$

Axel Mathieu-Mahias
Université Paris-Saclay, UVSQ, CNRS, Laboratoire de mathématiques de Versailles
78000, Versailles, France
axel.mathieu-mahias@uvsq.fr

Michaël Quisquater
Université Paris-Saclay, UVSQ, CNRS, Laboratoire de mathématiques de Versailles
78000, Versailles, France
michael.quisquater@uvsq.fr

ABSTRACT

Lin, Chung and Han introduced in 2014 the LCH-basis in order to derive FFT-based multipoint evaluation and interpolation algorithms with respect to this polynomial basis. Considering an affine space of $n = 2^j$ points, their algorithms require $O(n \cdot \log_2 n)$ operations in $\mathbb{F}_{2^r}$. The LCH-basis has then been extended over finite fields of characteristic p by Lin et al. in 2016 and an n-point evaluation algorithm has been derived for $n = p^j$ with complexity $O(n \cdot \log_p n \cdot p)$. However, the problem of interpolating polynomials represented in such a basis over $\mathbb{F}_{p^r}$ has not been addressed.

In this paper, we fill this gap and also derive a faster algorithm for evaluating polynomials in the LCH-basis at multiple points over $\mathbb{F}_{p^r}$. We follow a different approach where we represent the multipoint evaluation and interpolation maps by well-defined matrices. We present factorizations of such matrices into the product of sparse matrices which can be evaluated efficiently. These factorizations lead to fast algorithms for both the multipoint evaluation and the interpolation of polynomials represented in the LCH-basis at $n = p^j$ points with optimized complexity $O(n \cdot \log_2 n \cdot \log_2 p \cdot \log_2 \log_2 p)$.

A particular attention is paid to provide in-place algorithms with high memory-locality. Our implementations written in C confirm that our approach improves the original transforms.

CCS CONCEPTS

• **Mathematics of computing → Computation of transforms**; **Computations in finite fields**.

KEYWORDS

Finite fields, multipoint evaluation, interpolation, LCH-basis

ACM Reference Format:
Axel Mathieu-Mahias and Michaël Quisquater. 2020. Fast Multipoint Evaluation and Interpolation of Polynomials in the LCH-basis over $\mathbb{F}_{p^r}$. In *International Symposium on Symbolic and Algebraic Computation (ISSAC '20), July 20–23, 2020, Kalamata, Greece.* ACM, New York, NY, USA, 8 pages. https://doi.org/10.1145/3373207.3404009

ISSAC, 2020,

ACM ISBN 978-1-4503-7100-1/20/07…$15.00
https://doi.org/10.1145/3373207.3404009

1 INTRODUCTION

Let $\mathbb{F}_{p^r}$ be the finite field with p^r elements and let us denote the set of polynomials with respect to the monomial basis $\{1, x, x^2, \ldots\}$ by $\mathbb{F}_{p^r}[x]$. Let $\mathsf{M}(n)$ denote the number of field operations required to multiply two polynomials in $\mathbb{F}_{p^r}[x]$ of degree less than n, which may be taken to be in $O(n \log_2 n \log_2 \log_2 n)$. The standard multipoint evaluation and interpolation problems at n points of such polynomials over $\mathbb{F}_{p^r}$ can be solved with $O(\mathsf{M}(n) \log_2 n)$ operations in $\mathbb{F}_{p^r}$ [5].

The multipoint evaluation and interpolation can be performed with even lower algebraic complexities when applied to particular sets of evaluation points. The so-called Fast Fourier transform (FFT) [3] based algorithms offer complexities as low as $O(n \log_2 n)$ [11–14].

Related work. The LCH-basis presented in [13] is based on subspace polynomials over $\mathbb{F}_{2^r}$. It has been shown that the resulting FFT-based multipoint evaluation and interpolation on a affine space of $n = 2^j$ points of polynomials that are represented in that basis can be done in $O(n \cdot \log_2 n)$ operations over $\mathbb{F}_{2^r}$. The transforms have been mostly employed in coding theory and in particular applied to Reed-Solomon codes [9–13]. Basis conversion algorithms are also proposed in [12]. Furthermore, fast polynomial arithmetic in the LCH-basis are presented in [11]. In particular, a fast polynomial division and a fast half-GCD algorithm are derived. Besides, fast multiplications for long binary polynomials can be found in [2].

Regarding finite fields of characteristic p, [12] extends the LCH-basis over $\mathbb{F}_{p^r}$. However, only a solution for the forward transform is presented. It leads to a multipoint evaluation of polynomials in the LCH-basis of degree less than $n = p^j$ in $O(n \cdot \log_p n \cdot p)$ operations of $\mathbb{F}_{p^r}$. To the best of our knowledge, no solution has been provided to solving the interpolation problem of such polynomials over $\mathbb{F}_{p^r}$.

Contribution. This paper presents another approach than the original solution presented in [11] for solving the multipoint evaluation and interpolation problems. Namely, we express the map of the multipoint evaluation of polynomials in the LCH-basis by a matrix.

We then present a factorization of this matrix into the product of sparse matrices and show how to perform the global computation efficiently. Our approach relies on standard fast polynomial arithmetic over $\mathbb{F}_{p^r}$. It leads to a solution for multipoint evaluation at $n = p^j$ points of polynomials that are represented in the LCH-basis over $\mathbb{F}_{p^r}$ optimized from $O(n \cdot \log_p n \cdot p)$ to $O(n \cdot \log_2 n \cdot \log_2 p \cdot \log_2 \log_2 p)$.

We deduce the invertibility of the matrix corresponding to the evaluation map from its factorization. As the inverse matrix corresponds to the interpolation map, we therefore also provide a solution for fast interpolation of polynomials in the LCH-basis over $\mathbb{F}_{p^r}$ in $O(n\cdot\log_2 n\cdot\log_2 p\cdot\log_2\log_2 p)$. To the best of our knowledge, no solution was provided until now for the interpolation problem of such polynomials over $\mathbb{F}_{p^r}$. Moreover our approach leads to in-place algorithms with high memory-locality.

Outline. The paper is organized as follows. Section 2 expresses the problem of evaluating and interpolating polynomials the LCH-basis over $\mathbb{F}_{p^r}$. Section 3 gives useful prerequisites. Then, Section 4 expresses the evaluation map by a matrix and presents our factorization into the product of sparse matrices. The matrix of the inverse map is also exhibited. Section 5 shows how to derive fast algorithms for multipoint evaluation and interpolation of polynomials in the LCH-basis over $\mathbb{F}_{p^r}$. Experimental results are given in Section 6 and finally Section 7 concludes the paper.

2 DESCRIPTION OF THE PROBLEM

The first three paragraphs present material introduced in [12, 13]. Let $v_0,\dots,v_{r-1}$ be a basis of $\mathbb{F}_{p^r}$ over $\mathbb{F}_p$, *i.e.* $\langle v_0,\dots,v_{r-1}\rangle = \mathbb{F}_{p^r}$ and let $e_0,\dots,e_{r-1}$ be the canonical basis of $\mathbb{F}_p^r$, *i.e.*, $\langle e_0,\dots,e_{r-1}\rangle = \mathbb{F}_p^r$ where any basis element e_k is a vector $(e_{0,k},\dots,e_{r-1,k})$ with $e_{i,k}=1$ if $k=i$ and 0 otherwise.

The LCH-basis over $\mathbb{F}_{p^r}$. For any $i,j\geq 0$ such that $i+j\leq r$, let U_i^j and V_i^j be respectively subspaces of $\mathbb{F}_{p^r}$ and $\mathbb{F}_p^r$ defined as

$$U_i^j = \langle v_i,\dots,v_{i+j-1}\rangle, \quad V_i^j = \langle e_i,\dots,e_{i+j-1}\rangle\,.$$

Observe that they form stricly ascending chains. Namely,

$$U_0^0\subset U_0^1\subset\ldots\subset U_0^r=\mathbb{F}_{p^r} \text{ and } V_0^0\subset V_0^1\subset\ldots\subset V_0^r=\mathbb{F}_p^r\,,$$

with $U_0^0=V_0^0=\{0\}$. Now let $L_{U_0^k}$ be linearized polynomials defined over $\mathbb{F}_{p^r}$ and of degree p^k such that $\ker\left(L_{U_0^k}\right)=U_0^k$. They can be recursively defined from $L_{U_0^0}(x)=x$ by the following relation

$$L_{U_0^{k+1}}(x) = L_{U_0^k}(x)^p - L_{U_0^k}(v_k)^{p-1}\cdot L_{U_0^k}(x)\,. \tag{1}$$

and they are linear (see [1]). Note that $L_{U_0^k}$ is invariant over the cosets of U_0^k in $\mathbb{F}_{p^r}$. Namely, for any $x\in U_0^k=\langle v_0,\dots,v_{k-1}\rangle$ and for any $y\in U_k^{r-k}=\langle v_k,\dots,v_{r-1}\rangle$,

$$L_{U_0^k}(x+y)=L_{U_0^k}(y)\,.$$

For any $x\in\mathbb{F}_{p^r}$ and for any $\omega=(\omega_0,\dots,\omega_{r-1})\in\mathbb{F}_p^r$ define

$$\chi_\omega(x)=\prod_{k=0}^{r-1} L_{U_0^k}(x)^{\omega_k}\,.$$

Identifying $\omega_j\in\mathbb{F}_p$ to its minimal representative over $\mathbb{N}$, the degree of $\chi_\omega(x)$ is therefore $\sum_{j=0}^{r-1}\omega_j\cdot p^j$.

Invariance of χ_ω. From the invariance property of $L_{U_0^k}$ and the fact that $U_0^k\subset U_0^{k+1}$ for any $k\in\{0,\dots,r-1\}$, it follows that $L_{U_0^k}$ is invariant over the cosets of U_0^s with $s\leq k$. Also, for any $\omega=(\omega_0,\dots,\omega_{r-1})\in V_i^j$, we have $\omega_k=0$ for any $k<i$. Therefore, $\chi_\omega(x)=\prod_{k=0}^{r-1}L_{U_0^k}(x)^{\omega_k}=\prod_{k=i}^{r-1}L_{U_0^k}(x)^{\omega_k}$. It turns out that χ_ω is invariant over the intersection of the set of cosets of U_0^s with $s\leq k$ for $k\in\{i,\dots,r-1\}$. This last intersection are the cosets of U_0^t with $t\leq i$.

Polynomials in the LCH-basis over $\mathbb{F}_{p^r}$. Any polynomial of $\mathbb{F}_{p^r}[x]/(x^{p^r}-x)$ can be written in the LCH-basis as

$$P(x)=\sum_{\omega\in\mathbb{F}_p^r}\alpha_\omega\cdot\chi_\omega(x)\,, \tag{2}$$

with $\alpha_\omega\in\mathbb{F}_{p^r}$ for any $\omega\in\mathbb{F}_p^r$. The classical multipoint evaluation problem consists in evaluating $P(x)$ for any $x\in\mathbb{F}_{p^r}$. The classical interpolation problem consists in recovering the coefficients α_ω with $\omega\in\mathbb{F}_p^r$ from the values $P(x)$ for any $x\in\mathbb{F}_{p^r}$.

In [12], the authors consider a variation of the problem where the polynomials are of the shape

$$\sum_{\omega\in V}\alpha_\omega\cdot\chi_\omega(x) \tag{3}$$

and the set of evaluation is $U+\mu$ with $\mu\in\mathbb{F}_{p^r}$, U and V being respectively subspaces of $\mathbb{F}_{p^r}$ and $\mathbb{F}_p^r$ with same cardinalities. A divide-and-conquer approach was first proposed in [13] for solving this multipoint evaluation problem over $\mathbb{F}_{2^r}$. By backtracking their method, the same authors derived an algorithm for the interpolation problem. The forward approach was then generalized to characteristic p in [12]. However, no method was provided for solving the interpolation problem over $\mathbb{F}_{p^r}$.

3 PREREQUISITES

In this section, we first define an ordering over the subspaces that are involved in the evaluation of some polynomial represented in the LCH-basis . Then, we remind some definitions about matrices and we state two results about them that are based on the chosen order.

Ordering. Consider a finite totally ordered set $(S,\leq)$ such that $\#S=n$, i.e. $S=\{s_0,s_1,\dots,s_{n-1}\}$ such that $s_i\leq s_j\Leftrightarrow i\leq j$. We define the *index* application $ind: S\mapsto\mathbb{N}$ defined by $ind(s_i)=i$ for any $i=0,\dots,n-1$.

Also for any arbitrary total order on $\mathbb{F}_p$, in the rest of the paper we consider the lexicographic order denoted by $\leq$ on $\mathbb{F}_p^r$. We may also define a total order on $\mathbb{F}_{p^r}$ by requiring that

$$\phi(\underline{x})\overset{\text{def}}{\leq}\phi(\underline{y})\Leftrightarrow\underline{x}\leq\underline{y} \text{ for any } \underline{x},\underline{y}\in\mathbb{F}_p^r\,.$$

The symbol $\leq$ is used indifferently to refer to the total order on (subsets) of $\mathbb{F}_p^r$ and $\mathbb{F}_{p^r}$.

Matrices. Consider two finite totally ordered subsets, i.e. $S_1=\{x_0,\dots,x_{n-1}\}$ and $S_2=\{y_0,\dots,y_{m-1}\}$ where the elements are respectively written from the smallest to the largest. In what follows

we handle matrices over $\mathbb{F}_{p^r}$. The matrix A is defined by

$$\left(\left(\mathsf{A}(x,y)\right)_{x\in S_1, y\in S_2} \triangleq \begin{pmatrix} a_{x_0,y_0} & a_{x_0,y_1} & \cdots & a_{x_0,y_{m-1}} \\ a_{x_1,y_0} & a_{x_1,y_1} & \cdots & a_{x_1,y_{m-1}} \\ \vdots & \vdots & \ddots & \vdots \\ a_{x_{n-1},y_0} & a_{x_{n-1},y_1} & \cdots & a_{x_{n-1},y_{m-1}} \end{pmatrix}.$$

The symbol $\otimes$ denotes the Kronecker product of matrices. The product of several matrices $\prod_{i=0}^{n} A_i$ stands for $A_0 \cdot A_1 \cdot \ldots \cdot A_n$. Finally, $\oplus_{i\in S} A_i$ denotes the block diagonal matrix $\mathrm{diag}(A_{s_0}, \ldots, A_{s_{n-1}})$. These notations can be found in [8]. The following well-known lemma can be found in [4].

Lemma 3.1. *For any two totally ordered sets $(S_1, \leq_1)$ and $(S_2, \leq_2)$, consider the sets $S_1 \times S_2$ and $S_2 \times S_1$ together with their respective lexicographic orders. Consider the matrix P defined by*

$$P = (P(x,y))_{x\in S_2\times S_1, y\in S_1\times S_2} \quad \text{with } P(x,y) = \begin{cases} 1 & \text{if } \pi(y) = x \\ 0 & \text{otherwise} \end{cases}$$

where $\pi : S_1 \times S_2 \longrightarrow S_2 \times S_1 : (s_1, s_2) \longrightarrow (s_2, s_1)$. Also, for any matrix $A = (A(x,y))_{x,y\in S_2\times S_1}$, we have

$$P^{\mathsf{T}} \cdot A \cdot P = (A(\pi(x), \pi(y)))_{x,y\in S_1\times S_2}\ . \tag{4}$$

Also, note that for any $(s_1, s_2), (s_3, s_4) \in S_1 \times S_2$ and A such that

$$A = \left(A((s_1,s_2),(s_3,s_4))\right)_{(s_1,s_2),(s_3,s_4)\in S_1\times S_2}\ ,$$

we then have

$$A = \left(\left(A((s_1,s_2),(s_3,s_4))\right)_{s_2,s_4\in S_2}\right)_{s_1,s_3\in S_1}\ . \tag{5}$$

Proof. Let us begin by proving relation (4). For any $z \in S_2 \times S_1$ and for any $y \in S_1 \times S_2$,

$$(\mathsf{A}\cdot\mathsf{P})(z,y) = \sum_{u\in S_2\times S_1} \mathsf{A}(z,u)\cdot\mathsf{P}(u,y) = \mathsf{A}(z,\pi(y))\,.$$

Therefore, for any $x, y \in S_1 \times S_2$,

$$(\mathsf{P}^{\mathsf{T}}\cdot\mathsf{A}\cdot\mathsf{P})(x,y) = \sum_{z\in S_2\times S_1} \mathsf{A}(z,\pi(y))\cdot\mathsf{P}(z,x) = \mathsf{A}(\pi(x),\pi(y))\,.$$

Let us now prove relation (5). For any two totally ordered sets $(S_1, \leq_1), (S_2, \leq_2)$ with $\#S_1 = n_1, \#S_2 = n_2$ and the lexicographic order on $S_1 \times S_2$, the index of $(s_1, s_2) \in S_1 \times S_2$ is given by

$$ind(s_1,s_2) = ind(s_1)\cdot\#S_2 + ind(s_2)\,. \tag{6}$$

Therefore, $A = \left(A((s_1,s_2),(s_3,s_4))\right)_{(s_1,s_2),(s_3,s_4)\in S_1\times S_2}$ is composed of $n_1 \times n_1$ blocks each of size $\#S_2 \times \#S_2$. Therefore,

$$\mathsf{A} = \left(\left(\mathsf{A}((s_1,s_2),(s_3,s_4))\right)_{s_2,s_4\in S_2}\right)_{s_1,s_3\in S_1}\ .$$

□

Now, consider the map

$$\pi_{1,j} : \mathbb{F}_p \times \mathbb{F}_p^{j-1} \mapsto \mathbb{F}_p^{j-1} \times \mathbb{F}_p : (s,t) \mapsto \pi(s,t) = (t,s)\,.$$

Identify the spaces $\mathbb{F}_p \times \mathbb{F}_p^{j-1}, \mathbb{F}_p^{j-1} \times \mathbb{F}_p$ and $\mathbb{F}_p^j$. Define π_0 as the identity map and

$$\pi_{k,j} = \begin{cases} \underbrace{\pi_{1,j}\circ\pi_{1,j}\circ\ldots\circ\pi_{1,j}}_{k} & \text{if } k > 0\,, \\ (\pi_{-k,j})^{-1} & \text{if } k < 0\,. \end{cases}$$

For any $k \in \mathbb{Z}$, define $\mathsf{P}_{k,j} = \left(\mathsf{P}_{k,j}(x,y)\right)_{x,y\in\mathbb{F}_p^j}$ where

$$\mathsf{P}_{k,j}(x,y) = \begin{cases} 1 & \text{if } \pi_{k,j}(y) = x \\ 0 & \text{otherwise.} \end{cases}$$

The second part of the following corollary can be found in [4].

Corollary 3.2. *We have $P_{j,j} = I_{p^j}$ and $P_{k,j} = P_{1,j}^{k \bmod j}$ for any $k \in \mathbb{Z}$. Moreover, if $A = (A(x,y))_{x,y\in\mathbb{F}_p^k}$ and $B = (B(x,y))_{x,y\in\mathbb{F}_p^{j-k}}$, then*

$$P_{k,j}^{\mathsf{T}} \cdot (B\otimes A)\cdot P_{k,j} = A\otimes B\,.$$

where $\mathbb{F}_p^k \times \mathbb{F}_p^{j-k}, \mathbb{F}_p^{j-k} \times \mathbb{F}_p^k$ and $\mathbb{F}_p^j$ are identified.

Proof. By definition of $\mathsf{P}_{k,j}$ and $\pi_{0,j}$, we have $\mathsf{P}_{0,j} = \mathsf{I}_{p^j}$. Also, for any $x, y \in \mathbb{F}_{p^j}$ and any $k \geq 1$

$$\left(\mathsf{P}_{k-1,j}\cdot\mathsf{P}_{1,j}\right)(x,y) = \sum_{v\in\mathbb{F}_p^j} \mathsf{P}_{k-1,j}(x,v)\cdot\mathsf{P}_{1,j}(v,y) = \begin{cases} 1 & \text{if } \pi_{k,j}(y) = x\,, \\ 0 & \text{otherwise.} \end{cases}$$

Therefore, $\mathsf{P}_{k,j} = \mathsf{P}_{k-1,j}\cdot\mathsf{P}_{1,j}$. It follows that $\mathsf{P}_{k,j} = \mathsf{P}_{1,j}^k$ for any $k \geq 0$. Observing that $\pi_{j,j} = \underbrace{\pi_{1,j}\circ\ldots\circ\pi_{1,j}}_{j} = \mathrm{id}$, we have $\mathsf{P}_{j,j} = \mathsf{I}_{p^j}$. Therefore, $\mathsf{P}_{k,j} = \mathsf{P}_{1,j}^{k \bmod j}$ for any $k \geq 0$. According to the definition of $\mathsf{P}_{k,j}$ and $\pi_{k,j}$, for any $k < 0, \mathsf{P}_{k,j} = \left(\mathsf{P}_{-k,j}\right)^{-1}$. The element $-k$ being positive, we have $\mathsf{P}_{-k,j} = \mathsf{P}_{1,j}^{-k}$. Therefore, $\mathsf{P}_{k,j} = \left(\mathsf{P}_{1,j}^{-k}\right)^{-1} = \left(\mathsf{P}_{1,j}^{-1}\right)^{-k}$. Also, $\mathsf{P}_{j,j} = \mathsf{P}_{1,j}^j = \mathsf{I}_{p^j}$. It follows that $\mathsf{P}_{1,j}^{-1} = \mathsf{P}_{1,j}^{j-1}$. It turns out that $\mathsf{P}_{k,j} = \left(\mathsf{P}_{1,j}^{j-1}\right)^{-k} = \mathsf{P}_{1,j}^{(j-1)\cdot(-k)} = \mathsf{P}_{1,j}^{(j-1)\cdot(-k) \bmod j} = \mathsf{P}_{1,j}^{k \bmod j}$.

The second part of the corollary follows immediately from Lemma 3.1. □

4 A FACTORIZATION OF TWO MATRICES

In this section, we start by reducing the multipoint evaluation and interpolation problems expressed in Section 2 to a canonical form. Then we associate a matrix to the multipoint evaluation map. We give a factorization of this matrix into the product of sparse matrices. This factorization allows us to derive its inverse matrix which corresponds to the inverse map. Finally, we show that the inverse matrix also factorizes into the product of sparse matrices.

Canonical form. Consider fixed integers i and j such that $i, j \geq 0$ and $i + j \leq r$. Remember that $U_i^j = \langle v_i, \ldots, v_{i+j-1}\rangle$ and $V_i^j = \langle e_i, \ldots, e_{i+j-1}\rangle$. In what follows $U_0^i, U_i^j, U_{i+j}^{r-(i+j)}$ and V_i^j will be denoted respectively by U_L, U, U_R and V for the sake of clarity. Note that $\mathbb{F}_{p^r} = U_L \oplus U \oplus U_R$. Consider the multipoint evaluation and interpolation problems on a subset of polynomials of the shape

$$P_V(x) = \sum_{\omega\in V} \alpha_\omega\cdot\chi_\omega(x) \quad \text{for any } x \in U + \mu\,. \tag{7}$$

Note that because the vector basis $(v_i)_i$ of $\mathbb{F}_{p^r}$ may be freely chosen, the space U may be any vector space of $\mathbb{F}_{p^r}$ with dimension j.

Let us show that these problems may be reduced to a canonical form. Without loss of generality, we may assume that $\mu \in U_L \oplus U_R$. For any $x \in U$

$$P_V(x+\mu) = \sum_{\omega \in V} \alpha_\omega \cdot \chi_\omega(x+\mu)\,.$$

Writting μ as $\mu_L + \mu_R$ where $\mu_L \in U_L$ and $\mu_R \in U_R$, we have for any $\omega \in V$ and any $x \in U$ that

$$\chi_\omega(x+\mu_L+\mu_R) = \chi_\omega(x+\mu_R)$$

because χ_ω is invariant on the cosets of U_L. Hence for any $x \in U$

$$P_V(x+\mu) = \sum_{\omega \in V} \alpha_\omega \cdot \chi_\omega(x+\mu_R)\,.$$

We deduce that evaluating $P_V(x)$ over $U+\mu$ reduces to the evaluation of P_V over $U+\mu_R$.

Matrices of evaluation and interpolation maps. Given $n = p^j$ points $x \in U + \mu_R$, we define the linear evaluation map $E : \mathbb{F}_{p^r}^n \mapsto \mathbb{F}_{p^r}^n$ by $E((\alpha_\omega)_{\omega\in V}) = (P_V(x))_{x\in U+\mu_R}$. This linear map can be represented by the matrix

$$\mathsf{X} \triangleq \left(\chi_\omega(x)\right)_{x\in U+\mu_R,\,\omega\in V}\,. \tag{8}$$

A factorization of X for fast evaluation. Let us present a factorization of X into the product of sparse matrices. Let us first introduce descending chains of $U+\mu_R$ and V. More precisely, for any $k \in \{0,\dots,j-1\}, U_k = \langle v_{i+k},\dots,v_{i+j-1}\rangle + \mu_R$ and $V_k = \langle e_{i+k},\dots,e_{i+j-1}\rangle$. Let $U_j = \{\mu_R\}$ and $V_j = (0,\dots,0)$. We have $U_s \subset U_t$ and $V_s \subset V_t$ if $s > t$, also $U_0 = U + \mu_R$ and $V_0 = V$. Also, let us define the increasing chain of sets $U_{L,k} = U_0^{i+k}$ for $k \in \{0,\dots,j-1\}$. These sets are related by the invariant $U_{L,k} \oplus U_k = U_L \oplus U_0 = U_0^{i+j} + \mu_R$ for any k which expresses different ways to decompose the affine space $U_0^{i+j} + \mu_R$. Finally, these sets are totally ordered according to Section 3.

Remark. Relation (9) in the following theorem gives essentially a matrix expression of the recurrence relation behind the divide-and-conquer approach developed in [12]. Its proof also relies on the invariance property of the LCH-basis .

Theorem 4.1. *Consider the linearized polynomials $L_k = L_{U_{L,k}}$ defined such that $\ker(L_k) = U_{L,k}$ for $k = 0,\dots,j-1$. The matrices X_k defined by $(\chi_\omega(x))_{x\in U_k,\,\omega\in V_k}$ satisfy the recursion*

$$X_k = \left(\bigoplus_{x\in U_{k+1}} V_k(x)\right) \cdot P_{1,j-k} \cdot \left(I_p \otimes X_{k+1}\right) \cdot P_{1,j-k}^{\mathsf{T}}\,. \tag{9}$$

for any $k = 0,\dots,j-1$ where V_k is the Vandermonde matrix defined as

$$V_k(x) = \left(L_k(x + c\cdot v_{i+k})^d\right)_{c,d\in\mathbb{F}_p}$$

for any $x \in U_{k+1}$ and $P_{1,j-k}$ is defined in Section 3. Also, $X = X_0$ and we have that

$$X = \prod_{k=0}^{j-1}\left(\left(\bigoplus_{x\in U_{k+1}} V_k(x)\right) \otimes I_{p^k}\right), \tag{10}$$

or equivalently

$$X = \prod_{k=0}^{j-1}\left(\left(\bigoplus_{x\in U_{k+1}} \left(I_{p^k} \otimes V_k(x)\right)\right) \cdot B_k\right) \tag{11}$$

where

$$B_k = \begin{cases} I_{p^{j-(k+2)}} \otimes \left(\left(I_p \otimes P_{1,k+1}\right)\cdot P_{1,k+2}^{\mathsf{T}}\right) & \text{if } 0 \le k < j-1\,,\\ P_{1,j} & \text{if } k = j-1\,. \end{cases} \tag{12}$$

and $P_{1,j}$, $P_{1,k+1}$, $P_{1,k+2}$ are defined in Section 3.

Proof. Let us prove relation (9) by first showing that

$$\mathsf{X}_k = \left(\bigoplus_{x\in U_{k+1}} \mathsf{V}_k(x)\right) \cdot \left(\mathsf{X}_{k+1} \otimes \mathsf{I}_p\right)\,. \tag{13}$$

In what follows $\delta_b(d)$ denotes the *Kronecker delta function, i.e.* $\delta_b(d) = 1$ if $b = d$ and 0 otherwise. On the one hand we have

$$\left(\bigoplus_{x\in U_{k+1}} \mathsf{V}_k(x)\right) = \left(\delta_x(z)\cdot\left(\left(L_k(x+c\cdot v_{i+k})\right)^b\right)_{c,b\in\mathbb{F}_p}\right)_{x,\,z\in U_{k+1}}\,.$$

Also, applying Lemma 3.1 on the right-hand side gives

$$\left(\delta_x(z)\cdot\left(L_k(x+c\cdot v_{i+k})\right)^b\right)_{(x,c),(z,b)\in U_{k+1}\times\mathbb{F}_p}\,.$$

Therefore for any $(x,c),(z,b) \in U_{k+1}\times\mathbb{F}_p$,

$$\left(\bigoplus_{x\in U_{k+1}} \mathsf{V}_k(x)\right)_{(x,c),(z,b)} = \delta_x(z)\cdot\left(L_k(x+c\cdot v_{i+k})\right)^b\,. \tag{14}$$

On the other hand we have

$$\begin{aligned}\mathsf{X}_{k+1}\otimes\mathsf{I}_p &= \left(\chi_\omega(z)\cdot\mathsf{I}_p\right)_{z\in U_{k+1},\,\omega\in V_{k+1}}\\ &= \left(\chi_{\phi_{k+1}(\omega)}(z)\cdot\mathsf{I}_p\right)_{z,\,\omega\in U_{k+1}}\end{aligned}$$

where

$$\phi_k : V_k \mapsto U_k : \omega = \sum_{s=i+k}^{i+j-1}\omega_s\cdot e_s \mapsto x = \left(\sum_{s=i+k}^{i+j-1}\omega_s\cdot v_s\right) + \mu_R\,.$$

We stress that this bijection preserves the order of all elements. Also, for any $z,\omega \in U_{k+1}$ we have

$$\left(\chi_{\phi_{k+1}(\omega)}(z)\cdot\mathsf{I}_p\right)_{z,\,\omega} = \left(\chi_{\phi_{k+1}(\omega)}(z)\cdot\left(\delta_b(d)\right)_{b,d\in\mathbb{F}_p}\right)_{z,\,\omega}\,,$$

Also, by Lemma 3.1, the right-hand side gives

$$\left(\chi_{\phi_{k+1}(\omega)}(z)\cdot\delta_b(d)\right)_{(z,b),(\omega,d)\in U_{k+1}\times\mathbb{F}_p}$$

It turns out that for any $(z,b) \in U_{k+1}\times\mathbb{F}_p$, for any $(\omega,d) \in V_{k+1}\times\mathbb{F}_p$,

$$\left(\mathsf{X}_{k+1}\otimes I_p\right)_{(z,b),(\omega,d)} = \chi_\omega(z)\cdot\delta_b(d)\,. \tag{15}$$

For any $(x,c),(z,b) \in U_{k+1}\times\mathbb{F}_p$, for any $(\omega,d) \in V_{k+1}\times\mathbb{F}_p$ we have that

$$\begin{aligned}&\left(\left(\bigoplus_{x\in U_{k+1}} \mathsf{V}_k(x)\right)\cdot\left(\mathsf{X}_{k+1}\otimes\mathsf{I}_p\right)\right)_{(x,c),(\omega,d)} =\\ &\qquad\sum_{z,b}\left(\bigoplus_{x\in U_{k+1}} \mathsf{V}_k(x)\right)_{(x,c),\,(z,b)}\cdot\left(\mathsf{X}_{k+1}\otimes\mathsf{I}_p\right)_{(z,b),\,(\omega,d)}\end{aligned}$$

and from (15) and (14), the right-hand side reduces to

$$\sum_{z,b}\delta_x(z)\cdot\left(L_k(x+c\cdot v_{i+k})\right)^b\cdot\chi_\omega(z)\cdot\delta_b(d)\,.$$

Therefore,

$$\left(\left(\bigoplus_{x\in U_{k+1}} \mathsf{V}_k(x)\right)\cdot(\mathsf{X}_{k+1}\otimes \mathsf{I}_p)\right)_{(x,c),(\omega,d)} = L_k(x+c\cdot v_{i+k})^d\cdot\chi_\omega(x)\,. \tag{16}$$

Observe that for any $\omega\in V_{k+1}$, $\chi_\omega(x)$ is invariant over the cosets of $U_{L,k+1}$. Also, $v_{i+k}\in U_{L,k+1}=\langle v_0,\ldots,v_{i+k}\rangle$. Thus,

$$L_k(x+c\cdot v_{i+k})^d\cdot\chi_\omega(x) = L_k(x+c\cdot v_{i+k})^d\cdot\chi_\omega(x+c\cdot v_{i+k})\,.$$

Note that $L_k(x+c\cdot v_{i+k})^d = \chi_{d\cdot e_{i+k}}(x+c\cdot v_{i+k})$ and therefore we have that

$$\begin{aligned}L_k(x+c\cdot v_{i+k})^d\cdot\chi_\omega(x) &= \chi_{d\cdot e_{i+k}}(x+c\cdot v_{i+k})\cdot\chi_\omega(x+c\cdot v_{i+k})\\ &= \chi_{\omega+d\cdot e_{i+k}}(x+c\cdot v_{i+k})\end{aligned}$$

where $\chi_{\omega+d\cdot e_{i+k}}(x+c\cdot v_{i+k}) = (\mathsf{X}_k)_{(x,c),\,(\omega,d)}$. Hence,

$$L_k(x+c\cdot v_{i+k})^d\cdot\chi_\omega(x) = (\mathsf{X}_k)_{(x,c),\,(\omega,d)}\ .$$

This proves (13). Finally, by gathering relation (16) with the above relation and applying corollary 3.2

$$\mathsf{X}_{k+1}\otimes\mathsf{I}_p = \mathsf{P}_{1,j-k}\cdot(\mathsf{I}_p\otimes\mathsf{X}_{k+1})\cdot\mathsf{P}^\intercal_{1,j-k}$$

which concludes the proof of relation (9).

Let us now prove relation (10) by induction. Consider the base case $k=j$. By definition, $\mathsf{X}_j=\left(\chi_\omega(x)\right)_{x\in U_j,\,\omega\in V_j}$ with $U_j=\{\mu_R\}$ and $V_j=\{(0,0,\ldots,0)\}$. Hence $\mathsf{X}_j=\prod_{k=0}^{r-1}\left(L_{U_{L,k}}(\mu_R)\right)^0=1$. Also, $\mathsf{X}_j=\prod\emptyset=1$. Now, assume it holds that

$$\mathsf{X}_n = \prod_{k=n}^{j-1}\left(\left(\bigoplus_{x\in U_{k+1}}\mathsf{V}_k(x)\right)\otimes\mathsf{I}_{p^{k-n}}\right)\,.$$

Starting by applying relation (13), we have

$$\mathsf{X}_{n-1} = \left(\bigoplus_{x\in U_n}\mathsf{V}_{n-1}(x)\right)\cdot(\mathsf{X}_n\otimes\mathsf{I}_p)\ .$$

Then, using the induction hypothesis we can write

$$\mathsf{X}_{n-1} = \left(\bigoplus_{x\in U_n}\mathsf{V}_{n-1}(x)\right)\cdot\left(\prod_{k=n}^{j-1}\left(\left(\bigoplus_{x\in U_{k+1}}\mathsf{V}_k(x)\right)\otimes\mathsf{I}_{p^{k-n}}\right)\otimes\mathsf{I}_p\right)\,.$$

Noting that $\mathsf{I}_p=\prod_{k=n}^{j-1}\mathsf{I}_p$ and $\prod_i\mathsf{A}_i\otimes\prod_i\mathsf{B}_i=\prod_i(\mathsf{A}_i\otimes\mathsf{B}_i)$,

$$\begin{aligned}\prod_{k=n}^{j-1}\left(\left(\bigoplus_{x\in U_{k+1}}\mathsf{V}_k(x)\right)\otimes\mathsf{I}_{p^{k-n}}\right)\otimes\mathsf{I}_p&\\ = \prod_{k=n}^{j-1}\left(\left(\bigoplus_{x\in U_{k+1}}\mathsf{V}_k(x)\right)\otimes\mathsf{I}_{p^{k-n}}\otimes\mathsf{I}_p\right)\,.&\end{aligned}$$

Also $\mathsf{I}_{p^{k-n}}\otimes\mathsf{I}_p=\mathsf{I}_{p^{k-n+1}}$, therefore

$$\mathsf{X}_{n-1} = \left(\bigoplus_{x\in U_n}\mathsf{V}_{n-1}(x)\right)\cdot\left(\prod_{k=n}^{j-1}\left(\left(\bigoplus_{x\in U_{k+1}}\mathsf{V}_k(x)\right)\otimes\mathsf{I}_{p^{k-n+1}}\right)\right)\,.$$

Since for any matrix A, $\mathsf{A}\otimes\mathsf{I}_1=\mathsf{A}$ and $\mathsf{I}_1=\mathsf{I}_{p^0}$, then

$$\mathsf{X}_{n-1} = \left(\left(\bigoplus_{x\in U_n}\mathsf{V}_{n-1}(x)\right)\otimes\mathsf{I}_{p^0}\right)\cdot\left(\prod_{k=n}^{j-1}\left(\left(\bigoplus_{x\in U_{k+1}}\mathsf{V}_k(x)\right)\otimes\mathsf{I}_{p^{k-n+1}}\right)\right)\,,$$

or equivalently

$$\mathsf{X}_{n-1} = \prod_{k=n-1}^{j-1}\left(\left(\bigoplus_{x\in U_{k+1}}\mathsf{V}_k(x)\right)\otimes I_{p^{k-n+1}}\right)\,.$$

Let us now prove relation (11). Observe that

$$\mathsf{X} = \prod_{k=0}^{j-1}\left(\bigoplus_{x\in U_{k+1}}\mathsf{V}_k(x)\right)\otimes\mathsf{I}_{p^k} = \prod_{k=0}^{j-1}\left(\bigoplus_{x\in U_{k+1}}\left(\mathsf{V}_k(x)\otimes\mathsf{I}_{p^k}\right)\right)\,.$$

Also, according to corollary 3.2,

$$\mathsf{V}_k(x)\otimes\mathsf{I}_{p^k} = \mathsf{P}^\intercal_{1,k+1}\cdot\left(\mathsf{I}_{p^k}\otimes\mathsf{V}_k(x)\right)\cdot\mathsf{P}_{1,k+1}\,.$$

Therefore,

$$\mathsf{X} = \prod_{k=0}^{j-1}\left(\bigoplus_{x\in U_{k+1}}\left(\mathsf{P}^\intercal_{1,k+1}\left(\mathsf{I}_{p^k}\otimes\mathsf{V}_k(x)\right)\cdot\mathsf{P}_{1,k+1}\right)\right)\,.$$

Noting that

$$\begin{aligned}\bigoplus_{x\in U_{k+1}}&\left(\mathsf{P}^\intercal_{1,k+1}\left(\mathsf{I}_{p^k}\otimes\mathsf{V}_k(x)\right)\cdot\mathsf{P}_{1,k+1}\right)\\ &= \mathsf{C}_k^\intercal\cdot\left(\bigoplus_{x\in U_{k+1}}\left(\mathsf{I}_{p^k}\otimes\mathsf{V}_k(x)\right)\right)\cdot\mathsf{C}_k\end{aligned}$$

where $\mathsf{C}_k=\left(\mathsf{I}_{p^{j-(k+1)}}\otimes\mathsf{P}_{1,k+1}\right)$ and $\mathsf{C}_k^\intercal=\left(\mathsf{I}_{p^{j-(k+1)}}\otimes\mathsf{P}^\intercal_{1,k+1}\right)$, we have

$$\mathsf{X} = \prod_{k=0}^{j-1}\left(\mathsf{C}_k^\intercal\cdot\left(\bigoplus_{x\in U_{k+1}}\left(\mathsf{I}_{p^k}\otimes\mathsf{V}_k(x)\right)\right)\cdot\mathsf{C}_k\right)\,.$$

Let us write this last formula in another way by combining successive factors. Let us evaluate C_0^T. We have

$$\mathsf{I}_{p^{j-(0+1)}}\otimes\mathsf{P}^\intercal_{1,0+1} = \mathsf{I}_{p^{j-1}}\otimes\mathsf{P}^\intercal_{1,1} = \mathsf{I}_{p^{j-1}}\otimes\mathsf{I}_p = \mathsf{I}_{p^j}\,.$$

Also C_{j-1} is

$$\mathsf{I}_{p^{j-j}}\otimes\mathsf{P}_{1,j} = \mathsf{I}_1\otimes\mathsf{P}_{1,j} = \mathsf{P}_{1,j}\,.$$

Finally, for any k such that $0\le k<j-1$, we have

$$\begin{aligned}\mathsf{C}_k\cdot\mathsf{C}_{k+1}^\intercal &= \left(\mathsf{I}_{p^{j-(k+1)}}\otimes\mathsf{P}_{1,k+1}\right)\cdot\left(\mathsf{I}_{p^{j-(k+2)}}\otimes\mathsf{P}^\intercal_{1,k+2}\right)\\ &= \left(\mathsf{I}_{p^{j-(k+2)}}\otimes\mathsf{I}_p\otimes\mathsf{P}_{1,k+1}\right)\cdot\left(\mathsf{I}_{p^{j-(k+2)}}\otimes\mathsf{P}^\intercal_{1,k+2}\right)\\ &= \mathsf{I}_{p^{j-(k+2)}}\otimes\left(\left(\mathsf{I}_p\otimes\mathsf{P}_{1,k+1}\right)\cdot\mathsf{P}^\intercal_{1,k+2}\right)\,.\end{aligned}$$

The result follows. □

A factorization of X^{-1} for fast interpolation. As explained previously, X^{-1} is the matrix of the interpolation map for which the following corollary gives a factorization into the product of sparse matrices.

Corollary 4.2. *The matrix X as defined in theorem 4.1 is invertible and the inverse matrix X^{-1} is given by*

$$X^{-1} = \prod_{k=0}^{j-1}\left(\bigoplus_{x\in U_{j-k}}V^{-1}_{j-k-1}(x)\right)\otimes I_{p^k} \tag{17}$$

or equivalently

$$X^{-1} = \prod_{k=0}^{j-1}\left(B^{-1}_{j-k-1}\cdot\left(\bigoplus_{x\in U_{j-k}}\left(I_{p^{j-k-1}}\otimes V^{-1}_{j-k-1}(x)\right)\right)\right) \tag{18}$$

where

$$B_k^{-1} = \begin{cases} I_{p^{j-(k+2)}} \otimes \left(P_{1,k+2} \cdot \left(I_p \otimes P_{1,k+1}^\intercal \right) \right) & \text{if } 0 \leq k < j-1\,, \\ P_{1,j}^\intercal & \text{if } k = j-1\,. \end{cases} \tag{19}$$

and $P_{1,j}$, $P_{1,k+1}$, $P_{1,k+2}$ are defined in Section 3.

Proof. By theorem 4.1, we have

$$\mathsf{X} = \prod_{k=0}^{j-1} \left(\bigoplus_{x \in U_{k+1}} \mathsf{V}_k(x) \right) \otimes \mathsf{I}_{p^k}\,.$$

Therefore, we have

$$\mathsf{X}^{-1} = \prod_{k=0}^{j-1} \left(\bigoplus_{x \in U_{j-k}} \mathsf{V}_{j-k-1}^{-1}(x) \right) \otimes \mathsf{I}_{p^k}\,.$$

Let us now prove relation (18). According to theorem 4.1,

$$\mathsf{X} = \prod_{k=0}^{j-1} \left(\left(\bigoplus_{x \in U_{k+1}} \left(\mathsf{I}_{p^k} \otimes \mathsf{V}_k(x) \right) \right) \cdot \mathsf{B}_k \right)$$

where

$$\mathsf{B}_k = \begin{cases} \mathsf{I}_{p^{j-(k+2)}} \otimes \left(\left(\mathsf{I}_p \otimes \mathsf{P}_{1,k+1} \right) \cdot \mathsf{P}_{1,k+2}^\intercal \right) & \text{if } 0 \leq k < j-1\,, \\ \mathsf{P}_{1,j} & \text{if } k = j-1\,. \end{cases}$$

Therefore

$$\mathsf{X}^{-1} = \left(\prod_{k=0}^{j-1} \left(\left(\bigoplus_{x \in U_{k+1}} \left(\mathsf{I}_{p^k} \otimes \mathsf{V}_k(x) \right) \right) \cdot \mathsf{B}_k \right) \right)^{-1}\,.$$

Remembering that $(\mathsf{A} \cdot \mathsf{B})^{-1} = \mathsf{B}^{-1} \cdot \mathsf{A}^{-1}$ and $(\mathsf{A} \otimes \mathsf{B})^{-1} = \mathsf{A}^{-1} \otimes \mathsf{B}^{-1}$ we have

$$\mathsf{X}^{-1} = \prod_{k=0}^{j-1} \left(\mathsf{B}_{j-k-1}^{-1} \cdot \left(\bigoplus_{x \in U_{j-k}} \left(\mathsf{I}_{p^{j-k-1}} \otimes \mathsf{V}_{j-k-1}^{-1}(x) \right) \right) \right)\,.$$

Let us compute B_k^{-1} for $0 \leq k \leq j-1$. For any k such that $0 \leq k < j-1$, we have

$$(\mathsf{B}_k)^{-1} = \left(\mathsf{I}_{p^{j-(k+2)}} \otimes \left(\left(\mathsf{I}_p \otimes \mathsf{P}_{1,k+1} \right) \cdot \mathsf{P}_{1,k+2}^\intercal \right) \right)^{-1}\,.$$

Noting that $(\mathsf{A} \otimes \mathsf{B})^{-1} = \mathsf{A}^{-1} \otimes \mathsf{B}^{-1}$, and $(\mathsf{A} \cdot \mathsf{B})^{-1} = \mathsf{B}^{-1} \cdot \mathsf{A}^{-1}$, $\mathsf{P}_{1,k+2}^{-1} = \mathsf{P}_{1,k+2}^\intercal$ this gives

$$(\mathsf{B}_k)^{-1} = \mathsf{I}_{p^{j-(k+2)}} \otimes \left(\mathsf{P}_{1,k+2} \cdot \left(\mathsf{I}_p \otimes \mathsf{P}_{1,k+1}^\intercal \right) \right)\,.$$

Finally,

$$\mathsf{B}_{j-1}^{-1} = \left(\mathsf{P}_{1,j} \right)^{-1}\,.$$

The result follows □

5 FAST MULTIPOINT EVALUATION AND INTERPOLATION IN THE LCH-BASIS

In this section we present algorithms for fast multipoint evaluation and interpolation of polynomials represented in the LCH-basis over $\mathbb{F}_{p^r}$. These algorithms are respectively given in theorem 4.1 and corollary 4.2.

We use an *algebraic complexity model*, where the running time of an algorithm is measured in terms of the number of operations in $\mathbb{F}_{p^r}$. As customary, we use the O-notation to neglect constant factors. We denote by $\mathsf{M} : \mathbb{N} \to \mathbb{N}$ a function such that polynomials in $\mathbb{F}_{p^r}[X]$ of degree at most n can be multiplied in $\mathsf{M}(n)$ operations in $\mathbb{F}_{p^r}$. Using FFT multiplication, we can take $\mathsf{M}(n) \in O(n \log_2 n \log_2 log_2 n)$. Our algorithms rely on fast interpolation and multipoint evaluation of polynomials written in the monomial basis. Using algorithms of [5, Ch. 10], both p-point evaluation and interpolation can be done in $O(\mathsf{M}(p) \cdot \log_2 p)$.

5.1 Fast multipoint evaluation algorithm in the LCH-basis over $\mathbb{F}_{p^r}$

In what follows, we show that any sparse matrix involved in the factorized expression of the matrix X can be processed efficiently by evaluations at p points of well-defined polynomials in the monomial basis with the standard fast algorithm described above.

Consider a polynomial represented in the LCH-basis , i.e. $P_V(x) = \sum_{\omega \in V} \alpha_\omega \cdot \chi_\omega(x)$ and recall that the evaluation map at the points $x \in U + \mu_R$ is represented by the matrix $\mathsf{X} = (\chi_\omega(x))_{x \in U + \mu_R, \omega \in V}$ where U and V are totally ordered. The multipoint evaluation of $P_V(x)$ at the p^j points $x \in U + \mu_R$ therefore amounts to compute

$$(P_V(x))_{x \in U + \mu_R} = \mathsf{X} \cdot (\alpha_\omega)_{\omega \in V}\,. \tag{20}$$

From theorem 4.1, X factorizes into the product of sparse matrices which are bloc diagonal matrices for which each bloc is a Vandermonde matrix of order p. This gives

$$(P_V(x))_{x \in U + \mu_R} = \left(\prod_{k=0}^{j-1} \left(\left(\bigoplus_{x \in U_{k+1}} \left(\mathsf{I}_{p^k} \otimes \mathsf{V}_k(x) \right) \right) \cdot \mathsf{B}_k \right) \right) \cdot (\alpha_\omega)_{\omega \in V}$$

where B_k is defined as in (12).

Now by letting $\nu(j) = (\alpha_\omega)_{\omega \in V}$, for any $k = j-1 \ldots 0$ we have

$$\nu(k) = \left(\left(\bigoplus_{x \in U_{k+1}} \left(\mathsf{I}_{p^k} \otimes \mathsf{V}_k(x) \right) \right) \cdot \mathsf{B}_k \right) \cdot \nu(k+1)\,, \tag{21}$$

which gives $\nu(0) = (P_V(x))_{x \in U + \mu_R}$. Note that the bloc diagonal matrix in (21) is composed of $\#U_{k+1}$ groups of p^k identical Vandermonde matrices of order p.

Fast processing of the sparse matrices. Let us denote by $\nu(k+1, x)$ the part of the vector $B_k \cdot \nu(k+1)$ that will be multiplied by $\mathsf{I}_{p^k} \otimes V_k(x)$. This vector may be further divided into p^k vectors of length p that will be denoted by $\nu(k+1, x, l)$ for $0 \leq l < p^k$. Thus

$$\nu(k, x, l) = \mathsf{V}_k(x) \cdot \nu(k+1, x, l) \tag{22}$$

for any iteration $j > k \geq 0$, for any $x \in U_{k+1}$ and for any $0 \leq l < p^k$. Observe now that (22) corresponds to p-point evaluations of well-defined polynomials of degree less than p over $\mathbb{F}_{p^r}[X]$. Namely, let $f \in \mathbb{F}_{p^r}[X]$ be polynomials of degree less than p defined as

$$f(X) = \sum_{d \in \mathbb{F}_p} \nu(k+1, x, l)_d \cdot X^d\,. \tag{23}$$

Therefore, evaluating f at the p points of the set $S_{x,k} = \{L_k(x + c \cdot v_{i+k}) \mid c \in \mathbb{F}_p\}$ corresponds indeed to the matrix-vector product of (22). Also, for any $x \in U_{k+1}$ there is by definition an $(c_j)_j$ such that $x = \sum_{j=i+k}^{r-1} c_j \cdot v_j$ and therefore for any $k = 0, \ldots, j-1$ we

have according to the linearity of the polynomial L_k that

$$S_{x,k} = \left\{ \sum_{j=i+k}^{r-1} c_j \cdot L_k(v_j) + c \cdot L_k(v_{i+k}) \mid c \in \mathbb{F}_p \right\}. \quad (24)$$

We have the following algorithm for the multipoint evaluation of P_V at the $n = p^j$ points $x \in U + \mu_R$.

Algorithm 1 Fast multipoint evaluation of a polynomial represented in the LCH-basis over $\mathbb{F}_{p^r}$

Require: $P_V(x) = \sum_{\omega \in V} \alpha_\omega \cdot \chi_\omega(x)$ where $V = V_i^j$ and the set $U + \mu_R$ where $U = U_i^j$. The vector basis $(v_i)_i$ of $\mathbb{F}_{p^r}$.
Ensure: $\nu(0) = (P_V(x))_{x \in U+\mu_R}$.
1: Let $\nu(j) = (\alpha_\omega)_{\omega \in V}$.
2: **for** k **from** $j-1$ **down to** 0 **do**
3: Compute $\nu(k+1) = \mathsf{B}_k \cdot \nu(k+1)$.
4: **for** any $x \in U_{k+1}$ **do**
5: Compute $S_{x,k}$ as in (24).
6: **for** l **from** 0 **to** $p^k - 1$ **do**
7: Let $f(X) = \sum_{d \in \mathbb{F}_p} \nu(k+1, x, l)_d \cdot X^d$.
8: Call "Standard fast multipoint evaluation" of [5, Ch. 10].
Input : f, the set of evaluation points $S_{x,k}$.
Output : A vector $\big(f(s)\big)_{s \in S_{x,k}}$.
9: Set $\nu_k = \nu_k \,||\, \big(f(s)\big)_{s \in S_{x,k}}$.
10: **end for**
11: **end for**
12: **end for**
13: **return** $\nu(0) = (P_V(x))_{x \in U+\mu_R}$.

Complexity. In what follows, $n = p^j$ represents the size of the data in terms of elements of $\mathbb{F}_{p^r}$. The sets $S_{x,k}$ of step 5 are computed recursively thanks the recursive definition (1) of the linearized polynomials. Computing all the sets $S_{k,x}$ costs at most $O(j \cdot p^j) = O(\log_2 n \cdot n)$ operations in $\mathbb{F}_{p^r}$. Also, noting that $\#U_{k+1} = p^{j-k-1}$, it can be seen that algorithm 1 calls the subroutine Standard fast multipoint evaluation (step 8) $j \cdot p^{j-1}$ times in total. Remembering that this subroutine is in $O(\mathsf{M}(p) \log_2 p)$, therefore step 8 requires

$$c \cdot j \cdot p^{j-1}(\mathsf{M}(p) \cdot \log_2 p) \in O(j \cdot p^{j-1} \cdot (\mathsf{M}(p) \log_2 p))$$

operations in $\mathbb{F}_{p^r}$ where c is a constant. Also, by taking $\mathsf{M}(p) \in O(p \log_2 p \log_2 \log_2 p)$ and considering the cost of step 5, we have that Algorithm 1 is in

$$O(j \cdot p^j \cdot \log_2^2 p \cdot \log_2 \log_2 p) = O(n \cdot \log_p n \cdot \log_2^2 p \cdot \log_2 \log_2 p),$$

or equivalently in

$$O(n \cdot \log_2 n \cdot \log_2 p \cdot \log_2 \log_2 p).$$

5.2 Fast interpolation algorithm in the LCH-basis over $\mathbb{F}_{p^r}$

In what follows we show that the sparse matrices involved in the factorized expressions of the matrix representing the interpolation map can be processed efficiently by applying the standard algorithm for fast interpolation of [5, Ch. 10].

Consider the coefficients $(\alpha_\omega)_{\omega \in V}$ of $P_V(x) = \sum_{\omega \in V} \alpha_\omega \cdot \chi_\omega(x)$ from the set of evaluations $(P_V(x))_{x \in U+\mu_R}$ with $\mu_R \in U_R$. By (20), we have

$$(P_V(x))_{x \in U+\mu_R} = \mathsf{X} \cdot (\alpha_\omega)_{\omega \in V}.$$

Since X is invertible by corollary 4.2, and

$$\mathsf{X}^{-1} = \prod_{k=0}^{j-1} \left(\mathsf{B}_{j-k-1}^{-1} \cdot \left(\bigoplus_{x \in U_{j-k}} \left(\mathsf{I}_{p^{j-k-1}} \otimes \mathsf{V}_{j-k-1}^{-1}(x) \right) \right) \right)$$

where B_{j-k-1}^{-1} is defined as in (19), clearly

$$(\alpha_\omega)_{\omega \in V} = \mathsf{X}^{-1} \cdot (P_V(x))_{x \in U+\mu_R}.$$

Remembering that $\nu(j) = (\alpha_\omega)_{\omega \in V}$ and that $\nu(0) = (P_V(x))_{x \in U+\mu_R}$, therefore for any $k = 0 \ldots j-1$ we have

$$\nu(k+1) = \left(\mathsf{B}_{j-k-1}^{-1} \cdot \left(\bigoplus_{x \in U_{j-k}} \left(\mathsf{I}_{p^{j-k-1}} \otimes \mathsf{V}_{j-k-1}^{-1}(x) \right) \right) \right) \cdot \nu(k) \quad (25)$$

which inverses (21). In this case, the elementary operation is

$$\mathsf{V}_k^{-1}(x) \cdot \nu(k, x, l)$$

where $\nu(k, x, l)$ denotes the p elements of $\nu(k)$ that have to be multiplied with $\mathsf{V}_{j-k-1}^{-1}(x)$ with respect to (25).

Fast processing of the sparse matrices. The above elementary operation corresponds to a standard fast polynomial interpolation. More precisely, by interpolating from the evaluations $\nu(k, x, l)$ and the p points of the set $\{L_k(x + c \cdot v_{i+k}) \mid c \in \mathbb{F}_p\}$, we retrieve the polynomial f as in (23) from which we extract the coefficients $\nu(k+1, x, l)$. By doing so for any $0 \le k < j$, for any $x \in U_{k+1}$ and for any $0 \le l < p^k$, we successively retrieve all the $\nu(k)$ and in particular $\nu(j) = (\alpha_\omega)_{\omega \in V}$. The algorithm is given below.

Algorithm 2 Fast interpolation of polynomials represented in the LCH-basis over $\mathbb{F}_{p^r}$

Require: $\nu(0) = (P_V(x))_{x \in U+\mu_R}$, the set $U + \mu_R$ (where $U = U_i^j$) and $V = V_i^j$. The vector basis $(v_i)_i$ of $\mathbb{F}_{p^r}$.
Ensure: $\nu(j) = (\alpha_\omega)_{\omega \in V}$.
1: **for** k **from** 0 **to** $j-1$ **do**
2: **for** any $x \in U_{k+1}$ **do**
3: Compute $S_{x,k}$ as in (24).
4: **for** l from 0 **to** $p^k - 1$ **do**
5: Call "Standard fast interpolation" of [5, Ch. 10].
Input : $\nu(k, x, l)$, the set of evaluation points $S_{x,k}$.
Output: $f(X) = \sum_{d \in \mathbb{F}_p} \nu(k+1, x, l)_d \cdot X^d$.
6: Set $\nu(k+1) = \nu(k+1) \,||\, \nu(k+1, x, l)$.
7: **end for**
8: **end for**
9: Compute $\nu(k+1) = \mathsf{B}_{j-k-1}^{-1} \cdot \nu(k+1)$.
10: **end for**
11: **return** $\nu(j) = (\alpha_\omega)_{\omega \in V}$.

Remark. Step 5 would normally require to evaluate the formal derivative $M(X)'$ of $M(X) = \prod_{k \in \mathbb{F}_p} (X - S_{x,k})$. Noting that $S_{x,k}$ is an affine set, $M(X)$ is $T(X) - T(L_k(x))$ where $T(X) = X^p - L_k(v_{i+k})^{p-1} \cdot X$. Therefore $M'(X) = T'(X) = -L_k(v_{i+k})^{p-1}$ which is a constant. Consequently $M'(X)$ does not need to be evaluated at the different

points as it would be required normally.

Complexity. In what follows, $n = p^j$ represents the size of the data in terms of elements of $\mathbb{F}_{p^r}$. The cost of algorithm 2 is mainly given by its steps 3 and 5. As previously explained, the cost of step 3 is at most $O(j \cdot p^j) = O(\log_p n \cdot n)$. Step 5 consists of multiple calls to the subroutine Standard fast interpolation. This subroutine is in $O(\mathsf{M}(p) \log_2 p)$ and is called as many times as the subroutine of algorithm 1. Therefore step 5 and more globally algorithm 2 are in

$$O(n \cdot \log_2 n \cdot \log_2 p \cdot \log_2 \log_2 p)\,.$$

In-place algorithms. The multipoint evaluation (resp. interpolation) of a polynomial expressed in the LCH-basis is a combination of permutations and calls to the subroutine Standard fast multipoint evaluation (resp. Standard fast interpolation) on p points. Permutations are based on perfect shuffle permutations. As explained in [15], a perfect shuffle can be expressed as the composition of two involutions and can therefore be implemented by simply swapping elements, which are in-place operations. The subroutines of fast multipoint evaluation and interpolation on p points can also be performed in-place following [6].

6 EXPERIMENTAL RESULTS

We implemented and ran our algorithms on an Intel Core i5 CPU at 3, 2 Ghz. Our implementations are written in *C* using the FLINT library [7]. We compared the timings of our algorithm for fast multipoint evaluation of polynomials over $\mathbb{F}_{p^r}$ with the original method proposed in [12].

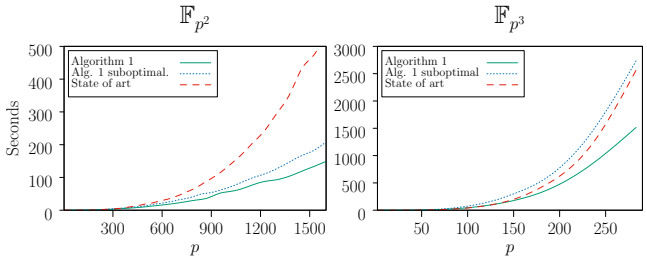


Figure 1: Timings for p^r-point evaluation over $\mathbb{F}_p^2$ (left) and $\mathbb{F}_p^3$ (right). Abscissa is characteristic p.

Algorithm 1 suboptimal consists in implementing formula (10) where the operands of the Kronecker product are commuted using a well-defined permutation. In this case, the sequence of Vandermonde matrices $V_k(x), x \in U_{k+1}$ is repeated p^k times. This lead to unnecessary memory transfers and/or recomputation of polynomials. Algorithm 1 based on formula (12) remedies this problem. In this case, each successive Vandermonde matrix is used p^k times in a row. Both algorithm 1 suboptimal and algorithm 1 were implemented following the classical (not in-place) version of the Standard fast multipoint evaluation and interpolation. In the state of the art [12], each single evaluation requires $O(p)$ operations while our approach performs p-point evaluations in $O\left(\mathsf{M}(p) \cdot \log_2(p)\right) = O\left(p \cdot \log_2^2 p \cdot \log_2 \log_2 p\right)$. This explains the improvement of total complexity for n-point evaluation of polynomials in the LCH-basis from $O(n \cdot \log_p n \cdot p)$ to $O(n \cdot \log_2 n \cdot \log_2 p \cdot \log_2 \log_2 p)$ as confirmed in Figure 1.

7 CONCLUSION

In this paper, we tackled the problems of fast multipoint evaluation and interpolation of polynomials represented in the LCH-basis over $\mathbb{F}_{p^r}$.

We provided a fast algorithm for the multipoint evaluation problem. We reduced such an evaluation to the problem of computing multiple multipoint evaluation of (standard) polynomials with respect to the monomial basis at p points over $\mathbb{F}_{p^r}$. By doing so, we optimized the complexity of the original method from $O(n \cdot \log_p n \cdot p)$ to $O(n \cdot \log_2 n \cdot \log_2 p \cdot \log_2 \log_2 p)$.

We also provided an algorithm for the fast interpolation problem which was left unsolved in [12] for finite fields of characteristic p. We reduced this problem to the one of computing multiple fast interpolation of (standard) polynomials with respect to the monomial basis at p points over $\mathbb{F}_{p^r}$. Our method is in $O(n \cdot \log_2 n \cdot \log_2 p \cdot \log_2 \log_2 p)$. We implemented both methods using the FLINT library and we showed that the improvement is confirmed in practice. Using permutations of the data, which are explicitely given, our algorithms satisfy high memory-locality. They can also be performed in-place.

ACKNOWLEDGMENTS

The authors would like to thank Robin Larrieu for giving us valuable information on some aspects of the FLINT library. They also wish to thank the reviewers of the paper for extremely detailed and fruitful comments and criticisms.

REFERENCES

[1] Elwyn R. Berlekamp. *Algebraic coding theory.* McGraw-Hill series in systems science. McGraw-Hill, 1968.
[2] Ming-Shing Chen, Chen-Mou Cheng, Po-Chun Kuo, Wen-Ding Li, and Bo-Yin Yang. Multiplying boolean polynomials with frobenius partitions in additive fast fourier transform. *CoRR*, abs/1803.11301, 2018.
[3] James W. Cooley and John W. Tukey. An algorithm for the machine calculation of complex Fourier series. *Math. Comput.*, 19:297–301, 1965.
[4] M. Davio. Kronecker products and shuffle algebra. *"IEEE Transactions on Computers"*, 30(2):116–125, 1981.
[5] Joachim Von Zur Gathen and Jurgen Gerhard. *Modern Computer Algebra.* Cambridge University Press, New York, NY, USA, 2 edition, 2003.
[6] Pascal Giorgi, Bruno Grenet, and Daniel S. Roche. Fast in-place algorithms for polynomial operations: division, evaluation, interpolation, 2020.
[7] W. Hart, F. Johansson, and S. Pancratz. FLINT: Fast Library for Number Theory, 2015. Version 2.5.2, http://flintlib.org.
[8] Leslie Hogben. *Handbook of linear algebra; 2nd ed.* Discrete Mathematics and Its Applications. Taylor and Francis, Hoboken, NJ, 2013.
[9] Runzhou Li, Qin Huang, and Zulin Wang. Encoding of non-binary quasi-cyclic codes by lin-chung-han transform. In *IEEE Information Theory Workshop, ITW 2018, Guangzhou, China, November 25-29, 2018*, pages 1–5, 2018.
[10] Sian-Jheng Lin, Tareq Y. Al-Naffouri, and Yunghsiang S. Han. Efficient frequency-domain decoding algorithms for reed-solomon codes. *CoRR*, abs/1503.05761, 2015.
[11] Sian-Jheng Lin, Tareq Y. Al-Naffouri, and Yunghsiang S. Han. FFT algorithm for binary extension finite fields and its application to reed-solomon codes. *IEEE Trans. Information Theory*, 62(10):5343–5358, 2016.
[12] Sian-Jheng Lin, Tareq Y. Al-Naffouri, Yunghsiang S. Han, and Wei-Ho Chung. Novel polynomial basis with fast fourier transform and its application to reed-solomon erasure codes. *IEEE Trans. Information Theory*, 62(11):6284–6299, 2016.
[13] Sian-Jheng Lin, Wei-Ho Chung, and Yunghsiang S. Han. Novel polynomial basis and its application to reed-solomon erasure codes. In *55th IEEE Annual Symposium on Foundations of Computer Science, FOCS 2014, Philadelphia, PA, USA, October 18-21, 2014*, pages 316–325, 2014.
[14] Joris van der Hoeven and Robin Larrieu. The frobenius FFT. In *Proceedings of the 2017 ACM on International Symposium on Symbolic and Algebraic Computation, ISSAC 2017, Kaiserslautern, Germany, July 25-28, 2017*, pages 437–444, 2017.
[15] Qingxuan Yang, John Ellis, Khalegh Mamakani, and Frank Ruskey. In-place permuting and perfect shuffling using involutions. *Information Processing Letters*, 113(10):386 – 391, 2013.

WhyMP, a Formally Verified Arbitrary-Precision Integer Library

Guillaume Melquiond
Université Paris-Saclay, CNRS, Inria, LRI
Orsay, France

Raphaël Rieu-Helft
TrustInSoft
Paris, France
Université Paris-Saclay, CNRS, Inria, LRI
Orsay, France

ABSTRACT

Arbitrary-precision integer libraries such as GMP are a critical building block of computer algebra systems. GMP provides state-of-the-art algorithms that are intricate enough to justify formal verification. In this paper, we present a C library that has been formally verified using the Why3 verification platform in about four person-years. This verification deals not only with safety, but with full functional correctness. It has been performed using a mixture of mechanically checked handwritten proofs and automated theorem proving. We have implemented and verified a nontrivial subset of GMP's algorithms, including their optimizations and intricacies. Our library provides the same interface as GMP and is almost as efficient for smaller inputs. We detail our verification methodology and the algorithms we have implemented, and include some benchmarks to compare our library with GMP.

CCS CONCEPTS

• **Mathematics of computing → Mathematical software**; • **Theory of computation → Hoare logic**; • **Computing methodologies → Exact arithmetic algorithms**.

ACM Reference Format:
Guillaume Melquiond and Raphaël Rieu-Helft. 2020. WhyMP, a Formally Verified Arbitrary-Precision Integer Library. In *International Symposium on Symbolic and Algebraic Computation (ISSAC '20), July 20–23, 2020, Kalamata, Greece.* ACM, New York, NY, USA, 8 pages. https://doi.org/10.1145/3373207.3404029

1 INTRODUCTION

The GNU Multi-Precision library, or GMP, is a widely used arbitrary-precision arithmetic library implemented in C and assembly. It provides state-of-the art algorithms for basic arithmetic operations and number-theoretic primitives. It is used in computer algebra software, as well as safety-critical contexts such as cryptography and security of Internet applications.

GMP is extensively tested, but some parts of the code are visited with very low probablity, such as $1/2^{64}$. This makes random testing a poor way of ensuring GMP's correctness. Moreover, most of the algorithms are quite intricate, so finding bugs through manual inspection of the code is challenging. As such, GMP has had its share of bugs.[1] We advocate using formal verification to ensure memory safety and the absence of correctness bugs for all inputs.

GMP features several layers, each one handling different kinds of numbers. The innermost one, `mpn`, handles natural numbers. The other three layers, `mpz`, `mpq`, and `mpf`, are mostly wrappers around `mpn` and handle relative numbers, rational numbers, and floating-point numbers respectively.

We have verified a subset of algorithms from the `mpn` and `mpz` layers of GMP using the Why3 verification platform using the following approach. We first implement the GMP algorithms in WhyML, the high-level specification and programming language that Why3 provides [7]. We also give them a formal specification based on GMP's documentation and our own understanding of the algorithms. Then, Why3 computes verification conditions that, once proved, guarantee that the WhyML functions are memory-safe and satisfy the specifications we provided. Using a collection of automated theorem provers, we check these verification conditions, thereby proving the functions correct. Finally, using Why3's extraction mechanism, we obtain an efficient and correct-by-construction C library that closely mirrors the original GMP code. We give more details on the verification process, guarantees, and caveats, in Section 2. The resulting C library, named WhyMP, can be found at

https://gitlab.inria.fr/why3/whymp/

WhyMP is not a full implementation of `mpn` and `mpz`. In particular, `mpn` contains many algorithms for each basic operation, so that the optimal one can be used, depending on the size of the inputs. We have implemented and verified at least one algorithm for each of addition, subtraction, multiplication, division, square root, modular exponentiation, and base conversion (I/O). In most cases, we have verified only the algorithm best suited to smaller numbers (typically up to 1,000 bits). The `mpz` wrapper is also a work in progress. Moreover, while our algorithms attempt to mirror GMP's implementation closely, there are a few differences. We provide a detailed list of functions and differences between WhyMP and GMP in Section 3.

While WhyMP does not fully implement GMP's API, it is compatible with GMP. Indeed, the functions have the same signatures and specifications. Therefore, in a C program that uses GMP, it is possible to substitute the calls to GMP for calls to the corresponding WhyMP functions. Moreover, WhyMP is roughly performance-competitive with versions of GMP that do not use handwritten assembly. This is easily explained by the fact that WhyMP closely mirrors GMP's code. Most of the performance difference comes from a small number of very short critical primitives. Therefore, it should be possible to check them carefully and add them to the trusted code base to recoup most of the performance loss. We

[1] Look for "division" at https://gmplib.org/gmp5.0.html

ISSAC '20, July 20–23, 2020, Kalamata, Greece

ACM ISBN 978-1-4503-7100-1/20/07…$15.00
https://doi.org/10.1145/3373207.3404029

present a more detailed benchmark of various configurations of GMP and WhyMP in Section 4.

2 VERIFICATION AND TCB

With software developed in a traditional way, users expect bugs to be plentiful. Only careful code reviews, safety analysis tools and other memory sanitizers, and lots of testing, both automated and manual, will ultimately ensure that software is bug-free with a high level of confidence.

Here, we follow a different approach: formal verification, and more specifically, deductive program verification. First, we mathematically specify what the library functions are supposed to compute. Then, using Why3, we turn both the code and its specification into a large logical formula. Finally, we look for a proof of this formula using automated theorem provers. If we succeed, it means that the code is safe and that it behaves as documented by the specification. Sections 2.1 and 2.2 give an overview of what the specification process entails.

Once formally verified, users should be able to assume that the library is bug-free, as its correctness has been proved and this proof has been mechanically checked. But as always, the devil resides in the details. So, Sections 2.3 and 2.4 carefully review all the hypotheses the correctness of our library depends on, in order to understand how large the user's leap of faith has to be.

2.1 Specifications

Adding a function to WhyMP starts with the conversion of the C code of a GMP function to WhyML. This manual process is mostly straightforward, as most C features used by GMP are mapped to our WhyML model of C. The next step is to provide a specification for the function. Let us consider the `mpn_copyi(r,x,n)` function as an example. It copies n limbs of an `mpn` number starting at the memory location pointed by x to a number starting at r.

The first two components of a specification are the *preconditions* and *postconditions*. Those are first-order formulas about the state of the program and the arguments of the function, as well as its result in the case of postconditions. Let us start with postconditions. They express the mathematical relation between the inputs and the outputs of a function. For `mpn_copyi`, there is an obvious postcondition: when the function returns, the limbs stored in memory in the range $[r; r + n)$ have the same value as the limbs that were stored in memory in the range $[x; x + n)$ when the function started.

This postcondition is not sufficient, though, as it does not state anything about other memory locations. An `mpn_copyi` implementation that would mess with all the other limbs in memory would satisfy the postcondition but would be meaningless. So, the function has a second postcondition, which states that the limbs stored outside the range $[x; x + n)$ have the same values they had at start.

Preconditions state sufficient conditions for a function to behave safely and accordingly to the postconditions. In the case of `mpn_copyi`, the precondition simply states that accessing the memory ranges $[x; x + n)$ and $[r; r + n)$ is safe.

With other verification tools, there would usually be a second precondition that states that the ranges $[x; x+n)$ and $[r; r+n)$ do not overlap. In the case of Why3, the type system ensures that x and r point into separate memory blocks. More precisely, Why3 statically rejects any program that can possibly perform a logical data race, that is, when there are two potentially aliased pointers and one at least is used for writing. So, when introducing new pointers, the user has to explicitly tell Why3 their relation to existing ones.

That is it for the part of specification dedicated to the functional correctness of a program. For verification purpose, we also need to provide *variants* and *invariants*. A variant is an element from a well-founded ordering, *e.g.*, a positive integer, whose decrease ensures that loops and recursive calls terminate. For instance, for divide-and-conquer algorithms, the decreasing value is generally the length of the integers passed as arguments. Note that, in general, variants do not tell anything about the time complexity of a function.

Unless a loop can be unrolled, there is generally no obvious way of turning its behavior into a first-order formula. That is why it is critical to annotate loops with invariants. In fact, for Why3, the body of a loop is a black box. The only thing the tool knows about the program state after a loop is that the loop exited and that its invariant holds. So, the invariant needs to be strong enough, so that the specification of the function can be verified, yet it needs to be *inductive* so that it is preserved by a loop iteration, and it needs to hold at loop start. So, finding proper invariants requires a deep understanding of why an algorithm works correctly.

We now have all the pieces to specify the WhyML code corresponding to `mpn_copyi`:

```
let wmpn_copyi (r x: ptr uint64) (n: int32): unit
  requires { valid x n /\ valid r n }
  ensures  { forall i. 0 <= i < n -> r[i] = x[i] }
  ensures  { forall i. i<0\/n<=i -> r[i]=old r[i] }
= let ref i = 0 in
  while (Int32.(<) i n) do
    variant { n - i }
    invariant {forall j. 0 <= j < i -> r[j] = x[j]}
    invariant {forall j. j<0\/i<=j -> r[j]=old r[j]}
    r[i] <- x[i];
    i <- i + 1;
  done
```

2.2 Proof effort

From the code and its specification, Why3 computes a verification condition, which is a first-order formula. If the code is safe and if its specification, including invariants, is adequate, the verification condition holds. So, in an ideal world, we would just have to submit it to an automated theorem prover and it would answer that it found a proof.

While this is true for a function as simple as `mpn_copyi`, automated theorem provers either give up or time out, in practice. In particular, whenever a verification condition contains non-linear terms, provers tend to get lost in their search for a proof. In that case, the user needs to annotate the WhyML code with assertions. This makes for a larger verification condition, but it can now be split into lots of smaller ones, which hopefully are more agreeable to automated provers. For example, a property as simple as $\forall x, y, z \in \mathbb{Z},\ y > 0 \Rightarrow (x \cdot y + z) \div y = x + z \div y$ might require about 10 user assertions before automated theorem provers succeed.

Unfortunately, verification conditions related to the correctness of GMP-like libraries are full of non-linearity. In fact, even the representation of an integer from its limbs is already awfully non-linear: $a = \sum_i a_i \beta^i$. As a consequence, a lot of time is spent annotating the

code with extra assertions. To alleviate this issue, we even implemented and formally verified our own decision procedure dedicated to proving these kinds of arithmetic facts [10].

Once all the verification conditions are proved, we use Why3 to extract a C library from the WhyML code [15].

2.3 Guarantees and assumptions

The functional correctness of the library states the adequacy between the code of the library and its specification. Thus, if the pre- and postconditions are incorrect, even if the code has been formally verified, it might still be unfit for any meaningful usage. So, let us see how bad this can get.

First, let us consider postconditions. The second postcondition of `wmpn_copyi` states that the function only touches some specific parts of memory. It might seem easy to forget this postcondition, but in practice, it hardly happens. Indeed, if it was missing or too weak, it would be impossible to verify any nontrivial code that calls this function, so this kind of mistake gets detected early.

As for the first postcondition of `wmpn_copyi`, it states what the function actually performs. Unfortunately, it is a bit too simple and not quite representative of most functions of a GMP-like library. So, let us have a look at the WhyML signature of `mpz_add` as well as its first postcondition. This function takes three `mpz` numbers and stores the sum of the last two into the first one.

```
let wmpz_add (w u v: mpz_ptr): unit
  ensures { value_of w mpz =
        old (value_of u mpz + value_of v mpz) }
```

In the postcondition above, `mpz` is a global variable that keeps track of all the `mpz` numbers in memory. This variable is *ghost*, *i.e.*, only visible from the specification; it has no existence in the code of the function and is erased from the generated code [6]. The expression `(value_of` x `mpz)` designates the mathematical integer represented in memory by some `mpz` number x. The plus operator is the mathematical integer addition and it has no computational content. Thus, the postcondition states that, when the function returns, the integer represented by w is the sum of the two integers that were represented by u and v at the entry of the function. So, there is no difficulty for the user to trust that the function actually performs an addition. The situation is similar for all the functions of the library, as they can always be described by a simple relation between mathematical integers represented by inputs and outputs.

For preconditions, the situation is a bit more subtle. They have to be as weak as possible, as the callers of a function have to make sure that the program state satisfies its preconditions before calling it. That is true of all calls inside WhyMP, but once the library has been turned into C functions, nothing prevents the user from calling them with bad arguments or in an inconsistent state. As the functions are not programmed in a defensive fashion, they might then fail in unpredictable ways.

For `wmpn_copyi`, the precondition states that the input pointers are valid for accessing a large enough zone. Similarly, for `wmpz_add`, there is a validity precondition on the `mpz` numbers used as inputs and outputs. There is nothing surprising about these preconditions; GMP functions have the same requirements.

As mentioned earlier, pointer-manipulating functions also have some implicit preconditions about the aliasing of pointers. In particular, the `wmpn_copyi` function cannot be called if the ranges $[r; r+n)$ and $[x; x+n)$ overlap. Yet, GMP documentation of `mpn_copyi` implicitly states that overlap is allowed, as long as $r < x$.[2] That does not mean that `wmpn_copyi` will behave badly when passed aliased pointers; it just means that the formal verification does not cover this case. So, the function, as presented above, is not a perfect replacement for `mpn_copyi`.[3]

The header file `wmp.h` states all the preconditions that are not present in GMP's documentation. They are all related to aliased pointers, so only functions from the `mpn` layer are impacted. Since `mpz` numbers abstract the notion of pointer away, the WhyMP functions from the `mpz` layer do not suffer from these formal deficiencies. They have no hidden preconditions; they all behave in accordance with GMP's documentation.

The last point regarding correctness is termination. While the specifications do not tell anything about the time or space complexity of the functions, they have something to say about their termination. Indeed, by default, Why3 implicitly enforces *total* correctness, that is, assuming that the preconditions hold, the function terminates and the outputs satisfy the postconditions. In the case of our library, all the functions have thus been formally proved to return, but under the assumptions that the program does not run out of resources. More precisely, given valid inputs, functions from our library either return correct results, or they abruptly terminate the program because of a heap overflow, or they signal a stack overflow, *e.g.*, by a segmentation fault, as would GMP.

2.4 Trusted code base

As we have seen, the user has to understand the mathematical specification of the library (which is not much different from understanding its documentation), but at no point does the user need to understand the code in any way. Yet, to have confidence in the library, the user still needs to trust several other components.

First, given the WhyML code of the library as well as its specification, Why3 generates a set of verification conditions by a calculus of weakest precondition [9]. So, the user has to trust that Why3 has performed this computation correctly, that is, if all the verification conditions hold, then the specification adequately describes the behavior of the library. Why3 is a generic verification platform that has been used in numerous occasions, and the calculus of weakest precondition is a well-known approach to program verification, so this is not the component the user should worry about.

The verification conditions produced by Why3 are first-order formulas, which are often too complicated to be proved by a human. So, Why3 dispatches them to automated theorem provers, either SMT solvers or superposition-based provers [4]. The user needs to trust that the verification conditions were properly converted to the input languages of the provers and that the provers did not succeed in proving an incorrect theorem. For our library, we use the SMT solvers Alt-Ergo, CVC3, CVC4, and Z3, and the superposition-based E prover. All these theorem provers are off-the-shelf tools

[2] The "i" in `mpn_copyi` actually means "increasing".

[3] Contrarily to the simplified version presented in this paper, the full specification of `wmpn_copyi` in WhyMP permits aliased pointers, just as GMP.

that are widely used. Unfortunately, that does not make them bug-free, so the usual approach to increase the confidence in a WhyML development is to consider a verification condition to be proved only if several provers agree on it.

At this point, the user should be confident that the library is correct. But this is still WhyML code; it has to be converted to C code. Why3 is responsible for the extraction to C and the user needs to trust that none of the meaningful properties of the WhyML code were lost in the translation to C. There are three parts to it. The first one is the translation itself: WhyML constructs should be translated to C constructs. To increase the confidence, we have kept this translation as simple as possible. In particular, we did not try to convert any high-level feature of WhyML, such as automatic memory management or higher-order functions. As a consequence, the translation from WhyML to C is mostly syntactic [15].

The second part is the model of the C language we have implemented in WhyML. For example, we have defined an abstract type `ptr` to represent C pointers, as well as some abstract functions to read and write the pointed location. These functions have specifications that require the pointers to be valid and ensure that the memory is consistent, *e.g.*, reading a valid memory location after writing to it gives back the written value. All these WhyML functions are then mapped to C functions or operators. So, the user needs to trust that our specifications of these functions properly model the semantics of the C language. We had to improve the memory model, as its original version [15] was not expressive enough to support aliasing, which is critical for some functions.

The third part is composed of the arithmetic primitives used by our library. Indeed, WhyMP heavily relies on the availability of a multiplication of one limb by one limb returning two limbs, and conversely, of a division of two limbs by one limb. As with the memory model, these primitives are defined as abstract functions in WhyML and are mapped to handwritten C functions. Fortunately, most of those functions are trivial, as we rely on 128-bit support from C compilers. For example, here is the primitive for division:

```
uint64_t div64_2by1
    (uint64_t ul, uint64_t uh, uint64_t d)
{ return (((uint128_t)uh << 64) | ul) / d; }
```

The most complicated primitive is the one used to compute an 8-bit approximation of the square root for any integer between 128 and 511. This is implemented as a plain array of integer literals, but due to a technical limitation of Why3, we cannot express this in WhyML. So, we have performed the verification outside Why3 [11].

At this point, we have a C library that satisfies a meaningful specification. The last step is to compile and link it. So, the user also needs to trust that the compiler will not perform an incorrect optimization that would mess with the C code. This is quite a leap of faith, but no larger than the one needed when compiling any C library out there.

3 VERIFIED ALGORITHMS

Each WhyMP function comes at a significant cost in terms of time and proof effort, so only a subset of GMP's functions have been implemented. Moreover, while we strive to mirror GMP's algorithms as closely as possible, some differences remain. Some of these differences are due to time constraints, others come from technical limitations of the Why3 platform. Finally, some of GMP functions are specialized according to whether the hardware provides some non-standard primitives natively. In these cases, we only considered the "generic" version of the algorithm, that is, the one where no particular primitive is expected to be provided by the hardware. Let us review WhyMP's algorithms and the differences between GMP and WhyMP. More details on the algorithms themselves can be found in previous work [14].

3.1 Addition, subtraction

The algorithms for addition and subtraction in GMP are the schoolbook ones, and they are reproduced identically in WhyMP. However, almost all `mpn` addition and subtraction functions allow parameter overlap, which results in an unfortunately large amount of almost-identical variants of the addition and subtraction algorithms in WhyMP. Since these functions are very commonly used, we have tweaked the memory model in order to be able to prove generic versions of these functions that allow overlapping parameters. In the end, the exported versions of addition and subtraction can be called by external users in the same way as their GMP counterparts. However, they cannot always be called internally, due to limitations in Why3's type system, so the library still has a large amount of addition and subtraction variants for internal use.

3.2 Multiplication

GMP features more than ten different multiplication algorithms, which are each called when they are optimal depending on the sizes of the operands. These algorithms can be split into three categories: the schoolbook algorithm, suited for smaller numbers, Toom-Cook variants, and finally Schönhage-Strassen multiplication, which is only used on very large numbers (about 500,000 bits on x86_64). GMP's Toom-Cook variants have names of the form `toom_`xy, with $x \geq y \geq 2$. They start by splitting their larger operand into x parts and the smaller into y parts. All these parts need to have roughly equal length, so the ratio between the lengths of the operands should be roughly x to y. In addition, the asymptotic complexity decreases when y grows, so for larger numbers, variants that split operands into more parts should be called.

For a specific range of number sizes, GMP performs multiplication by calling `toom_22` and `toom_32` depending on the relative sizes of the operands. We have implemented and verified these two functions. Thus, WhyMP's multiplication is comparable with GMP's one until the inputs reach the threshold where GMP starts using `toom_33`. For very unbalanced operands, GMP provides a wrapper that first splits the larger one into many smaller segments, and then calls `toom_42` on each one. As we have not verified `toom_42`, the wrapper for multiplication is changed slightly to call `toom_32` instead. As will be seen in the benchmarks, this has little impact on performances.

Finally, GMP features a specialized function for squaring integers faster than with the general multiplication. It is not yet implemented in WhyMP.

3.3 Division

There are two main division algorithms in GMP: a so-called schoolbook algorithm, and a (subquadratic) divide-and-conquer algorithm.

The schoolbook algorithm is far from trivial. For example, each candidate quotient digit is computed using a 3-limb by 2-limb division algorithm [12], using a precomputed pseudo-inverse of the top two limbs of the divisor). This 3-by-2 division is more costly than the usual 2-by-1 division but greatly reduces the number of subsequent adjustment steps. WhyMP implements this algorithm faithfully. However, GMP's schoolbook algorithm uses an entirely different algorithm when the length of the denominator is more than half that of the numerator. The goal is for the complexity to depend only on the size of the quotient. WhyMP does not implement this second algorithm. The performance disparity becomes significant when the denominator is very close to the numerator in length. Moreover, WhyMP does not implement divide-and-conquer division.

3.4 Square root

GMP implements a divide-and-conquer square root, with a very intricate base case that uses precomputed 8-bit approximations and performs only two Newton iterations and a fast adjustment to compute the square root of a 64-bit number. WhyMP implements the exact same square-root algorithms as GMP [11]. However, the complexity of the square root is dominated by that of the long division. Therefore, WhyMP's square root is quadratic (like the division) whereas GMP's is subquadratic thanks to the divide-and-conquer division. The absence of a dedicated squaring function is also felt somewhat.

3.5 Modular exponentiation

GMP features a modular exponentiation algorithm that implements the sliding-window method and uses Montgomery reduction so that only one division is needed in the whole computation. We have implemented and verified the same algorithm in WhyMP. Once again, the main performance difference comes from the algorithm's dependencies. Indeed, modular exponentiation involves a lot of squaring, so the lack of a dedicated squaring function hurts WhyMP's performance.

GMP also features a variant of the modular exponentiation algorithm that is designed to be side-channel secure. More precisely, its control flow and memory accesses do not depend on the values of the operands. We have also verified this function, however it relies on a side-channel secure division, whose verification is still a work in progress. As a result, WhyMP's side-channel resistant modular exponentiation is not usable yet. Moreover, the formal verification of this function only offers guarantees on its correctness. We currently do not have any good way to prove that it is indeed side-channel resistant.

3.6 Base conversions and I/O

GMP's I/O functions include algorithms that translate a large number into a string that represents it in an arbitrary base (between 2 and 62) and vice versa. This algorithm is surprisingly intricate, in particular when converting from/to base 10. Due to time constraints, we have chosen to instead verify the conversion algorithms from Mini-GMP, the standalone version of GMP that is distributed alongside it. These algorithms are simpler, and we expect that I/O is usually not the bottleneck in computations on large numbers.

3.7 The `mpz` layer

The `mpz` layer is a wrapper around `mpn` that takes care of number signs and storage. Most users of GMP interact only with this layer, so that they do not have to manually manage memory allocations. Functions of `mpz` typically do not perform any computation themselves. Instead, they call the corresponding `mpn` function and handle the various cases required depending on the signs and lengths of the operands.

However, for each arithmetic operation, there can be several `mpz` functions that each handle various cases, even though they all rely on the same `mpn` function. For example, there are about twenty division functions in GMP, so that the best one can be used depending on whether the quotient, the remainder or both are needed, whether the divisor is a machine integer or a large integer, and the rounding mode. While this is the most extreme example, the `mpz` layer does represent a large amount of work. In the end, WhyMP's `mpz` layer is still largely a work in progress.

3.8 Compatibility concerns

The signatures of the WhyML functions of our library are such that the generated C functions have the exact same signature as GMP functions. Numbers are also represented the same way in memory, that is, `mpn` numbers are pointers to an array of limbs stored from least significant to most significant, while `mpz` numbers are a record whose third field is an `mpn` number. Thus, one can easily pass a number from one library to a function of the other library.

There are two potential sources of incompatibility, as our library lacks a bit of genericity. First, it only works with 64-bit limbs, so it cannot be interfaced with a 32-bit GMP. If the user code only uses `mpz` numbers and does not need to mix both libraries, this incompatibility does not matter. Second, our library does not support the custom memory handler of GMP. In particular, it performs its allocations using `malloc`. Thus, `mpz` numbers from one library will wreak havoc when passed to the other library and freed, unless GMP's default memory handler is used.

4 BENCHMARKS

The next sections show how GMP and WhyMP compare on three benchmarks: multiplication, square root, and a primality test. More precisely, three variants of GMP and three variants of WhyMP are tested. Indeed, the direct comparison between GMP and WhyMP is not that meaningful, as GMP relies on native assembly routines. So, in addition to the timings of WhyMP and GMP, four other timings are measured to give a better view of the performances.

First, GMP is also compiled without support for assembly, which means that only the generic C code is compiled. GMP without assembly and WhyMP are not exactly in the same ballpark though, since they do not use the same primitive operations for doing a $64 \times 64 \rightarrow 128$ multiplication and a 128-by-64 division. Indeed, in assembly-free GMP, these are implemented in C using only 64-bit operations, while WhyMP delegates these operations to the 128-bit support of the C compiler.

Second, to measure the impact of these two primitives, WhyMP is also compiled in a way such that their 128-bit implementation is replaced by the 64-bit one from GMP without assembly.

Third, the timings of Mini-GMP are measured. Mini-GMP is a C library "*intended for applications which need arithmetic on numbers larger than a machine word, but which don't need to handle very large numbers very efficiently.*" It is distributed along GMP. It uses the same kind of implementation as GMP without assembly for the two primitives above, that is, it uses only 64-bit operations.

Finally, some low-level `mpn` functions of WhyMP are replaced by their respective GMP counterparts, as these functions are typically written in assembly. Those functions are `add_n` (resp. `sub_n`), which computes the sum (resp. difference) of equally-sized `mpn` numbers; `add` and `sub`, for `mpn` numbers with different sizes; `mul_1`, which multiplies an `mpn` number by a single limb; `addmul_1` (resp. `addmul_2`), which multiplies an `mpn` number by a single limb (resp. a two-limb number), and then accumulates the product into the destination; and `submul_1`, which accumulates the opposite of the product. Note that we could have replaced a lot more functions of WhyMP by their assembly counterparts from GMP, including rather complicated ones, *e.g.*, division by two-limb numbers. Instead, we chose to focus on a few simple functions, so as to not blow the trusted code base out of proportions, which would defeat the point of formally verifying an arithmetic library.

The version of GMP is 6.1.2. The benchmarks are executed on an Intel Xeon E5-2450 at 2.50 GHz. All the libraries are compiled using GCC 8.3.0 using the options selected by GMP, *i.e.*, "`-O2 -march=sandybridge -mtune=sand... -fomit-frame-pointer`".

Figure 1 shows the timings obtained on the various benchmarks. On every figure, abscissas are the number of 64-bit limbs, while ordinates are the time in microseconds. All the figures are in log-log scale, so that the asymptotic complexity is apparent. Performance-wise, the general ordering of the plots is the same on every figure: GMP is the fastest, then comes WhyMP with GMP's assembly primitives, then WhyMP, then GMP without assembly support, then WhyMP without 128-bit support, and Mini-GMP is the slowest.

4.1 Multiplication

The first benchmark simply tests multiplication for various sizes of `mpn` numbers, so as to exercise both the base-case multiplication as well as Toom-Cook algorithms. Two cases are tested: equal-sized inputs, and $n \times 24n$ unbalanced inputs.

The unbalanced case tests the algorithmic differences between WhyMP and GMP. Indeed, WhyMP performs 16 calls to `toom_32`, which results in 64 $\frac{n}{2} \times \frac{n}{2}$ multiplications, while GMP performs 12 calls to `toom_42`, which results in 60 $\frac{n}{2} \times \frac{n}{2}$ multiplications. Due to the extra cost of interpolation for `toom_42`, WhyMP hardly suffers from not having `toom_42` at this level of unbalance.

Comparing the plots of Mini-GMP, WhyMP without 128-bit support, and GMP without assembly, makes it apparent when the libraries switch to different algorithms. Mini-GMP sticks with the quadratic schoolbook algorithm, while WhyMP and GMP switch to `toom_22` around $n = 30$, and then GMP switches to `toom_33` around $n = 60$. Starting around $n = 170$ (`toom_44` for GMP), the lack of higher variants of Toom-Cook in WhyMP becomes noticeable, as the library becomes progressively slower with respect to GMP. For $n \leq 170$, WhyMP is at most twice as slow as GMP, and when replacing the primitive operations with the assembly ones from GMP, the slowdown does not exceed 20%. The smaller n is, the smaller the slowdown, down to about 5% for $n \leq 20$.

4.2 Square root

The second benchmark tests the square root for various sizes of `mpn` numbers. GMP's algorithm performs a long division, so WhyMP greatly suffers from featuring only the schoolbook division, despite using the same divide-and-conquer square-root algorithm as GMP. This makes WhyMP with assembly about 50% slower than GMP for $n \leq 600$. Without assembly, WhyMP is twice as slow for $n \leq 90$, and thrice as slow for $n \leq 600$. As for Mini-GMP, its poor performance (up to ×150 times slower for $n \leq 600$) can be explained by the use of a converging sequence $y_{n+1} = (x/y_n + y_n)/2$, rather than a dedicated algorithm.

4.3 Miller-Rabin's primality test

The third benchmark implements Miller-Rabin's primality test for number sizes commonly encountered in cryptography applications. This is a simple implementation inspired from GMP's one. It exercises the `mpz` layer as well as the modular exponentiation. Note that the modular exponentiation used in WhyMP is just a wrapper over `mpn_powm`, so it supports neither even modulos nor negative exponents, contrarily to `mpz_powm`. WhyMP is 110% slower than GMP for $n \leq 28$, and 140% slower for $n \leq 60$. With assembly primitives, the slowdown is less than 30% for $n \leq 60$.

4.4 Evaluation

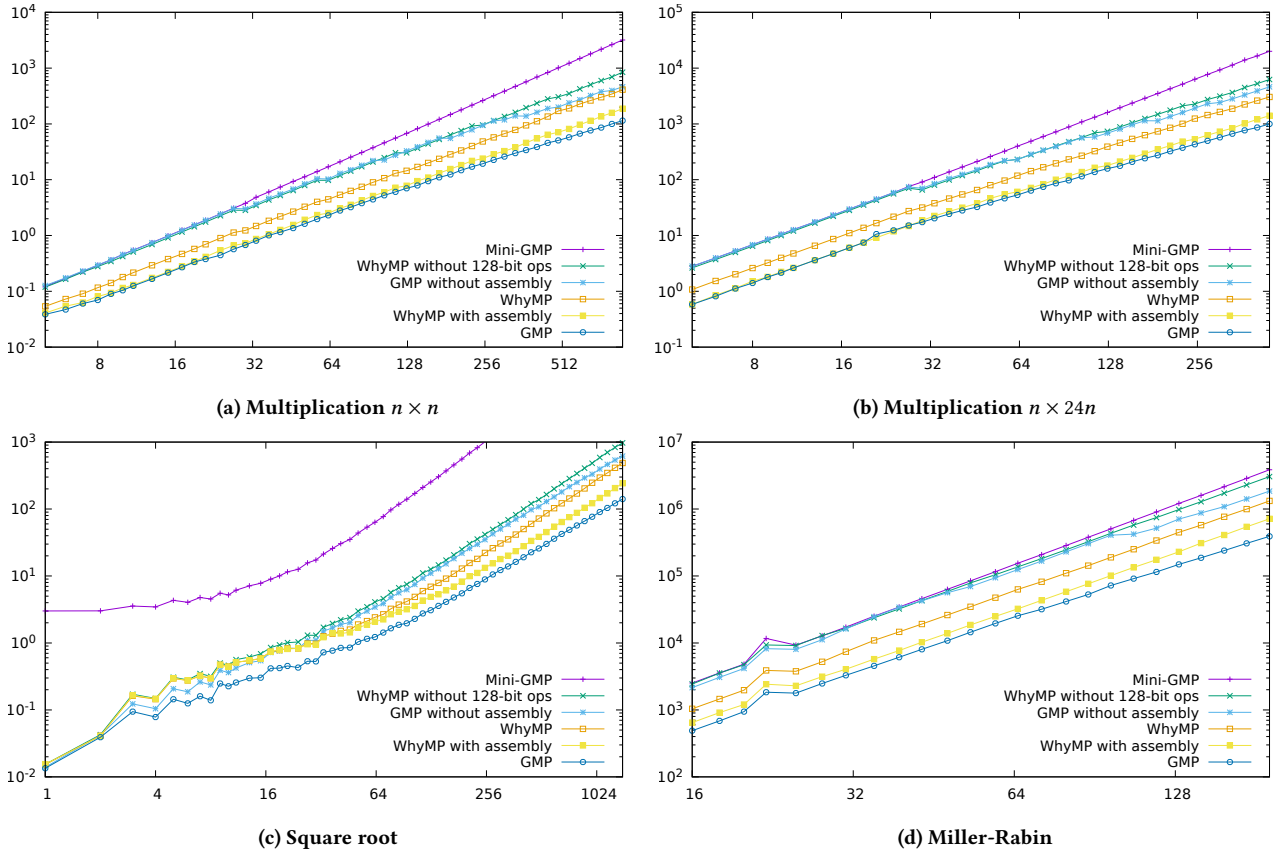
Overall, two factors have a large impact on performance: the complexity of the algorithms, and the quality of the underlying arithmetic primitives. On large numbers, WhyMP's multiplication and division falls behind even that of the assembly-free version of GMP when the latter switches to a more efficient algorithm. In all other cases, the algorithms are similar enough that the primitives seem to be the deciding factor. What we conclude from this is that WhyMP's algorithms are close enough to the original that most of the performance difference comes from the primitives written in handwritten assembly, at least for smaller inputs.

5 RELATED WORK

We have used the Why3 tool [4, 5, 7] to develop a formally verified arbitrary-precision integer arithmetic library that closely mirrors GMP. We obtain a verified and efficient C library. Previous work generally does not deal with a large number of highly optimized algorithms. As far as we know, this work is the first formally verified arbitrary-precision integer library that has comparable performance to the state of the art. Let us discuss a few examples of existing verifications of arithmetic libraries.

Bertot *et al.* verified GMP's square root general case algorithm [3] using Coq. Our Why3 proof of that algorithm is directly inspired from their article. Their formalization is rather similar to ours, but their proof effort is even larger, as Why3 proofs are partially automated in a way Coq proofs are not.

Myreen and Curello verified an arbitrary-precision integer arithmetic library [13] using the HOL4 theorem prover. Their work covers the four basic arithmetic operations, but not the square root or modular exponentiation. They did not attempt to produce highly



(a) Multiplication $n \times n$

(b) Multiplication $n \times 24n$

(c) Square root

(d) Miller-Rabin

Figure 1: Timings for multiplication, square root, and Miller-Rabin.

efficient code. However, their verification goes all the way down to x86 machine code, using formally verified compilers and decompilers. They also manage to automate most of the proofs involving pointer reasoning, despite using an interactive tool.

Affeldt used Coq to verify a binary extended GCD algorithm implemented in a variant of MIPS assembly [1], as well as the basic arithmetic functions the algorithm depends on. The work uses GMP's number representation and a memory model based on separation logic. The author verifies an implementation of the algorithm in a pseudo-code language and proves the fact that the pseudocode correctly simulates the MIPS assembly code.

Fischer verified a modular exponentiation library [8] using Isabelle/HOL and a framework for verifying imperative programs developed by Schirmer [16]. The library is not meant to be efficient. For example, it represents arbitrary-precision integers as garbage-collected doubly-linked lists of machine integers. The author reports running into issues inside the tool due to the large number of invariants and conditions needed to keep track of aliasing. This is exactly the kind of issue that we avoid by forcing function parameters separation, at the cost of some expressivity (see the discussion on `mpn_copyi` in Section 2.3).

Berghofer used Isabelle/HOL to develop a verified bignum library programmed in the Spark fragment of the Ada programming language [2]. The library provides modular exponentiation as well as the primitives required to implement it. The modular exponentiation algorithm is a simple square-and-multiply one, without the sliding-window optimization or the Montgomery reduction that are featured in GMP and WhyMP. However, the proof effort (2,000 lines of Isabelle written over three weeks) is surprisingly low.

Schoolderman used Why3 to verify hand-optimised Karatsuba multiplication assembly routines for the AVR architecture [17]. The algorithms are not arbitrary-precision, instead there are many routines, each specialized for a particular operand size up to 96×96 bits. This allows the loops to be unrolled, so the algorithms are branch-free and the proofs are much easier for SMT solvers.

Finally, Zinzindohoué *et al.* developed a formally verified cryptography library written in F* and extracted to C [18]. It implements the full NaCl API, and includes a bignum library. The extracted code is as fast as state-of-the-art C implementations, and part of it is now deployed in the Mozilla Firefox web browser. Their approach is very similar to ours in that it consists in verifying the algorithms in a high-level language suited for verification, and then compiling them to C. The integers have a small, fixed size that depends on the choice of elliptic curve. Again, the fact that the number sizes are known makes the problem much easier for automated solvers. As a result, their proof enjoys a higher degree of automation than ours. Thus, while their specifications are similar or larger in length, their code requires much fewer annotations than ours.

6 CONCLUSION

WhyMP is an arbitrary-precision integer library. It has been developed, specified, and annotated, using the WhyML language. Its correctness has been formally verified using Why3 and several external theorem provers, mostly SMT ones. The formal verification includes the functional correctness, *i.e.*, the relations between function inputs and outputs match the definition of the corresponding mathematical operators. The memory representation of integers is identical to GMP's, and the functions have the same signatures, which makes WhyMP a potential substitute to GMP.

The compatibility with GMP is not limited to the interface. The library also implements the vast majority of the state-of-the-art tricks found in GMP, as it was implemented after a detailed analysis of GMP's code. This makes it competitive with GMP in some specific cases. For instance, WhyMP is much faster than the pure C variant of GMP. Yet, GMP implements numerous finely tuned assembly routines, which makes WhyMP twice as slow as the standard GMP. By selecting a few multiply-and-accumulate primitives, WhyMP can be brought closer, making it only 5% to 20% slower than GMP, depending on the operation.

In the process of verifying WhyMP, we found one bug in the comparison function of GMP[4] that occurs for very large inputs (several gigabytes). This is exactly the sort of bug that is easy to find using formal methods, but hard to test against effectively. Our work also influenced the development of GMP in another way. Our correctness proof for the divide-and-conquer multiplication ended up being so intricate (much more than what GMP's developers thought) that they preferred to modify the code, so that its correctness became more obvious.

This does not mean that GMP is now formally verified, although our work increases further the (already high) confidence in its correctness. To the best of our knowledge, such macro-heavy C code mixed with assembly is completely out of reach of any existing verification framework, due to the combinatorial explosion that arises from all the possible architectures and compilation options. If one really wanted to tackle a formal verification at the level of the C code, Mini-GMP would make a much more sensible target. Still, it would require a large proof development on the mathematical side, though smaller than ours, as Mini-GMP's algorithms are much simpler than GMP's and ours.

During this work, the main obstacle was due to automatic solvers. While the resulting verification process can be said to be automatic, it is only so because the WhyML code was heavily annotated, to the point where it can be seen as a pen-a-paper proof of algorithms. Nonetheless, this is a machine-checked proof. Some related works were much more successful in actually performing an automatic verification. But it was only for functions on fixed-size integers, as their loops are fully unrollable during verification. This is certainly not the case of GMP.

As a consequence of the constant fight to get the external solvers to automatically prove WhyMP's correctness, formally verifying functions currently consumes too much time. This explains why our library provides few variants of the functions yet, despite a proof effort of four person-years. Still, we intend to add at least a divide-and-conquer division, so that WhyMP can tackle slightly larger numbers, not only during division, but also square root and modular exponentiation. WhyMP also needs some side-channel resistant functions, so that modular exponentiation can be used in security-sensitive cryptography applications. Finally, we should investigate how to verify assembly code for some mainstream instruction sets, so as to close the performance gap with GMP.

[4] https://gmplib.org/list-archives/gmp-bugs/2020-February/004733.html

REFERENCES

[1] Reynald Affeldt. 2013. On Construction of a Library of Formally Verified Low-level Arithmetic Functions. *Innovations in Systems and Software Engineering* 9, 2 (2013), 59–77. https://doi.org/10.1007/s11334-013-0195-x

[2] Stefan Berghofer. 2012. Verification of Dependable Software using SPARK and Isabelle. In *6th International Workshop on Systems Software Verification (OpenAccess Series in Informatics (OASIcs), Vol. 24)*. Dagstuhl, Germany, 15–31. https://doi.org/10.4230/OASIcs.SSV.2011.15

[3] Yves Bertot, Nicolas Magaud, and Paul Zimmermann. 2002. A Proof of GMP Square Root. *Journal of Automated Reasoning* 29, 3-4 (2002), 225–252. https://doi.org/10.1023/A:1021987403425

[4] François Bobot, Jean-Christophe Filliâtre, Claude Marché, and Andrei Paskevich. 2011. Why3: Shepherd Your Herd of Provers. In *Boogie 2011: First International Workshop on Intermediate Verification Languages*. Wrocław, Poland, 53–64. https://hal.inria.fr/hal-00790310.

[5] Jean-Christophe Filliâtre. 2013. One Logic To Use Them All. In *24th International Conference on Automated Deduction (Lecture Notes in Artificial Intelligence, Vol. 7898)*. Lake Placid, USA, 1–20.

[6] Jean-Christophe Filliâtre, Léon Gondelman, and Andrei Paskevich. 2016. The Spirit of Ghost Code. *Formal Methods in System Design* 48, 3 (2016), 152–174. https://doi.org/10.1007/s10703-016-0243-x

[7] Jean-Christophe Filliâtre and Andrei Paskevich. 2013. Why3 — Where Programs Meet Provers. In *22nd European Symposium on Programming (Lecture Notes in Computer Science, Vol. 7792)*. Heidelberg, Germany, 125–128.

[8] Sabine Fischer. 2008. Formal Verification of a Big Integer Library. In *DATE Workshop on Dependable Software Systems*. http://www-wjp.cs.uni-sb.de/publikationen/Fi08DATE.pdf

[9] Robert W. Floyd. 1993. Assigning Meanings to Programs. In *Program Verification*. Springer, 65–81.

[10] Guillaume Melquiond and Raphaël Rieu-Helft. 2018. A Why3 Framework for Reflection Proofs and its Application to GMP's Algorithms. In *9th International Joint Conference on Automated Reasoning (Lecture Notes in Computer Science, Vol. 10900)*. Oxford, United Kingdom, 178–193. https://doi.org/10.1007/978-3-319-94205-6_13

[11] Guillaume Melquiond and Raphaël Rieu-Helft. 2019. Formal Verification of a State-of-the-Art Integer Square Root. In *IEEE 26th Symposium on Computer Arithmetic*. Kyoto, Japan. https://hal.inria.fr/hal-02092970

[12] Niels Moller and Torbjörn Granlund. 2011. Improved Division by Invariant Integers. *IEEE Trans. Comput.* 60, 2 (2011), 165–175. https://doi.org/10.1109/TC.2010.143

[13] Magnus O. Myreen and Gregorio Curello. 2013. Proof Pearl: A Verified Bignum Implementation in x86-64 Machine Code. In *3rd International Conference on Certified Programs and Proofs (Lecture Notes in Computer Science, Vol. 8307)*. Melbourne, Australia, 66–81. https://doi.org/10.1007/978-3-319-03545-1_5

[14] Raphaël Rieu-Helft. 2019. A Why3 Proof of GMP Algorithms. *Journal of Formalized Reasoning* 12, 1 (2019), 53–97. https://doi.org/10.6092/issn.1972-5787/9730

[15] Raphaël Rieu-Helft, Claude Marché, and Guillaume Melquiond. 2017. How to Get an Efficient yet Verified Arbitrary-Precision Integer Library. In *9th Working Conference on Verified Software: Theories, Tools, and Experiments (Lecture Notes in Computer Science, Vol. 10712)*. Heidelberg, Germany, 84–101. https://doi.org/10.1007/978-3-319-72308-2_6

[16] Norbert Schirmer. 2005. A Verification Environment for Sequential Imperative Programs in Isabelle/HOL. In *International Conference on Logic for Programming Artificial Intelligence and Reasoning*. 398–414.

[17] Marc Schoolderman. 2017. Verifying Branch-Free Assembly Code in Why3. In *Working Conference on Verified Software: Theories, Tools, and Experiments*. 66–83.

[18] Jean Karim Zinzindohoué, Karthikeyan Bhargavan, Jonathan Protzenko, and Benjamin Beurdouche. 2017. HACL*: A Verified Modern Cryptographic Library. Cryptology ePrint Archive, Report 2017/536. https://eprint.iacr.org/2017/536.

On Parameterized Complexity of the Word Search Problem in the Baumslag–Gersten Group

Alexei Miasnikov*
Andrey Nikolaev*
amiasnik@stevens.edu
anikolae@stevens.edu
Stevens Institute of Technology
Hoboken, NJ, USA

ABSTRACT

We consider the word search problem in the Baumslag–Gersten group GB. We show that the parameterized complexity of this problem, where the area of van Kampen diagram serves as a parameter, is polynomial in the length of the input and the parameter. This contrasts the well-known result that the Dehn function and the time complexity of the word search problem in GB are non-elementary.

CCS CONCEPTS

• **Theory of computation** → **Fixed parameter tractability**; *Complexity classes*; • **Mathematics of computing** → *Discrete mathematics.*

KEYWORDS

parameterized complexity, fixed parameter tractability, word problem, word search problem, Baumslag–Solitar group, Baumslag–Gersten group, Dehn function

ACM Reference Format:
Alexei Miasnikov and Andrey Nikolaev. 2020. On Parameterized Complexity of the Word Search Problem in the Baumslag–Gersten Group. In *International Symposium on Symbolic and Algebraic Computation (ISSAC '20), July 20–23, 2020, Kalamata, Greece.* ACM, New York, NY, USA, 4 pages. https://doi.org/10.1145/3373207.3404042

1 INTRODUCTION AND PRELIMINARIES

1.1 Word problem in groups and parameterized complexity

The word problem in groups was introduced by Dehn around 1910 and has been a subject of primary interest in group theory since. A classical result of Novikov [10] (and later, independently, Boone [2]) is that there exist finitely presented groups with algorithmically undecidable word problem. There are examples of groups where the word problem is decidable but **NP**-complete [12].

Parameterized complexity was introduced by Downey and Fellows [5] in 1999. They proposed to study the complexity of algorithmic problems in terms of the size of input and parameters of input or output. This allows a finer classification of hard problems. For example, the famously **NP**-complete vertex cover problem [7] on a graph with n vertices can be solved in time $O(1.2738^k + kn)$ if k is the number of vertices in the cover [3]. Problems solvable in time $f(k)n^c$, where f is a computable function, are called fixed parameter tractable. The class of such problems is denoted FPT. Another classical **NP**-complete problem, dominating set problem [7], is known to be not fixed parameter tractable, unless parameterized complexity classes FPT and W[2] coincide [5], which is generally thought unlikely.

The word search problem in a group G asks to represent a given word w, provided it is equal to 1 in G, as a product of conjugates of given defining relators or, equivalently, in geometric terms, to find a van Kampen diagram representing w (see Section 1.2 for details). If the group G is finitely (or recursively) presented then the search word problem in G is decidable, so the real question is about the algorithmic complexity of the problem. In theoretical computer science this problem could be properly framed in terms of proof complexity: what is the time complexity to find a "proof" that a given word w is equal to 1 in G? Intuitively, the complexity of this problem may come from two sources: the whole combinatorial complexity of the corresponding van Kampen diagram; or the sheer size of the answer (the area of the diagram). Note that the area of van Kampen diagrams in groups is described by the famous Dehn functions, which are by now quite well-studied in finitely presented groups. On the other hand, not much is known about the complexity of van Kampen diagrams as combinatorial objects. Furthermore, the complexity of the word search problem in G is not necessary related to the complexity of the word problem itself. Indeed, the word problem in Baumslag–Solitar group $BS(1,2)$ is decidable in polynomial time (the group is linear), but the Dehn function of $BS(1,2)$ is exponential (see Section 1.3). More dramatically, the word problem in the Baumslag–Gersten group GB is decidable, surprisingly, in polynomial time [9], even though the Dehn function in this group is non-elementary, i.e., not bounded by any fixed iterate of the exponential function [6]. Therefore, the complexity of the word search problem in GB is non-elementary.

In this paper we introduce a natural version of the parameterized complexity of the word search problem in groups and use it to identify the source of complexity of this problem, as mentioned above. Namely, we study the parameterized complexity of the word search problem in the Baumslag–Gersten group GB, where the area k of a

*Both authors contributed equally to this research.

ISSAC '20, July 20–23, 2020, Kalamata, Greece

ACM ISBN 978-1-4503-7100-1/20/07…$15.00
https://doi.org/10.1145/3373207.3404042

van Kampen diagram serves as the parameter. We show that in this case the parameterized complexity of the word search problem in GB is polynomial of low degree in both length of the input and the parameter k. In other words, we show that the word search problem for GB is fixed parameter tractable with a polynomial function $f(k)$. In this sense, the complexity of the word search problem in GB comes purely from the size of the answer.

In conclusion, we would like to note that the main purpose of this paper is to introduce the parameterized complexity to algorithmic and geometric group theory and to show that it sheds some light on important aspects of the word problem in groups. In particular, the result above demonstrates that, firstly, while the complexity of the word search problem in a finitely presented (one-relator) group GB is immense, it comes purely from the size of the answer, and nothing else. Secondly, this result rehabilitates, in some sense, the classical decision algorithm for the word problem in HNN-extensions, which amounts to applying Britton's lemma repeatedly. The algorithm is simple and geometrically very transparent, though sometimes inefficient. At least in the case of the group GB, it is clear now that this algorithm is as efficient as a word search problem algorithm can be. It seems probable that, after proper reformulation, the result may hold in arbitrary HNN-extensions. Thirdly, there is no any reason to believe that the methods of [9] (the so-called power circuits) could be applied to arbitrary one-relator groups to show that the word problem in such groups is decidable in polynomial time. However, the classical Britton's algorithm is still applicable there, which gives a way to approach the parameterized complexity of the word search problem in these groups.

1.2 Word problem, van Kampen diagrams, area as a parameter, asphericity

Given a group presentation $G = \langle X \mid R \rangle$, we can consider the *word problem* for this presentation: given a word w in $X \cup X^{-1}$, establish whether $w = 1$ in G. (Words in $X \cup X^{-1}$ are called *group words* in X; since we do not consider any other kind, throughout this paper we simply call them *words* in X.) Equivalently, the word problem asks whether there are words $u_1, \dots, u_k$ in X and $r_1, \dots, r_k \in R \cup R^{-1}$ such that

$$u_1 r_1 u_1^{-1} \cdot u_2 r_2 u_2^{-1} \cdots u_k r_k u_k^{-1} = w \tag{1}$$

in the free group on X; that is, the two words are equal up to free cancellation. The word length $|w| = n$ (number of letters in w) serves as the length of input for this problem. Equation (1) is equivalent to existence of a so-called *van Kampen diagram* (or *disc diagram*): a planar graph with edges labeled by letters from $X \cup X^{-1}$, each face labeled by a cyclic permutation of an element of $R \cup R^{-1}$, and such that its boundary reads the word w (up to free cancellation). For example, the van Kampen diagram in Figure 1 (left) shows that in the abelian group $\langle a, b \mid a^{-1}b^{-1}ab \rangle$, the word $a^{-3}b^{-1}abab^{-2}ab^2$ is equal to 1. Indeed, from the diagram we can observe that

$$a^{-3}b^{-1}abab^{-2}ab^2 = a^{-2}(a^{-1}b^{-1}ab)a^2 \cdot (a^{-1}b^{-1}ab) \cdot b^{-1}(a^{-1}b^{-1}ab)b$$

in the free group, which resolves this instance of the word problem positively in the sense of equation (1). Conversely, given the above equality, we can recover the corresponding van Kampen diagram by drawing a bouquet like the one in Figure 1 (right) and folding edges with the same labels.

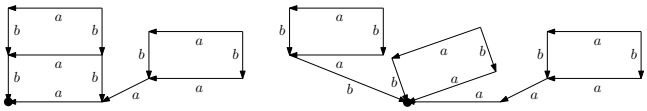


Figure 1: From the marked vertex, the boundary reads $a^{-1}a^{-1}a^{-1}b^{-1}abab^{-1}b^{-1}abb = a^{-3}b^{-1}abab^{-2}ab^2$.

By the *word search problem* we mean the following: given a word w in X and the information that $w = 1$ in G, find a corresponding expression (1). Observe that the parameter k in (1) is the number of faces in the corresponding van Kampen diagram. The latter is usually called the *area* of a van Kampen diagram. Since each edge in a reduced van Kampen diagram is a boundary edge or a face edge, there is a quadratic (in terms of $n = |w|$ and k) bound on the total length of expression in the left hand side of (1). With that in mind, the area k can serve as a parameter for the size of answer in the word search problem.

The function $f(n) = \max\{k \mid w = 1, |w| \le n\}$ is called the Dehn function of a presentation. Its growth class does not depend on a particular choice of presentation for a given group. In this sense, the Dehn function is defined for a group, and it tells how large an area is required to establish that a word of length n is equal to 1 in the group.

1.2.1 Asphericity. Generally, a word w representing identity in a given group presentation $\langle X \mid R \rangle$ may possess substantially different expressions (1). However, in such event, we can glue the two corresponding disc diagrams along their common boundary w, and after free cancellation we can obtain a spherical diagram, see Figure 2. Group presentations that do not admit non-trivial spherical diagrams are called *aspherical*. This notion is discussed in [8, Sections III.10, 11]. In [11], aspherical group presentations were studied in explicit combinatorial terms. Note that, strictly speaking, there are different notions of asphericity, defined in different terms (topological, cohomological, combinatorial). They are equivalent to each other under additional mild assumptions, which hold in the cases considered in this paper. In particular, the notion of asphericity we use in this paper is usually called *combinatorial asphericity*. To avoid excessive exhibition here, we refer the reader to [8, Section III.11] and [11, Section 2.2] for further details.

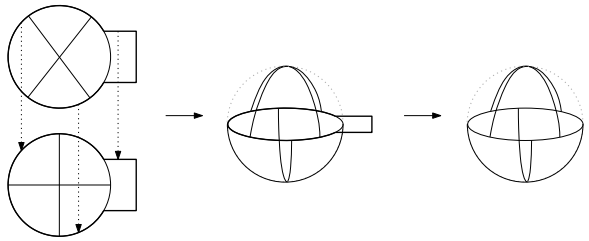


Figure 2: Producing a spherical diagram from two disc diagrams for the same word.

Our reasoning in Section 2 relies on uniqueness (up to free cancellation) of disc diagrams for certain group presentations, i.e., on asphericity of those groups presentations. Specifically, we use the following special case of [11, Lemma 2.5] (which was first stated in [4] in different terms).

Lemma 1.1 ([11]). *Let $\mathcal{P}_1 = \langle X \mid R\rangle$ be a presentation of a group G and let $\mathcal{P}_2 = \langle X, t \mid R, t^{-1}utv^{-1}\rangle$, where u, v are words in X, be the standard presentation of an HNN-extension of G. Suppose that $t^{-1}utv^{-1}$ is not conjugate to any relator in $R \cup R^{-1}$, and that u, v have infinite order in $\mathcal{P}_1$. Then $\mathcal{P}_2$ is aspherical if $\mathcal{P}_1$ is aspherical.*

1.3 Baumslag–Solitar groups, Baumslag–Gersten group *GB*

Baumslag–Solitar groups are the groups $BS(m, n)$ with presentation $\langle a, b \mid (a^m)^b = a^n\rangle$. A group $BS(m, n)$ can be viewed as an HNN-extension of $\langle a\rangle$ with stable letter b and associated subgroups $\langle a^m\rangle$, $\langle a^n\rangle$. These groups are important in combinatorial group theory for their role as a showcase of a variety of geometric and combinatorial properties of groups. One such remarkable feature is that while an algorithm to solve the word problem is simple (it is given by Britton's lemma for HNN-extensions), the Dehn function of this group is exponential if $|m| \neq |n|$. For example, this can be observed in $BS(1, 2)$ by considering equalities of the form

$$b^{-N}ab^N \cdot a = a \cdot b^{-N}ab^N, \quad N = 1, 2, \ldots.$$

The Baumslag–Gersten group (also known as Gilbert Baumslag group) is a group with presentation

$$\langle a, t \mid a^{a^t} = a^2\rangle.$$

For our purposes it is more convenient to rewrite this as

$$GB = \langle a, b, t \mid a^b = a^2, a^t = b\rangle,$$

which shows that GB is an HNN-extension of the Baumslag–Solitar group $BS(1, 2)$ with stable letter t and associated subgroups $\langle a\rangle$, $\langle b\rangle$. This group was first introduced by Baumslag [1]. Later Gersten showed in [6] that the Dehn function of GB is non-elementary, i.e., grows faster than any fixed iterate of the exponential function. However, Myasnikov, Ushakov, and Won recently showed in [9] that the word problem in GB is polynomial time decidable through use of the so-called power circuits. In Section 2 we show that the word search problem in this group is polynomial in n, k, where n is the length of input and k is the area of a van Kampen diagram.

2 VAN KAMPEN DIAGRAMS IN BAUMSLAG–SOLITAR AND BAUMSLAG–GERSTEN GROUPS

Lemma 2.1. *Parameterized complexity of the word search problem in Baumslag–Solitar group $BS(1, 2)$ is polynomial in n and k, where n is the size of input and k is the area of a van Kampen diagram.*

Proof. Consider Baumslag–Solitar group $BS(1, 2) = \langle a, b \mid a^b = a^2\rangle$, which can be viewed as an HNN-extension of the cyclic group $\langle a\rangle$. We recall that by Britton's lemma if a freely reduced word w in a, b represents 1 in $BS(1, 2)$, then it has a subword $b^{-1}a^m b$ or a subword $ba^{2m}b^{-1}$. As an iterative step in the solution of the word search problem, we can replace the former with a^{2m} and the latter with a^m, thus reducing the number of occurrences of b, b^{-1} in the word. Since by Lemma 1.1 $BS(1, 2)$ is aspherical, it follows that this rewriting can be read in a minimal van Kampen diagram, see Figure 3. (Asphericity of $BS(1, 2)$ also follows since this group is one-relator, see, for example, [8, Proposition III.11.1].)

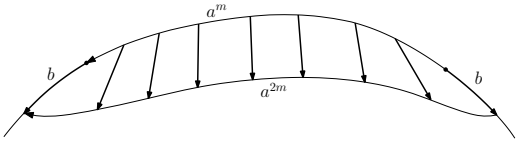


Figure 3: $b^{-1}a^m b = a^{2m}$.

Suppose a word w in a, b of length n is given as the input of the word search problem, with parameter k. We will rewrite the word $w = w_0$ into $w_1, w_2, \ldots$, so that every w_i has length bounded by a linear function of n, k; the number of rewriting steps will be bounded by n.

Given a word w_i, we search for a subword of the form $b^{-1}a^m b$ or $ba^{2m}b^{-1}$ and replace it with a^{2m} or a^m, respectively, thus obtaining w_{i+1} after free cancellation. As discussed above, if the word w represents 1 in $BS(1, 2)$, then the word w_{i+1} can be read along a boundary of a subdiagram in the original van Kampen diagram D_0 for w, and therefore its length after free cancellation does not exceed twice the number of edges in the diagram, at most

$$2(n + 5k) = 2n + 10k = m_0.$$

Indeed, the number of edges is not more than (perimeter)+(maximal length of relator)·(area of D_0), which is $n + 5k$. Note that the length of the rewritten word before the free cancellation does not exceed $2m_0$, since the length of w_i is bounded by the same m_0.

The search takes at most m_0^2 steps, the replacement at most $3m_0$ steps, including free cancellation. The number of steps does not exceed n, since every rewriting eliminates a pair of occurrences of b, b^{-1}. Therefore, overall number of steps until an empty word is reached or the procedure fails does not exceed

$$n \cdot ((2n + 10k)^2 + 3(2n + 10k)).$$

As we compute each word w_{i+1} from w_i, we can record the relation applied at each step, and the corresponding location in the word w_i. This allows us to recover the van Kampen diagram once an empty word is reached in polynomial time $P(n, k)$. □

As we mentioned in Section 1.3, the Baumslag–Gersten group

$$GB = \langle a, b, t \mid a^b = a^2, a^t = b\rangle$$

is an HNN-extension of the Baumslag–Solitar group $BS(1, 2) = \langle a, b \mid a^b = a^2\rangle$ by an isomorphism $\varphi : \langle a\rangle \to \langle b\rangle$, $\varphi(a) = b$. In Lemma 2.1, we have shown that there is a polynomial upper bound $P(n, k)$ on the parameterized complexity of the word search problem in $BS(1, 2)$, where n denotes the size of the input word and k the maximum area of a van Kampen diagram. We exploit this fact and the reasoning similar to the one we used in its proof to deal with the word search problem in the Baumslag–Gersten group.

Theorem 2.2. *Parameterized complexity of the word search problem in the Baumslag–Gersten group GB is polynomial in n and k, where n is the size of input and k is the area of a van Kampen diagram.*

Proof. Given a word w in a, b, t of length n and a parameter k, we consider subwords of the form

$$t^{-1}u(a,b)t \quad \text{or} \quad tu(a,b)t^{-1}.$$

We search for such occurrences where, respectively, $u(a,b) \in \langle a \rangle$ or $u(a,b) \in \langle b \rangle$. Start with the former. Suppose $u(a,b) = a^m$. By Lemma 1.1 it follows that the considered presentation of GB is aspherical. Then by Britton's lemma the word $u(a,b)a^{-m}$ can be read along a boundary of a subdiagram of the van Kampen diagram for w of area at most k, so $|m|$ cannot exceed twice the number of edges in the diagram:

$$|m| \leq 2(n + 5k) = m_0.$$

Therefore, establishing whether $u(a,b) \in \langle a \rangle$ reduces to solving $2m_0 + 1$ word problems $u(a,b) = a^m$, $-m_0 \leq m \leq m_0$, in $BS(1,2)$. By Lemma 2.1, this can be done in a time that does not exceed

$$\sum_{|m| \leq m_0} P(|u(a,b)| + m, k) = R(|u(a,b)|, m_0, k).$$

Notice that $R(|u(a,b)|, m_0, k)$ is a polynomial in $|u(a,b)|, m_0, k$.

Further observe that, under the condition $u(a,b) = a^m$, after rewriting $t^{-1}u(a,b)t = b^m$ the resulting (after free cancellation) word can still be read in the original van Kampen diagram (an example is shown in Figure 4), so iterating this process will never produce a word longer than m_0. Therefore, $|u(a,b)| \leq m_0$, so the function $R(|u(a,b)|, m_0, k)$ is bounded by a polynomial $R(m_0, m_0, k) = Q(n,k)$.

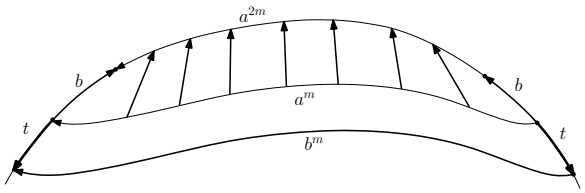


Figure 4: $ba^{2m}b^{-1} = a^m$, $t^{-1}a^m t = b^m$.

The reasoning for the latter case, $u(a,b) \in \langle b \rangle$, is done similarly.

The described rewriting process can iterate at most n times (the number of occurrences of t and t^{-1}). After that we arrive at a word w' in a, b. We use the algorithm provided by Lemma 2.1 to solve the word search problem for w' in $BS(1,2)$. After that, the van Kampen diagram for w can be recovered the same way as in Lemma 2.1. □

ACKNOWLEDGMENTS

The work is supported by Mathematical Center in Akademgorodok.

REFERENCES

[1] G. Baumslag. 1969. A non-cyclic one-relator group all of whose finite quotients are cyclic: To Bernhard Hermann Neumann on his 60th birthday. *Journal of the Australian Mathematical Society* 10, 3–4 (1969), 497–498. https://doi.org/10.1017/S1446788700007783

[2] W.W. Boone. 1958. The word problem. *Proceedings of the National Academy of Sciences of the United States of America* 44, 10 (1958), 1061.

[3] J. Chen, I. A Kanj, and G. Xia. 2006. Improved parameterized upper bounds for vertex cover. In *International symposium on mathematical foundations of computer science*. Springer, 238–249.

[4] I.M. Chiswell, D.J. Collins, and J. Huebschmann. 1981. Aspherical group presentations. *Mathematische Zeitschrift* 178, 1 (1981), 1–36.

[5] R.G. Downey and M.R. Fellows. 2012. *Parameterized complexity*. Springer Science & Business Media.

[6] S.M. Gersten. 1992. *Dehn Functions and l1-norms of Finite Presentations*. Springer New York, New York, NY, 195–224. https://doi.org/10.1007/978-1-4613-9730-4_9

[7] R.M. Karp. 1972. Reducibility among combinatorial problems. In *Complexity of computer computations*. Springer, 85–103.

[8] R. Lyndon and P. Schupp. 2001. *Combinatorial Group Theory*. Springer.

[9] A. Myasnikov, A. Ushakov, and D.W. Won. 2011. The Word Problem in the Baumslag group with a non-elementary Dehn function is polynomial time decidable. *Journal of Algebra* 345, 1 (2011), 324–342. https://doi.org/10.1016/j.jalgebra.2011.07.024

[10] P.S. Novikov. 1955. On algoritmic unsolvability of the word problem in the theory of groups (in Russian). *Tr. Mat. Inst. Akad. Nauk SSSR, Izd. Akad. Nauk SSSR* 44 (1955).

[11] M. Sapir. 2014. A Higman embedding preserving asphericity. *Journal of the American Mathematical Society* 27, 1 (2014), 1–42.

[12] M.V. Sapir, J.-C. Birget, and E. Rips. 2002. Isoperimetric and isodiametric functions of groups. *Ann. Math.* 156, 2 (2002), 345–466.

On the Chordality of Ordinary Differential Triangular Decomposition in Top-down Style

Chenqi Mou
LMIB – School of Mathematical Sciences /
Beijing Advanced Innovation Center for Big Data and Brain Computing,
Beihang University, Beijing 100191, China
chenqi.mou@buaa.edu.cn

ABSTRACT

In this paper we extend existing theoretical results on chordal graphs in algebraic triangular decomposition in top-down style to the ordinary differential case. We first propose the concept of differential associated graph of an ordinary differential polynomial set, and then for two typical algorithms in top-down style for ordinary differential triangular decomposition based on the pseudo-division and subresultant regular subchain respectively, we prove that when the input differential polynomial set has a chordal differential associated graph G and one perfect elimination ordering of G is used, the differential associated graph of any polynomial set in the decomposition process by these two algorithms is a subgraph of G.

CCS CONCEPTS

• **Computing methodologies** → **Symbolic and algebraic manipulation**; • **Theory of computation** → *Design and analysis of algorithms.*

KEYWORDS

Differential triangular decomposition, chordal graph, top-down style, pseudo-division

ACM Reference Format:
Chenqi Mou. 2020. On the Chordality of Ordinary Differential Triangular Decomposition in Top-down Style. In *International Symposium on Symbolic and Algebraic Computation (ISSAC '20), July 20–23, 2020, Kalamata, Greece.* ACM, New York, NY, USA, 8 pages. https://doi.org/10.1145/3373207.3403999

1 INTRODUCTION

Differential algebra, founded by Ritt [23, 24] and developed by Kolchin [16] and many others, is the subject to study differential polynomial systems from an algebraic viewpoint. Since its advent, the development of differential algebra has been along with computation and algorithms, like computation of characteristic sets for prime differential ideals [24], algorithmic elimination theory for differential polynomial systems [27], and the coherence property needed for the computation with partial differential polynomial systems [26]. Combined with methods from computer algebra, effective algorithmic methods in differential algebra have been studied, improved, and implemented [1–4, 10, 13, 15, 17, 32, 37] to solve various problems related to differential polynomials, e.g., solving systems of ordinary differential equations [32], identifying the observability of control systems [7, 8], and verification of linearizability for ordinary differential equations [18], etc.

In his seminal book [24], Ritt proposed the concept of characteristic sets for differential polynomial ideals. Characteristic sets are a typical kind of differential triangular sets. In the ordinary differential case, differential triangular sets are ordered sets of ordinary differential polynomials whose greatest differential indeterminates strictly increase. The process to decompose any differential polynomial set into finitely many differential triangular sets or systems with associated zero or ideal relationships is called differential triangular decomposition, and there exist effective methods to perform differential triangular decomposition [1–3, 15, 17, 32]. The readers are referred to the tutorial [14] for more details on this subject.

In this paper we are mainly interested in algorithms in top-down style for ordinary differential triangular decomposition. The top-down strategy in triangular decomposition means that the variables (in the algebraic case) or the differential indeterminates (in the ordinary differential case) appearing in the input polynomial set are handled in a strictly decreasing order, and it has been used to design various algorithms for algebraic and differential triangular decomposition [5, 9, 17, 31–34].

The tool we use in this paper to study and analyze ordinary differential triangular decomposition in top-down style is the chordal graph, also called triangulated graphs, see, e.g.,[12, Chap. 4]. Chordal graphs have very special structures and thus have been applied in many scientific and engineering areas like optimization [28], in particular, to design algorithms for sparse Gaussian elimination [11, 22, 25] and sparse sums-of-squares decomposition [29, 30, 36]. The connections between chordal graphs and triangular decomposition was first revealed in [6], where the new concept of chordal networks was proposed with an effective algorithm for computing them based on triangular decomposition. Inspired by their works, the author of this paper and his collaborators proved that several algorithms in top-down style for triangular decomposition preserve the chordality of the input polynomial set and proposed sparse algorithms for triangular decomposition making use of the perfect elimination orderings of chordal graphs based on these newly proved theoretical results [19–21].

This paper is an attempt to extend the existing analyses on the chordality of algebraic triangular decomposition in top-down style

ISSAC '20, July 20–23, 2020, Kalamata, Greece

ACM ISBN 978-1-4503-7100-1/20/07…$15.00
https://doi.org/10.1145/3373207.3403999

to differential triangular decomposition, and obviously the ordinary differential case is our first target. In this paper we propose the concept of differential associated graph of an ordinary differential polynomial set and clarify its differences from the (algebraic) associated graph. Then we prove that when the input polynomial set is chordal, two algorithms in top-down style for ordinary differential triangular decomposition based on the pseudo-division [2, 32] and subresultant regular subchain [15, 35] preserve the chordality, successfully extending the theoretical results for algebraic triangular decomposition to the ordinary differential case. Like in the algebraic case [20], sparse algorithms for ordinary differential triangular decomposition may also be proposed with more focused study and experiments based on the results above, yet they fall out of the scopes of this paper.

This paper is organized in the following way. After recalling basic notions and notations for differential algebra and differential triangular decomposition in Section 2, we associate any differential polynomial set with a graph based on its differential indeterminates in Section 3. Then in Sections 4 and 5 respectively, we reformulate two typical algorithms in top-down style for differential triangular decomposition based on the pseudo-division and subresultant regular subchain, and then prove that these two algorithms preserve the chordality of the input differential polynomial sets if the perfect elimination orderings are used. We conclude this paper with remarks on the underlying problems for extending the results in this paper to partial differential triangular decomposition in Section 6.

2 PRELIMINARIES

2.1 Ordinary differential polynomial ring

Let $\mathbb{K}$ be an ordinary differential field of characteristic 0 with the derivation $\delta = \frac{\mathrm{d}}{\mathrm{d}t}$ and $x_1, \ldots, x_n$ be differential indeterminates ordered as $x_1 < \cdots < x_n$. Denote the set $\{x_1, \ldots, x_n\}$ by $\boldsymbol{x}$ and the derivatives $\delta^j x_i = \frac{\mathrm{d}^j x_i}{\mathrm{d}t^j}$ by x_{ij} for all $i = 1, \ldots, n$ and $j \in \mathbb{Z}_{\geq 0}$ (with $x_{i0} := x_i$). Then the ordinary differential polynomial ring $\mathbb{K}\{\boldsymbol{x}\}$ is the polynomial ring over $\mathbb{K}$ in the infinitely many variables $\{x_{ij} | i = 1, \ldots, n, j \in \mathbb{Z}_{\geq 0}\}$ equipped with the derivation δ. In this paper we only study the ordinary differential case, and thus the word "ordinary" is usually omitted for simplicity. Let $\mathcal{F}$ be a differential polynomial set in $\mathbb{K}\{\boldsymbol{x}\}$. Then the *differential ideal generated by* $\mathcal{F}$, denoted by $[\mathcal{F}]$, is the ideal generated by the polynomials in $\mathcal{F}$ and all their derivatives.

We fix a differential ordering $<_d$ on all the derivatives as

$$\begin{aligned} & x_1 <_d x_{11} <_d x_{12} <_d \cdots \\ <_d\ & x_2 <_d x_{21} <_d x_{22} <_d \cdots \\ & \vdots \\ <_d\ & x_n <_d x_{n1} <_d x_{n2} <_d \cdots . \end{aligned}$$

This ordering is a typical elimination one [14]. For a differential polynomial $F \in \mathbb{K}\{\boldsymbol{x}\}$, its *lead* $\mathrm{ld}(F)$ is the derivative which effectively appears in F and is of the highest rank with respect to (short as w.r.t. hereafter) $<_d$, and its *order* $\mathrm{ord}(F)$ is the index of the differential indeterminate which effectively appears in F and is greatest w.r.t. the indeterminate order $<$ (if F involves none of $x_1, \ldots, x_n$, then $\mathrm{ord}(F) := 0$). Viewed as a univariate polynomial in its lead $\mathrm{ld}(F)$, the polynomial F can be rewritten as

$$F = F_d \,\mathrm{ld}(F)^d + F_{d-1} \,\mathrm{ld}(F)^{d-1} + \cdots + F_1 \,\mathrm{ld}(F) + F_0,$$

where $\mathrm{ld}(F_i) <_d \mathrm{ld}(F)$ for all $i = 0, \ldots, d$ and $F_d \neq 0$. The polynomials F_d and $F - F_d \,\mathrm{ld}(F)^d$ in the above formulae are called the *initial* and *tail* of F respectively, denoted by $\mathrm{ini}(F)$ and $\mathrm{tail}(F)$. The formal derivative of F w.r.t. $\mathrm{ld}(F)$ is called the *separant* of F and denoted by $\mathrm{sep}(F)$, that is

$$\mathrm{sep}(F) = \frac{\partial F}{\partial\, \mathrm{ld}(F)} = d F_d \,\mathrm{ld}(F)^{d-1} + (d-1) F_{d-1} \,\mathrm{ld}(F)^{d-2} + \cdots + F_1.$$

Let $F, G \in \mathbb{K}\{\boldsymbol{x}\}$ be two differential polynomials. Then F is said to be *partially reduced* w.r.t. G if no proper derivative of $\mathrm{ld}(G)$ appears in F and *reduced* w.r.t. G if it is partially reduced w.r.t. G and $\deg(F, \mathrm{ld}(G)) < \deg(G, \mathrm{ld}(G))$. We say F is of lower rank than G if $\mathrm{ld}(F) <_d \mathrm{ld}(G)$, or $\mathrm{ld}(F) = \mathrm{ld}(G)$ but $\deg(F, \mathrm{ld}(F)) < \deg(G, \mathrm{ld}(G))$. For any finite set of differential polynomials, we can always find a differential polynomial with a minimal rank.

2.2 Differential triangular decomposition

Definition 2.1. A finite ordered set $\mathcal{T} = [T_1, \ldots, T_r]$ of differential polynomials in $\mathbb{K}\{\boldsymbol{x}\}$ is called a *weak differential triangular set* if $0 < \mathrm{ord}(T_1) < \cdots < \mathrm{ord}(T_r)$. Furthermore, if for each $i = 1, \ldots, r$, T_i is partially reduced w.r.t. T_j $(j \neq i, 1 \leq j \leq r)$, then $\mathcal{T}$ is called a *differential triangular set.*

The definition of a weak differential triangular set $\mathcal{T}$ in a general (ordinary or partial) differential ring requires that for any two distinct polynomials $T_i, T_j \in \mathcal{T}$, we have that $\mathrm{ld}(T_i)$ is not a derivative of $\mathrm{ld}(T_j)$. Since in the ordinary differential ring $\mathbb{K}\{\boldsymbol{x}\}$ there is only one derivation, this definition is equivalent to the one for weak differential triangular sets in Definition 2.1 above. Though the variables of differential polynomials in $\mathbb{K}\{\boldsymbol{x}\}$ are the derivatives $\{x_{ij} | i = 1, \ldots, n, j \in \mathbb{Z}_{\geq 0}\}$, ordinary differential triangular sets are defined w.r.t. the differential indeterminates $x_1, \ldots, x_n$.

Let $\mathcal{P}$ and $\mathcal{Q}$ be two differential polynomial sets in $\mathbb{K}\{\boldsymbol{x}\}$, and $\tilde{\mathbb{K}}$ be some differential extension of the differential field $\mathbb{K}$. We denote the set of common zeros of the differential polynomials in $\mathcal{P}$ which are not zeros of any polynomial in $\mathcal{Q}$ by

$$\mathrm{Z}_{\tilde{\mathbb{K}}}(\mathcal{P}/\mathcal{Q}) := \{\overline{\boldsymbol{x}} \in \tilde{\mathbb{K}}^n | P(\overline{\boldsymbol{x}}) = 0, Q(\overline{\boldsymbol{x}}) \neq 0, \forall P \in \mathcal{P}, Q \in \mathcal{Q}\}.$$

We write $\mathrm{Z}_{\tilde{\mathbb{K}}}(\mathcal{P}/\mathcal{Q})$ as $\mathrm{Z}(\mathcal{P}/\mathcal{Q})$ when $\tilde{\mathbb{K}}$ is not explicitly specified, $\mathrm{Z}(\{P\}/\mathcal{Q})$ as $\mathrm{Z}(P/\mathcal{Q})$, and $\mathrm{Z}(\mathcal{T}/\emptyset)$ as $\mathrm{Z}(\mathcal{T})$ respectively.

Definition 2.2. An ordered set $(\mathcal{T}, \mathcal{U})$ with $\mathcal{T}$ a weak differential triangular set and $\mathcal{U}$ a differential polynomial set (possibly an empty one) in $\mathbb{K}\{\boldsymbol{x}\}$ is called a *weak differential triangular system* if for each $T \in \mathcal{T}$, we have

$$\mathrm{Z}(\mathrm{ini}(T)) \cap \mathrm{Z}(\mathcal{T}/\mathcal{U}) = \emptyset \quad \text{and} \quad \mathrm{Z}(\mathrm{sep}(T)) \cap \mathrm{Z}(\mathcal{T}/\mathcal{U}) = \emptyset.$$

Furthermore, $(\mathcal{T}, \mathcal{U})$ is called a *differential triangular system* if $\mathcal{T}$ is a differential triangular set.

Compared with the algebraic triangular system [35], the above definition imposes extra constraints on the separants of the polynomials in the differential triangular sets.

Definition 2.3. Let $\mathcal{F}$ be a differential polynomial set $\mathcal{F} \subset \mathbb{K}\{\boldsymbol{x}\}$. Then a finite number of (weak) differential triangular systems

$(\mathcal{T}_1, \mathcal{U}_1), \ldots, (\mathcal{T}_s, \mathcal{U}_s)$ is called a *differential triangular decomposition* of $\mathcal{F}$ into (weak) differential triangular systems if $\mathsf{Z}(\mathcal{F}) = \bigcup_{i=1}^{s} \mathsf{Z}(\mathcal{T}_i/\mathcal{U}_i)$.

The process to construct triangular decomposition of $\mathcal{F}$ into (weak) differential triangular systems is also called *differential triangular decomposition* for $\mathcal{F}$. We are mainly interested in the solutions of $\mathcal{F} = 0$, then the zero relationship $\mathsf{Z}(\mathcal{F}) = \bigcup_{i=1}^{s} \mathsf{Z}(\mathcal{T}_i/\mathcal{U}_i)$ indicates that this is reduced to compute all the zero sets $\mathsf{Z}(\mathcal{T}_i/\mathcal{U}_i)$ for $i = 1, \ldots, s$, which are much easier because of the triangular structure of differential triangular systems [1, 14, 32].

For a differential polynomial set $\mathcal{F} \subset \mathbb{K}\{\boldsymbol{x}\}$, we denote $\mathcal{F}^{(i)} := \{F \in \mathcal{F} \mid \mathrm{ord}(F) = i\}$ for $i = 0, \ldots, n$, and the smallest integer i $(0 \le i \le n)$ such that $\#\mathcal{F}^{(j)} = 0$ or 1 for each $j = i{+}1, \ldots, n$ is called the *level* of $\mathcal{F}$ and denoted by $\mathrm{level}(\mathcal{F})$. Obviously, if $\mathrm{level}(\mathcal{F}) = 0$ and $\mathcal{F}^{(0)} = \emptyset$, then $\mathcal{F}$ forms a weak differential triangular set.

This paper is mainly focused on algorithms for triangular decomposition in top-down style for ordinary differential polynomial sets. Let $\mathcal{F}$ be a differential polynomial set in $\mathbb{K}\{\boldsymbol{x}\}$ and Φ be a set of pairs of differential polynomial sets, initialized as $\{(\mathcal{F}, \emptyset)\}$. Then an algorithm for computing differential triangular decomposition of $\mathcal{F}$ is said to be in *top-down style* if for each $(\mathcal{P}, \mathcal{Q}) \in \Phi$ with $\mathrm{level}(\mathcal{P}) = k > 0$, this algorithm handles the differential polynomials in $\mathcal{P}^{(k)}$ and $\mathcal{Q}^{(k)}$ to produce finitely many differential polynomials sets $\mathcal{P}_1, \ldots, \mathcal{P}_s$ and $\mathcal{Q}_1, \ldots, \mathcal{Q}_s$ such that the following conditions hold:

(a) $\mathsf{Z}(\mathcal{P}/\mathcal{Q}) = \bigcup_{i=1}^{s} \mathsf{Z}(\mathcal{P}_i/\mathcal{Q}_i)$;
(b) for each $i = 1, \ldots, s$, $\mathcal{P}_i^{(j)} = \mathcal{P}^{(j)}$ and $\mathcal{Q}_i^{(j)} = \mathcal{Q}^{(j)}$ for $j = k+1, \ldots, n$;
(c) there exists some integer ℓ $(1 \le \ell \le s)$ such that $\#\mathcal{P}_\ell^{(k)} = 0$ or 1, and the other $(\mathcal{P}_i, \mathcal{Q}_i)$ $(i \ne \ell)$ are put into Φ for later computation.

The definition above for differential triangular decomposition in top-down style is an analogue of the corresponding one for (algebraic) triangular decomposition [20], for the ordinary differential and algebraic triangular sets are both defined w.r.t. $x_1, \ldots, x_n$.

2.3 Differential pseudo-division and subresultant regular subchain

Differential pseudo-division and computation of subresultant regular subchains are two common operations to reduce differential polynomials to new ones of lower ranks, together with associated zero relationships. Corresponding splitting strategies are also designed based on these operations in differential triangular decomposition.

Differential pseudo-division

Let $F, G \in \mathbb{K}\{\boldsymbol{x}\}$ be two differential polynomials. Then there exist methods to (partially) reduce F w.r.t. G: one can find nonnegative integers a, b, and c such that $\mathrm{sep}(G)^a F = R_1 \mod [G]$ and $\mathrm{sep}(G)^b\, \mathrm{ini}(G)^c F = R_2 \mod [G]$, and R_1 and R_2 are respectively partially reduced and reduced w.r.t. G. R_1 and R_2 here are called the *partial differential* and *differential pseudo remainders* of F w.r.t. G respectively, denoted by $\mathrm{pd\text{-}prem}(F, G)$ and $\mathrm{d\text{-}prem}(F, G)$, and the process above for computing R_2 from F and G is called the *differential pseudo-division* of F w.r.t. G [37].

Based on the differential pseudo-division, the zero relationship in Proposition 2.4 holds (see [32, Sec. 3] for a proof). The splitting strategy based on this zero relationship is commonly used in algorithms for differential triangular decomposition [2, 32], and it is also the one used in Algorithm 1 we study in Section 4.

Proposition 2.4. *Let $\mathcal{P}, \mathcal{Q}$ be two differential polynomial sets and T be a differential polynomial in $\mathcal{P}^{(k)}$. Then the following zero relationship holds*

$$\begin{aligned}\mathsf{Z}(\mathcal{P}/\mathcal{Q}) = {} & \mathsf{Z}(\mathcal{P}\setminus\{T\} \cup \{\mathrm{ini}(T), \mathrm{tail}(T)\}/\mathcal{Q}) \\ & \cup \mathsf{Z}(\mathcal{P} \cup \{\mathrm{sep}(T)\}/\mathcal{Q} \cup \{\mathrm{ini}(T)\}) \\ & \cup \mathsf{Z}(\mathcal{P}'/\mathcal{Q} \cup \{\mathrm{ini}(T), \mathrm{sep}(T)\}),\end{aligned}$$

where $\mathcal{P}' := \mathcal{P}\setminus\mathcal{P}^{(k)} \cup \{T\} \cup \{\mathrm{d\text{-}prem}(P, T) : P \in \mathcal{P}^{(k)}\setminus\{T\}\}$.

Subresultant regular subchain

Let $F, G \in \mathbb{K}\{\boldsymbol{x}\}$ be two differential polynomials such that the derivative x_{ij} appears effectively in both of them, and let X be the set of all the derivatives appearing in F or G except x_{ij}. Then write $F = \sum_{k=0}^{p} a_k\, x_{ij}^k$ and $G = \sum_{\ell=0}^{q} b_\ell\, x_{ij}^\ell$ with $a_k, b_\ell \in \mathbb{K}[X]$.

The kth *subresultant* H_k of F and G w.r.t. x_{ij} can be constructed via the determinants of certain sub-matrices of the Sylvester matrix of F and G for $k = 0, \ldots, \mu - 1$, where $\mu := p - 1$ when $p > q$ and $\mu := q$ otherwise. In particular, the kth subresultant H_k is said to be *regular* if its degree equals k. Then the sequence $H_{\mu-1}, H_{\mu-2}, \ldots, H_0$ is called the *subresultant chain* of F and G w.r.t. x_{ij}. Furthermore, let $H_{d_1}, \ldots, H_{d_r}$ be the regular subresultants in $H_{\mu-1}, \ldots, H_0$ with $d_1 > \cdots > d_r$. Then the sequence $H_{d_1}, \ldots, H_{d_r}$ is called the *subresultant regular subchain* (short as SRS) of F and G w.r.t. x_{ij}.

The zero relationship in Proposition 2.5 holds among polynomials obtained from the subresultant regular subchain of two polynomials (a similar proposition (Prop. 5.2) is presented in [15] in the language of ideals). The splitting strategy based on this zero relationship is used in Algorithm 2 we study in Section 5. Note that in Proposition 2.5 below, each H_i is of lower rank than T_2 for $i = 2, \ldots, r$, and $\mathrm{lc}(F, x)$ denotes the leading coefficient of F viewed as a univariate polynomial in x. Note that x does not need to appear in F effectively, and when it does not appear $\mathrm{lc}(F, x) = F$ naturally.

Proposition 2.5 ([35, Lem. 2.4.2]). *Let $T_1, T_2 \in \mathbb{K}\{\boldsymbol{x}\}$ be two differential polynomials such that $\mathrm{ld}(T_1) = \mathrm{ld}(T_2) = x_{ij}$ and T_2 is of lower rank than T_1, and $H_2, \ldots, H_r$ be the subresultant regular subchain of T_1 and T_2 w.r.t. x_{ij}. Denote $I := \mathrm{lc}(T_2, x_{ij})$ and $I_i := \mathrm{lc}(H_i, x_{ij})$ for $i = 2, \ldots, r$. Then the following zero relationship holds*

$$\mathsf{Z}(\{T_1, T_2\}/I) = \bigcup_{i=2}^{r} \mathsf{Z}(\{H_i, I_{i+1}, \ldots, I_r\}/II_i).$$

3 DIFFERENTIAL ASSOCIATED GRAPHS

We first recall the definition of chordal graphs via the perfect elimination ordering, and then associate an arbitrary differential polynomial set with a graph to reflect how the differential indeterminates are connected in the set.

Definition 3.1. Let $G = (V, E)$ be an undirected graph, where $V = \{x_1, \ldots, x_n\}$ is the set of its vertices and E the set of its edges. Then a vertex ordering $x_{i_1} < x_{i_2} < \cdots < x_{i_n}$ is called a *perfect*

elimination ordering of G if for each $j = i_1, \ldots, i_n$, any two distinct vertices in

$$\{x_k : x_k < x_j \text{ and } (x_k, x_j) \in E\} \tag{1}$$

are connected by an edge in E. A graph G is said to be *chordal* if it has a perfect elimination ordering.

For a differential polynomial $F \in \mathbb{K}\{\boldsymbol{x}\}$, we define the *differential indeterminate support* of F to be

$$\mathrm{supp}(F) = \{x_i | \exists \text{ some integer } j \in \mathbb{Z}_{\geq 0} \text{ such that } \deg(F, x_{ij}) > 0\}.$$

For a differential polynomial set $\mathcal{F} \subset \mathbb{K}\{\boldsymbol{x}\}$, $\mathrm{supp}(\mathcal{F}) := \bigcup_{F \in \mathcal{F}} \mathrm{supp}(F)$.

Definition 3.2. Let $\mathcal{F} \subset \mathbb{K}\{\boldsymbol{x}\}$ be a differential polynomial set. Then the *differential associated graph* of $\mathcal{F}$, denoted by $\tilde{G}(\mathcal{F})$, is an undirected graph (V, E) such that the vertices $V := \mathrm{supp}(\mathcal{F})$ and $E := \{(x_i, x_j) : \exists F \in \mathcal{F}$ such that $x_i, x_j \in \mathrm{supp}(F)\}$.

Definition 3.3. A differential polynomial set $\mathcal{F} \subset \mathbb{K}\{\boldsymbol{x}\}$ is said to be *chordal* if $\tilde{G}(\mathcal{F})$ is chordal.

The definition above for differential associated graphs is an analogue of the (algebraic) associated graphs defined in [6, 19]. The difference between these two definitions lies in the fact that for a differential polynomial, its variables are indeed among the derivatives x_{ij} instead of among the differential indeterminates $x_1, \ldots, x_n$. Denote the set of all possible derivatives $\{x_{ij} | i = 1, \ldots, n, j \in \mathbb{Z}_{\geq 0}\}$ by X. Then for a differential polynomial set, when viewed as an algebraic polynomial set in $\mathbb{K}[X]$, the vertices of its (algebraic) associated graph are from X, while the vertices of its differential associated graph are from $x_1, \ldots, x_n$. In this sense, in fact we group the cluster of vertices from $x_i, x_{i1}, x_{i2}, \ldots$ in the (algebraic) associated graph into one vertex x_i in the differential associated graph for each $i = 1, \ldots, n$. This difference is illustrated in the following example.

Consider a differential polynomial set $\mathcal{F}$ in $\mathbb{K}\{x_1, x_2, x_3, x_4, x_5\}$ with $\mathbb{K} = \mathbb{Q}(t)$:

$$\begin{aligned}\mathcal{F} := \{&tx_{22} + x_2 + x_1, x_3 + tx_{11} + x_1,\\ &(1+t)x_{41}^2 + x_{21} + tx_2 + x_{32}^2, x_{51}^2 + x_{31}x_2 + x_3x_{21}\}.\end{aligned} \tag{2}$$

Then the differential associated graph $\tilde{G}(\mathcal{F})$ and (algebraic) associated graph $G(\mathcal{F})$ are shown in Figure 1 below.

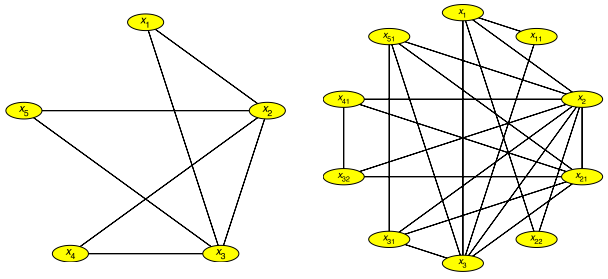


Figure 1: Differential associated graph $\tilde{G}(\mathcal{F})$ (left) and (algebraic) associated graph $G(\mathcal{F})$ (right) for $\mathcal{F}$ in (2)

The differential associated graph $\tilde{G}(\mathcal{F})$ is a chordal one and thus $\mathcal{F}$ is chordal. For one's information, the (algebraic) associated graph $G(\mathcal{F})$ is also a chordal one with $x_1 < x_2 < x_{22} < x_3 < x_{11} < x_{21} < x_{31} < x_{51} < x_{41} < x_{32}$ as one perfect elimination ordering. The underlying reason we choose not to use the (algebraic) associated graph for the differential one is that otherwise the vertices of the associated graphs of our interest may change in the process of common operations in differential triangular decomposition, like the differential pseudo-division as shown below.

Consider two differential polynomials $F = x_{31} + x_1$ and $G = x_2x_3 + x_{21} - 1$ in $\mathbb{Q}(t)\{x_1, x_2, x_3\}$. All the derivatives effectively appearing in $\{F, G\}$ are $x_1, x_2, x_{21}, x_3, x_{31}$. If one computes

$$R := \text{d-prem}(F, G) = x_1x_2^2 - x_2x_{22} + x_{21}^2 - x_{21}$$

and replace F with R, then the derivatives appearing in $\{R, G\}$ are $x_1, x_2, x_{21}, x_{22}, x_3$, with x_{22} newly introduced and x_{31} missing.

If we adopted the definition of (algebraic) associated graph for the differential associated graph of a differential polynomial set, we would have to study graphs with different vertices for the changes of the differential associated graphs in differential triangular decomposition, and we do not see an effective way to do so.

4 CHORDALITY OF WANG'S METHOD FOR ORDINARY DIFFERENTIAL TRIANGULAR DECOMPOSITION

In this section and Section 5 to follow, we study the changes of the differential associated graphs of the polynomial sets appearing in the process of ordinary differential triangular decomposition by two typical algorithms in top-down style under the condition that the input differential polynomial set is chordal and one of its perfect elimination orderings is used. The aim is to prove that all these differential associated graphs are subgraphs of the input chordal graph, or in other words, to prove that these two algorithms preserves chordality. The method we use is to first reformulate these two algorithms in a strictly top-down style to identify the relationships between the polynomial sets in an arbitrary node and in all cases of its child nodes, and then to prove that the differential associated graphs are subgraphs of the chordal graph case by case: this is the same method used in the earlier study on the chordality of algorithms for triangular decomposition [19–21].

The algorithm, due to Dongming Wang [32], studied in this section is the simplest one structurally whose splitting is based on the differential pseudo-division, and it is an analogue of the so-called Wang's algorithm for triangular decomposition in the algebraic case [31]. It is worth mentioning that the splitting strategy follows the constructive method for elimination by Seidenberg [27] and the Rosenfeld-Gröbner algorithm [2] adopts the same strategy in the ordinary differential case [15].

4.1 Algorithm reformulation

Next we reformulate the original algorithm for ordinary differential triangular decomposition in top-down style in [32], which is essentially a depth-first search of the decomposition tree, into Algorithm 1 below, which focuses more on the relationships between any node in the decomposition process and its child nodes and thus is convenient for our inductive proof on the chordality of the polynomial sets in the decomposition process.

In Algorithm 1 (and Algorithm 2 in Section 5), we use the data structure $(\mathcal{P}, \mathcal{Q}, k)$ to represent a node with two polynomial sets

$\mathcal{P}$ and $\mathcal{Q}$ such that $\#\mathcal{P}^{(i)} = 0$ or 1 for $i = k+1, \ldots, n$. For a set Φ consisting of tuples in the form $(\mathcal{P}, \mathcal{Q}, i)$, let $\Phi^{(k)} := \{(\mathcal{P}, \mathcal{Q}, i) \in \Phi \mid i = k\}$. The subroutine pop($\mathcal{S}$) returns an element from a set $\mathcal{S}$ and then removes it from $\mathcal{S}$.

Note that in the original presentation of Wang's algorithm in [32], one step is to compute d-prem(Q, T) for $Q \in \mathcal{Q}$, but this step is against the top-down style (see [20, Sec. 5.1] for more discussions). So in Algorithm 1 below, we remove this step to guarantee that the modified algorithm is in top-down style. After this modification, the output differential triangular systems are not guaranteed to be *fine* (see [35, pp. 23] for the definition) but the correctness of the algorithm is not affected.

Algorithm 1: Wang's method for ordinary differential triangular decomposition $\Psi := \text{WangDiff}(\mathcal{F})$

Input: $\mathcal{F}$, a differential polynomial set in $\mathbb{K}\{\boldsymbol{x}\}$
Output: Ψ, a set of finitely many weak differential triangular systems which form a triangular decomposition of $\mathcal{F}$

$\Phi := \{(\mathcal{F}, \emptyset, n)\}$; $\Psi := \emptyset$;
for $k = n, \ldots, 1$ **do**
  while $\Phi^{(k)} \neq \emptyset$ **do**
    $(\mathcal{P}, \mathcal{Q}, k) :=$ pop($\Phi^{(k)}$);
    if $\#\mathcal{P}^{(k)} > 1$ **then**
      $T :=$ a polynomial in $\mathcal{P}^{(k)}$ with a minimal rank;
      $\mathcal{P}' := \mathcal{P} \setminus \mathcal{P}^{(k)} \cup \{T\} \cup \{\text{d-prem}(P, T) : P \in \mathcal{P}^{(k)} \setminus \{T\}\}$;
      $\Phi := \Phi \cup \{(\mathcal{P} \setminus \{T\} \cup \{\text{ini}(T), \text{tail}(T)\}, \mathcal{Q}, k)\} \cup \{(\mathcal{P} \cup \{\text{sep}(T)\}, \mathcal{Q} \cup \{\text{ini}(T)\}, k)\} \cup \{(\mathcal{P}', \mathcal{Q} \cup \{\text{ini}(T), \text{sep}(T)\}, k)\}$;
    else if $\#\mathcal{P}^{(k)} = 1$ **then**
      $T :=$ the polynomial in $\mathcal{P}^{(k)}$;
      $\Phi := \Phi \cup \{(\mathcal{P}, \mathcal{Q} \cup \{\text{ini}(T), \text{sep}(T)\}, k-1)\} \cup \{(\mathcal{P} \cup \{\text{sep}(T)\}, \mathcal{Q} \cup \{\text{ini}(T)\}, k)\} \cup \{(\mathcal{P} \setminus \{T\} \cup \{\text{ini}(T), \text{tail}(T)\}, \mathcal{Q}, k)\}$;
    else
      $\Phi := \Phi \cup \{(\mathcal{P}, \mathcal{Q}, k-1)\}$;
for $(\mathcal{P}, \mathcal{Q}, 0) \in \Phi^{(0)}$ **do**
  if $\mathcal{P}^{(0)} \setminus \{0\} = \emptyset$ **then**
    $\Psi := \Psi \cup \{(\mathcal{P} \setminus \{0\}, \mathcal{Q})\}$;
return Ψ;

As shown in Algorithm 1, for any node $(\mathcal{P}, \mathcal{Q}, k)$ in the decomposition process with $k \geq 1$, according to whether $\#\mathcal{P}^{(k)} > 1$ or $= 1$ it has three child nodes as follows: when $\#\mathcal{P}^{(k)} > 1$ they are $(\mathcal{P}_1, \mathcal{Q}_1, k)$, $(\mathcal{P}_2, \mathcal{Q}_2, k)$, and $(\mathcal{P}_3, \mathcal{Q}, k)$, where $\mathcal{P}_1, \mathcal{P}_2, \mathcal{P}_3, \mathcal{Q}_1$, and $\mathcal{Q}_2$ are defined as

$$\begin{aligned} \mathcal{P}_1 &= \mathcal{P} \setminus \mathcal{P}^{(k)} \cup \{T\} \cup \{\text{d-prem}(P, T) : P \in \mathcal{P}^{(k)} \setminus \{T\}\}, \\ \mathcal{Q}_1 &= \mathcal{Q} \cup \{\text{ini}(T), \text{sep}(T)\}, \\ \mathcal{P}_2 &= \mathcal{P} \cup \{\text{sep}(T)\}, \\ \mathcal{Q}_2 &= \mathcal{Q} \cup \{\text{ini}(T)\}, \\ \mathcal{P}_3 &= \mathcal{P} \setminus \{T\} \cup \{\text{ini}(T), \text{tail}(T)\}; \end{aligned} \tag{3}$$

and when $\#\mathcal{P}^{(k)} = 1$ they are $(\mathcal{P}, \mathcal{Q}_1, k-1)$, $(\mathcal{P}_2, \mathcal{Q}_2, k)$, and $(\mathcal{P}_3, \mathcal{Q}, k)$, where $\mathcal{P}_2, \mathcal{P}_3, \mathcal{Q}_1$, and $\mathcal{Q}_2$ are defined as in (3). The relationships between the parent and child nodes in the splittings are illustrated in Figure 2 below.

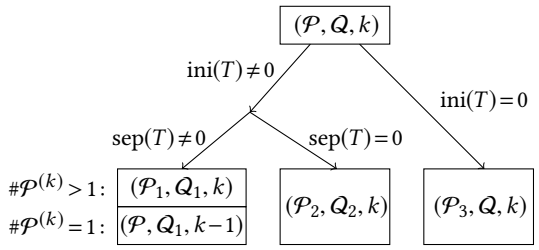


Figure 2: Splittings in Wang's algorithm for ordinary differential triangular decomposition

4.2 Chordality of differential polynomial sets

Next we first prove that the differential pseudo-division does not destroy the relationships between the differential indeterminates defined by the input chordal differential associated graph, and then prove that the differential associated graph of any polynomial set appearing in the decomposition process of WangDiff($\mathcal{F}$), and consequently that of each computed weak differential triangular set, is a subgraph of $\tilde{G}(\mathcal{F})$ if $\mathcal{F}$ is chordal.

Proposition 4.1. *Let $\mathcal{F} \subset \mathbb{K}\{\boldsymbol{x}\}$ be a chordal differential polynomial set with $x_1 < \cdots < x_n$ as one perfect elimination ordering of $\tilde{G}(\mathcal{F})$, $(\mathcal{P}, \mathcal{Q}, k)$ be any node appearing in the decomposition process of* WangDiff($\mathcal{F}$) *such that $\#\mathcal{P}^{(k)} > 1$ and $\tilde{G}(\mathcal{P}) \subset \tilde{G}(\mathcal{F})$, T be a differential polynomial in $\mathcal{P}^{(k)}$ with a minimal rank, and $\mathcal{P}_1$ be defined as in* (3)*. Then $\tilde{G}(\mathcal{P}_1) \subset \tilde{G}(\mathcal{F})$.*

Proof. Clearly supp($\mathcal{P}_1$) $\subset$ supp($\mathcal{P}$), and thus to prove $\tilde{G}(\mathcal{P}_1) \subset \tilde{G}(\mathcal{F})$ it suffices to show that any edge $(x_i, x_j) \in \tilde{G}(\mathcal{P}_1)$ is also in $\tilde{G}(\mathcal{F})$.

For any edge $(x_i, x_j) \in \tilde{G}(\mathcal{P}_1)$, by Definition 3.2 there exists a polynomial $P \in \mathcal{P}_1$ such that $x_i, x_j \in$ supp(P). (1) If $P \in \mathcal{P} \setminus \mathcal{P}^{(k)} \cup \{T\}$, then $P \in \mathcal{P}$ and thus $(x_i, x_j) \in \tilde{G}(\mathcal{P}) \subset \tilde{G}(\mathcal{F})$. (2) If $P \in \{\text{d-prem}(P, T) : P \in \mathcal{P}^{(k)} \setminus \{T\}\}$, then there exists a polynomial $\tilde{P} \in \mathcal{P}^{(k)} \setminus \{T\}$ such that $P =$ d-prem($\tilde{P}, T$), and thus $x_i, x_j \in$ supp($\tilde{P}$) $\cup$ supp(T).

(2.1) If $x_i, x_j \in$ supp($\tilde{P}$) or $x_i, x_j \in$ supp(T), then by $\tilde{P}, T \in \mathcal{P}$ we know that $(x_i, x_j) \in \tilde{G}(\mathcal{P})$. By the assumption $\tilde{G}(\mathcal{P}) \subset \tilde{G}(\mathcal{F})$, we have $(x_i, x_j) \in \tilde{G}(\mathcal{F})$.

(2.2) Otherwise, without loss of generality, we can assume $x_i \in$ supp($\tilde{P}$) and $x_j \in$ supp(T). Then by $\tilde{P}, T \in \mathcal{P}^{(k)}$ we know that $x_i, x_k \in$ supp($\tilde{P}$) and $x_j, x_k \in$ supp(T), and thus $(x_i, x_k) \in \tilde{G}(\mathcal{P}) \subset \tilde{G}(\mathcal{F})$ and $(x_j, x_k) \in \tilde{G}(\mathcal{P}) \subset \tilde{G}(\mathcal{F})$. Then the chordality of $\tilde{G}(\mathcal{F})$ implies $(x_i, x_j) \in \tilde{G}(\mathcal{F})$ and this ends the proof. □

Theorem 4.2. *Let $\mathcal{F} \subset \mathbb{K}[\boldsymbol{x}]$ be a chordal differential polynomial set with $x_1 < \cdots < x_n$ as one perfect elimination ordering of $\tilde{G}(\mathcal{F})$ and $(\mathcal{P}, \mathcal{Q}, k)$ be any node appearing in the decomposition process of* WangDiff($\mathcal{F}$)*. Then $\tilde{G}(\mathcal{P}) \subset \tilde{G}(\mathcal{F})$.*

Proof. The decomposition process of WangDiff($\mathcal{F}$) can be viewed as building a decomposition tree rooted at $(\mathcal{F}, \emptyset, n)$, with the child nodes spawned in the way described in Figure 2. We induce on the depth d of $(\mathcal{P}, \mathcal{Q}, k)$ in this decomposition tree. When $d = 0$, the conclusion naturally holds with $\mathcal{P} = \mathcal{F}$. Now assume that for any node $(\mathcal{P}, \mathcal{Q}, k)$ of depth d in the decomposition tree, we have

$\tilde{G}(\mathcal{P}) \subset \tilde{G}(\mathcal{F})$. Let $(\tilde{\mathcal{P}}, \tilde{\mathcal{Q}}, \tilde{k})$ be of depth $d+1$ and $(\mathcal{P}, \mathcal{Q}, k)$ be its parent node of depth d in the decomposition process. Then it suffices to show that $\tilde{G}(\tilde{\mathcal{P}}) \subset \tilde{G}(\mathcal{F})$ for $\tilde{\mathcal{P}} = \mathcal{P}_1$, $\mathcal{P}_2$, and $\mathcal{P}_3$, where $\mathcal{P}_i$ is constructed as in (3) from $\mathcal{P}$ for $i = 1, 2, 3$ with a polynomial $T \in \mathcal{P}^{(k)}$ with a minimal rank.

Case (1), $\tilde{\mathcal{P}} = \mathcal{P}_1$: the conclusion has been proved in Proposition 4.1.

Case (2), $\tilde{\mathcal{P}} = \mathcal{P}_2$: it is easy to see that $\mathrm{supp}(\mathcal{P}_2) = \mathrm{supp}(\mathcal{P})$. For any edge $(x_i, x_j) \in \tilde{G}(\mathcal{P}_2)$, there exists a polynomial $P \in \mathcal{P}_2 = \mathcal{P} \cup \{\mathrm{sep}(T)\}$ such that $x_i, x_j \in \mathrm{supp}(P)$. When $P \in \mathcal{P}$, clearly $(x_i, x_j) \in \tilde{G}(\mathcal{P})$; when $P = \mathrm{sep}(T)$, we have $x_i, x_j \in \mathrm{supp}(T)$ and thus with $T \in \mathcal{P}$ we have $(x_i, x_j) \in \tilde{G}(\mathcal{P})$.

Case (3), $\tilde{\mathcal{P}} = \mathcal{P}_3$: the proof is the same as that for [20, Prop. 20] in the algebraic case. □

Since the weak differential triangular systems computed by WangDiff($\mathcal{F}$) are extracted from the leaves in the decomposition tree (see line 16 of Algorithm 1), directly we have the following corollary.

Corollary 4.3. *Let $\mathcal{F} \subset \mathbb{K}[\boldsymbol{x}]$ be a chordal differential polynomial set with $x_1 < \cdots < x_n$ as one perfect elimination ordering of $\tilde{G}(\mathcal{F})$ and $(\mathcal{T}_1, \mathcal{U}_1), \ldots, (\mathcal{T}_s, \mathcal{U}_s)$ be the weak differential triangular systems computed by* WangDiff($\mathcal{F}$). *Then $\tilde{G}(\mathcal{T}_i) \subset \tilde{G}(\mathcal{F})$ for $i = 1, \ldots, s$.*

In the algebraic case, the variable sparsity of an arbitrary polynomial set $\mathcal{F} \subset \mathbb{K}[\boldsymbol{x}]$ is reflected in the associated graph $G(\mathcal{F})$ [20, Def. 30]. As an analogue of this, the sparsity for differential indeterminates of a differential polynomial set can be similarly defined. Then the results obtained in Theorem 4.2 and Corollary 4.3 can be interpreted as: when the input differential polynomial set $\mathcal{F} \subset \mathbb{K}\{\boldsymbol{x}\}$ is chordal, Algorithm 1 preserves its sparsity for differential indeterminates. Therefore, sparse versions of this algorithm may be designed accordingly, as what is done in [20]. These comments are also valid for Algorithm 2 studied in Section 5.

4.3 An illustrative example

Let us consider a differential polynomial set $\mathcal{F} := \{F_1, F_2, F_3, F_4\} \subset \mathbb{Q}(t)\{x_1, x_2, x_3, x_4\}$, where

$$F_1 = x_2 + x_1 + 2, \quad F_2 = (x_{21} + 1)x_3 + x_1,$$
$$F_3 = x_3x_4 + x_{31} - 1, \quad F_4 = x_{41} + x_2.$$

The differential associated graph $\tilde{G}(\mathcal{F})$ is shown in the left hand of Figure 3 below.

Next the process to compute a weak differential triangular decomposition with WangDiff($\mathcal{F}$) is demonstrated. First for $k = 4$, $(\mathcal{F}, \emptyset, 4)$ is the only node and $F_3 = x_3x_4 + x_{31} - 1$ is chosen as the polynomial in $\mathcal{F}^{(4)}$ with a minimal rank. Then the differential pseudo-division is performed, spawning two child nodes $(\mathcal{P}_1, \mathcal{Q}_1, 4) = (\{F_1, F_2, F_3, R_1\}, \{x_3\}, 4)$ and $(\{F_1, F_2, x_3, x_{31} - 1, F_4\}, \emptyset, 4)$, where

$$R_1 = \text{d-prem}(F_4, F_3) = -x_3x_{32} + x_{31}^2 - x_{31} + x_2x_3^2.$$

We continue with the node $(\mathcal{P}_1, \mathcal{Q}_1, 4)$, since there is only one polynomial $F_3 \in \mathcal{P}_1^{(4)}$, this node falls into the case $\#\mathcal{P}^{(4)} = 1$ in Algorithm 1, and only one child node $(\mathcal{P}_1, \mathcal{Q}_1, 3)$ is spawned. Continuing with this node, the polynomial $F_2 \in \mathcal{P}_1^{(3)}$ is chosen as the one with a minimal rank. After the differential pseudo-division of R_1 w.r.t. F_2, two child nodes $(\mathcal{P}_2, \mathcal{Q}_2, 3) = (\{F_1, F_2, F_3, R_2\}, \{x_3, x_{21} + 1\}, 3)$ and $(\{F_1, F_3, R, x_{21} + 1, x_1\}, \{x_3\}, 3)$ are spawned, where

$$\begin{aligned} R_2 = \text{d-prem}(R_1, F_2) =& (x_1^2x_{21} + x_1^2)x_{23} - x_1^2x_{22}^2 - (x_1x_{21}^2 - 2x_1x_{21} - x_1)x_{22} \\ &+ x_{11}x_{21}^3 + (x_1^2x_2 - x_1x_{12} + x_{11}^2 + 3x_{11})x_{21}^2 \\ &+ (2x_1^2x_2 - 2x_1x_{12} + 2x_{11}^2 + 3x_{11})x_{21} \\ &+ x_1^2x_2 - x_1x_{12} + x_{11}^2 + x_{11}. \end{aligned}$$

We continue with the node $(\mathcal{P}_2, \mathcal{Q}_2, 3)$, then in the case $\#\mathcal{P}_2^{(3)} = 1$ only one child node $(\mathcal{P}_2, \mathcal{Q}_2, 2)$ is spawned. With $F_1 \in \mathcal{P}_2^{(2)}$ chosen as the one with a minimal rank, the pseudo-division of R_2 w.r.t. F_1 results in one child node, since $\mathrm{ini}(F_1) = \mathrm{sep}(F_1) = 1$: $(\mathcal{P}_3, \mathcal{Q}_3, 2) = (\{F_1, F_2, F_3, R_3\}, \{x_3, x_{21} + 1\}, 2)$, where

$$\begin{aligned} R_3 = \text{d-prem}(R_2, F_1) =& (x_1^2x_{11} - x_1^2)x_{13} - x_1^2x_{12}^2 + x_{11}^3 - (x_1^3 + 2x_1^2 + 2)x_{11}^2 \\ &+ (2x_1^3 + 4x_1^2 + 1)x_{11} - x_1^3 - 2x_1^2. \end{aligned}$$

After the case $\#\mathcal{P}_3^{(2)} = 1$ is handled, the differential triangular set $\mathcal{T} = [R_3, F_1, F_2, F_3]$ is adjoined to Ψ. Then the algorithm will continue to handle the remaining nodes $(\{F_1, F_2, x_3, x_{31} - 1, F_4\}, \emptyset, 4)$ and $(\{F_1, F_3, R, x_{21} + 1, x_1\}, \{x_3\}, 3)$ in Φ to finish the process of differential triangular decomposition.

From the viewpoint of graphs, all the differential associated graphs $\tilde{G}(\mathcal{P}_1)$, $\tilde{G}(\mathcal{P}_2)$, $\tilde{G}(\mathcal{P}_3)$ and $\tilde{G}(\mathcal{T})$ are the same, shown in the right hand of Figure 3.

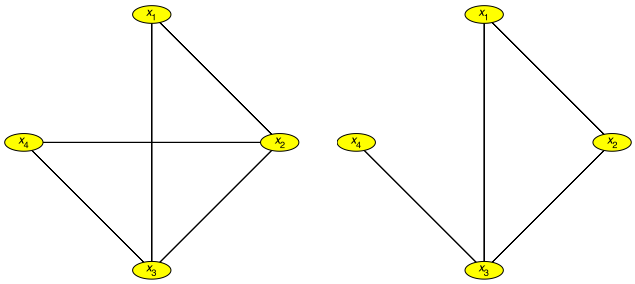


Figure 3: Differential associated graphs $\tilde{G}(\mathcal{F})$ (left) and $\tilde{G}(\mathcal{P}_1) = \tilde{G}(\mathcal{P}_2) = \tilde{G}(\mathcal{P}_3) = \tilde{G}(\mathcal{T})$ (right)

5 CHORDALITY OF SRS-BASED ALGORITHM FOR ORDINARY DIFFERENTIAL TRIANGULAR DECOMPOSITION

5.1 Algorithm reformulation

The algorithm in top-down style for ordinary differential triangular decomposition based on computation of subresultant regular subchains are first reproduced below as Algorithm 2. The subroutine SubRegSubchain(P, Q, x) with $\deg(P, x) > \deg(Q, x)$ returns the subresultant regular subchain $H_2, \ldots, H_r$ of P and Q w.r.t. x. Note that in line 10 only computation of the partial pseudo-reminder is needed [15].

As shown in Algorithm 2, for every node $(\mathcal{P}, \mathcal{Q}, k)$ in the decomposition process, its child nodes are summarized into 3 cases as follows.

Algorithm 2: SRS-based algorithm for ordinary differential triangular decomposition $\Psi := \text{TriDecSRS}(\mathcal{F})$

Input: $\mathcal{F}$, a differential polynomial set in $\mathbb{K}[\boldsymbol{x}]$
Output: Ψ, a set of finitely many weak triangular systems which form a triangular decomposition of $\mathcal{F}$

1 $\Phi := \{(\mathcal{F}, \emptyset, n)\}$; $\Psi := \emptyset$;
2 **for** $k = n, \ldots, 1$ **do**
3 **while** $\Phi^{(k)} \neq \emptyset$ **do**
4 $(\mathcal{P}, \mathcal{Q}, k) := \text{pop}(\Phi^{(k)})$;
5 **if** $\#\mathcal{P}^{(k)} > 1$ **then**
6 $T_2 :=$ a polynomial in $\mathcal{P}^{(k)}$ with a minimal rank;
7 $\Phi := \Phi \cup \{(\mathcal{P} \setminus \{T_2\} \cup \{\text{ini}(T_2), \text{tail}(T_2)\}, \mathcal{Q}, k)\}$ $\cup \{(\mathcal{P} \cup \{\text{sep}(T_2)\}, \mathcal{Q} \cup \{\text{ini}(T_2)\}, k)\}$;
8 $T_1 := \text{pop}(\mathcal{P}^{(k)} \setminus \{T_2\})$;
9 **if** $\text{ld}(T_1) >_d \text{ld}(T_2)$ **then**
10 $\Phi := \Phi \cup \{(\mathcal{P} \setminus \{T_1\} \cup \{\text{pd-prem}(T_1, T_2)\}, \mathcal{Q} \cup \{\text{sep}(T_2)\}, k)\}$;
11 **else**
12 $(H_2, \ldots, H_r) := \text{SubRegSubchain}(T_1, T_2, \text{ld}(T_2))$;
13 $\overline{r} := r$ if $\text{ld}(H_r) = \text{ld}(T_2)$ or $\overline{r} := r - 1$ otherwise;
14 **for** $i = 2, \ldots, \overline{r} - 1$ **do**
15 $\Phi := \Phi \cup \{(\mathcal{P} \setminus \{T_1, T_2\} \cup \{H_i, \text{lc}(H_{i+1}, \text{ld}(T_2)), \ldots, \text{lc}(H_r, \text{ld}(T_2))\}, \mathcal{Q} \cup \{\text{ini}(T_2), \text{lc}(H_i, \text{ld}(T_2))\}, k)\}$;
16 $\Phi := \Phi \cup \{(\mathcal{P} \setminus \{T_1, T_2\} \cup \{H_r, H_{\overline{r}}\}, \mathcal{Q} \cup \{\text{ini}(T_2), \text{lc}(H_{\overline{r}}, x_k)\}, k)\}$;
17 **else if** $\#\mathcal{P}^{(k)} = 1$ **then**
18 $T :=$ the polynomial in $\mathcal{P}^{(k)}$;
19 $\Phi := \Phi \cup \{(\mathcal{P}, \mathcal{Q} \cup \{\text{ini}(T), \text{sep}(T)\}, k-1)\} \cup \{(\mathcal{P} \cup \{\text{sep}(T)\}, \mathcal{Q} \cup \{\text{ini}(T)\}, k)\} \cup \{(\mathcal{P} \setminus \{T\} \cup \{\text{ini}(T), \text{tail}(T)\}, \mathcal{Q}, k)\}$;
20 **else**
21 $\Phi := \Phi \cup \{(\mathcal{P}, \mathcal{Q}, k-1)\}$;
22 **for** $(\mathcal{P}, \mathcal{Q}, 0) \in \Phi^{(0)}$ **do**
23 **if** $\mathcal{P}^{(0)} \setminus \{0\} = \emptyset$ **then**
24 $\Psi := \Psi \cup \{(\mathcal{P} \setminus \{0\}, \mathcal{Q})\}$;
25 **return** Ψ;

(1) When $\#\mathcal{P}^{(k)} > 1$, let T_2 and T_1 be chosen from $\mathcal{P}$ as in lines 6 and 8 in Algorithm 2 respectively.

(1.1) If $\text{ld}(T_1) >_d \text{ld}(T_2)$, then the child nodes are $(\mathcal{P}_1', \mathcal{Q}_1, k)$, $(\mathcal{P}_2, \mathcal{Q}_2, k)$, and $(\mathcal{P}_3, \mathcal{Q}, k)$, where the differential polynomial set $\mathcal{P}_1'$ is defined as

$$\mathcal{P}_1' := \mathcal{P} \setminus \{T_1\} \cup \{\text{pd-prem}(T_1, T_2)\}, \tag{4}$$

and the other sets are as defined in (3), with T_2 replacing T there.

(1.2) If $\text{ld}(T_1) = \text{ld}(T_2)$, then the child nodes are $(\mathcal{P}_{1,2}, \mathcal{Q}_{1,2}, k)$, …, $(\mathcal{P}_{1,\overline{r}}, \mathcal{Q}_{1,\overline{r}}, k)$, $(\mathcal{P}_2, \mathcal{Q}_2, k)$, and $(\mathcal{P}_3, \mathcal{Q}, k)$, where the differential polynomial sets $\mathcal{P}_{1,i}$ and $\mathcal{Q}_{1,i}$ are defined as

$$\mathcal{P}_{1,i} := \begin{cases} \mathcal{P} \setminus \{T_1, T_2\} \cup \{H_i, \text{lc}(H_{i+1}, \text{ld}(T_2)), \ldots, \text{lc}(H_r, \text{ld}(T_2))\}, & \\ \qquad\qquad i = 2, \ldots, \overline{r} - 1 & \\ \mathcal{P} \setminus \{T_1, T_2\} \cup \{H_r, H_{\overline{r}}\}, \qquad i = \overline{r} & \end{cases}$$
$$\mathcal{Q}_{1,i} := \mathcal{Q} \cup \{\text{ini}(T_2), \text{lc}(H_i, \text{ld}(T_2))\}, \qquad i = 2, \ldots, \overline{r} \tag{5}$$

and the other sets are as defined in (3), with T_2 replacing T there.

(2) When $\#\mathcal{P}^{(k)} = 1$, let T be chosen from $\mathcal{P}$ as in line 18 in Algorithm 2. Then the child nodes are $(\mathcal{P}_1, \mathcal{Q}_1, k-1)$, $(\mathcal{P}_2, \mathcal{Q}_2, k)$, and $(\mathcal{P}_3, \mathcal{Q}, k)$, where the polynomial sets are as defined in (3).

The splittings to the child nodes in the decomposition process in these three cases are summarized in Figure 4 below.

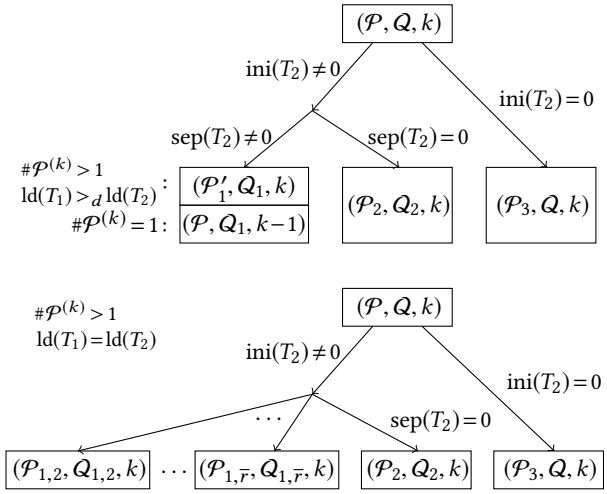


Figure 4: Splittings in SRS-based algorithm for ordinary differential triangular decomposition

5.2 Chordality of differential polynomial sets

Proposition 5.1. *Let $\mathcal{F} \subset \mathbb{K}\{\boldsymbol{x}\}$ be a chordal differential polynomial set with $x_1 < \cdots < x_n$ as one perfect elimination ordering of $\tilde{G}(\mathcal{F})$, and $(\mathcal{P}, \mathcal{Q}, k)$ be an arbitrary node in the decomposition process of* TriDecSRS($\mathcal{F}$) *such that $\#\mathcal{P}^{(k)} > 1$ and $\tilde{G}(\mathcal{P}) \subset \tilde{G}(\mathcal{F})$. Let T_2 and T_1 be chosen from $\mathcal{P}^{(k)}$ as in lines 6 and 8 in Algorithm 2 such that $\text{ld}(T_1) = \text{ld}(T_2)$, $H_2, \ldots, H_r$ be the subresultant regular subchain of T_1 and T_2 w.r.t $\text{ld}(T_2)$, and $\mathcal{P}_1'$ and $\mathcal{P}_{1,i}$ be defined as in* (4) *and* (5) *for $i = 2, \ldots, \overline{r}$ respectively. Then we have $\tilde{G}(\mathcal{P}_1') \subset \tilde{G}(\mathcal{F})$ and $\tilde{G}(\mathcal{P}_{1,i}) \subset \tilde{G}(\mathcal{F})$ for $i = 2, \ldots, \overline{r}$.*

Proof. The proof for $\tilde{G}(\mathcal{P}_1') \subset \tilde{G}(\mathcal{F})$ is easy, and next we focus on the inclusion $\tilde{G}(\mathcal{P}_{1,i}) \subset \tilde{G}(\mathcal{F})$. Clearly for each $i = 2, \ldots, \overline{r}$, $\text{supp}(H_i) \subset \text{supp}(T_1) \cup \text{supp}(T_2) \subset \text{supp}(\mathcal{P})$, and thus $\text{supp}(\mathcal{P}_{1,i}) \subset \text{supp}(\mathcal{P}) \subset \text{supp}(\mathcal{F})$. To prove the inclusion $\tilde{G}(\mathcal{P}_{1,i}) \subset \tilde{G}(\mathcal{F})$, it suffices to show that every edge of $\tilde{G}(\mathcal{P}_{1,i})$ is also an edge of $\tilde{G}(\mathcal{F})$.

For each $i = 2, \ldots, \overline{r}$, consider an arbitrary edge (x_p, x_q) of $\tilde{G}(\mathcal{P}_{1,i})$. Then by Definition 3.2 we know that there exists a differential polynomial $P \in \mathcal{P}_{1,i}$ such that $x_p, x_q \in \text{supp}(P)$.

(1) If $P \in \mathcal{P} \setminus \{T_1, T_2\} \subset \mathcal{P}$, then clearly $(x_p, x_q) \in \tilde{G}(\mathcal{P}) \subset \tilde{G}(\mathcal{F})$ by the assumption.

(2) Else $P \in \{H_i, \text{lc}(H_{i+1}, \text{ld}(T_2)), \ldots, \text{lc}(H_r, \text{ld}(T_2))\}$ in the case of $2 \leq i < \overline{r}$ or $P \in \{H_r, H_{\overline{r}}\}$ in the case of $i = \overline{r}$. Then $x_p, x_q \in \text{supp}(T_1) \cup \text{supp}(T_2)$.

(2.1) If $x_p, x_q \in \text{supp}(T_1)$ or $x_p, x_q \in \text{supp}(T_2)$, then clearly $(x_p, x_q) \in \tilde{G}(\mathcal{P}) \subset \tilde{G}(\mathcal{F})$.

(2.2) Else, without loss of generality, we can assume that $x_p \in \text{supp}(T_1)$ and $x_q \in \text{supp}(T_2)$. Note that $T_1, T_2 \in \mathcal{P}^{(k)}$, and thus $x_k \in \text{supp}(T_1)$ and $\text{supp}(T_2)$. Then $(x_p, x_k), (x_q, x_k) \in \tilde{G}(\mathcal{P}) \subset \tilde{G}(\mathcal{F})$.

With the fact that $x_p, x_q \leq x_k$, the chordality of $\tilde{G}(\mathcal{F})$ implies $(x_p, x_q) \subset \tilde{G}(\mathcal{F})$, and this ends the proof. □

Theorem 5.2. *Let $\mathcal{F} \subset \mathbb{K}[\boldsymbol{x}]$ be a chordal differential polynomial set with $x_1 < \cdots < x_n$ as one perfect elimination ordering of $\tilde{G}(\mathcal{F})$ and $(\mathcal{P}, \boldsymbol{Q}, k)$ be any node in the decomposition process of* TriDecSRS($\mathcal{F}$). *Then $\tilde{G}(\mathcal{P}) \subset \tilde{G}(\mathcal{F})$.*

Proof. The theorem can be proved inductively as in the proof of Theorem 4.2: the case for $\mathcal{P}_1', \mathcal{P}_{1,2}, \ldots, \mathcal{P}_{1,\overline{r}}$ is proved in Proposition 5.1, and the proofs for the cases $\mathcal{P}_2$ and $\mathcal{P}_3$ are the same as Cases (2) and (3) of the proof of Theorem 4.2. □

Corollary 5.3. *Let $\mathcal{F} \subset \mathbb{K}[\boldsymbol{x}]$ be a chordal differential polynomial set with $x_1 < \cdots < x_n$ as one perfect elimination ordering of $\tilde{G}(\mathcal{F})$ and $(\mathcal{T}_1, \mathcal{U}_1), \ldots, (\mathcal{T}_s, \mathcal{U}_s)$ be the weak differential triangular systems computed by* TriDecSRS($\mathcal{F}$). *Then $\tilde{G}(\mathcal{T}_i) \subset \tilde{G}(\mathcal{F})$ for $i = 1, \ldots, s$.*

6 CONCLUDING REMARKS

In this paper we prove that two typical algorithms in top-down style for ordinary differential triangular decomposition based on the pseudo-division and subresultant regular subchain preserve the chordality of the input differential polynomial sets. It is worth mentioning that when the differential associated graph G of the input differential polynomial set is not chordal, we can always find a chordal graph $\tilde{G} \supset G$ via the chordal completion and then work on $\tilde{G}$ instead.

It seems that there exist serious problems for the extension of the theoretical results obtained in this paper to partial differential triangular decomposition. Here we list a couple of the obstacles we think of below.

(1) As discussed below Definition 2.1, the definition of partial differential triangular sets is that for any two distinct polynomials T_i, T_j within, we have $\mathrm{ld}(T_i)$ is not a derivative of $\mathrm{ld}(T_j)$. This means that two polynomials in a partial differential triangular set may have leads as $\frac{\partial x_1}{\partial u}$ and $\frac{\partial x_1}{\partial v}$ of the same differential indeterminate. This will influence our definition of differential associated graph in the way that we need to work on the derivatives directly instead of the differential indeterminates.

(2) For computation of partial differential triangular sets, the property of coherence has to be considered in the decomposition process [26] and this is done by introducing the Δ-polynomials, which seems to be against the top-down strategy. So one has to be very careful to claim one algorithm for partial differential triangular decomposition is in top-down style.

Acknowledgements. This work was partially supported by the National Natural Science Foundation of China (NSFC 11971050 and 11771034) and Beijing Natural Science Foundation (Z180005). The author would like to thank Dongming Wang for inspirational discussions and the referees for their detailed and helpful reviews.

REFERENCES

[1] Thomas Bächler, Vladimir Gerdt, Markus Lange-Hegermann, and Daniel Robertz. 2012. Algorithmic Thomas decomposition of algebraic and differential systems. *J. Symbolic Comput.* 47, 10 (2012), 1233–1266.
[2] François Boulier, Daniel Lazard, François Ollivier, and Michel Petitot. 1995. Representation for the radical of a finitely generated differential ideal. In *Proceedings of ISSAC 1995*. ACM, 158–166.
[3] François Boulier, François Lemaire, and Marc Moreno Maza. 2010. Computing differential characteristic sets by change of ordering. *J. Symbolic Comput.* 45, 1 (2010), 124–149.
[4] François Boulier, François Lemaire, Adrien Poteaux, and Marc Moreno Maza. 2019. An equivalence theorem for regular differential chains. *J. Symbolic Comput.* 93 (2019), 34–55.
[5] Fengjuan Chai, Xiao-Shan Gao, and Chunming Yuan. 2008. A characteristic set method for solving Boolean equations and applications in cryptanalysis of stream ciphers. *J. Systems Science & Complexity* 21, 2 (2008), 191–208.
[6] Diego Cifuentes and Pablo A Parrilo. 2017. Chordal networks of polynomial ideals. *SIAM J. Appl. Algebra Geom.* 1, 1 (2017), 73–110.
[7] Michel Fliess. 1989. Automatique et corps différentiels. In *Forum Math.*, Vol. 1. Walter de Gruyter, Berlin/New York, 227–238.
[8] Michel Fliess and ST Glad. 1993. An algebraic approach to linear and nonlinear control. In *Essays on control*. Springer, 223–267.
[9] Xiao-Shan Gao and Zhenyu Huang. 2012. Characteristic set algorithms for equation solving in finite fields. *J. Symbolic Comput.* 47, 6 (2012), 655–679.
[10] Xiao-Shan Gao, Joris Van Der Hoeven, Chun-Ming Yuan, and Gui-Lin Zhang. 2009. Characteristic set method for differential–difference polynomial systems. *J. Symbolic Comput.* 44, 9 (2009), 1137–1163.
[11] John R. Gilbert. 1994. Predicting structure in sparse matrix computations. *SIAM J. Matrix Anal. Appl.* 15, 1 (1994), 62–79.
[12] Martin C. Golumbic. 2004. *Algorithmic Graph Theory and Perfect Graphs*. Elsevier.
[13] Evelyne Hubert. 2000. Factorization-free decomposition algorithms in differential algebra. *J. Symbolic Comput.* 29, 4-5 (2000), 641–662.
[14] Evelyne Hubert. 2003. Notes on triangular sets and triangulation-decomposition algorithms II: Differential systems. In *International Conference on Symbolic and Numerical Scientific Computation*. Springer, 40–87.
[15] Evelyne Hubert. 2004. Improvements to a triangulation-decomposition algorithm for ordinary differential systems in higher degree cases. In *Proceedings of ISSAC 2004*. ACM, 191–198.
[16] Ellis R. Kolchin. 1973. *Differential Algebra and Agebraic Groups*. Academic Press.
[17] Ziming Li and Dongming Wang. 1999. Coherent, regular and simple systems in zero decompositions of partial differential systems. *Systems Science and Mathematical Sciences* 12 (1999), 43–60.
[18] Dmitry A. Lyakhov, Vladimir P. Gerdt, and Dominik L. Michels. 2017. Algorithmic verification of linearizability for ordinary differential equations. In *Proceedings of ISSAC 2017*. ACM, 285–292.
[19] Chenqi Mou and Yang Bai. 2018. On the chordality of polynomial sets in triangular decomposition in top-down style. In *Proceedings of ISSAC 2018*. ACM, 287–294.
[20] Chenqi Mou, Yang Bai, and Jiahua Lai. 2019. Chordal graphs in triangular decomposition in top-down style. *J. Symbolic Comput.* (2019). In press.
[21] Chenqi Mou and Jiahua Lai. 2020. On the chordality of simple decomposition in top-down style. In *Proceedings of MACIS 2019*. Springer, 138–152.
[22] Seymour Parter. 1961. The use of linear graphs in Gauss elimination. *SIAM Rev.* 3, 2 (1961), 119–130.
[23] Joseph F. Ritt. 1932. *Differential Equations from the Algebraic Standpoint*. AMS.
[24] Joseph F. Ritt. 1950. *Differential Algebra*. AMS.
[25] Donald J. Rose. 1970. Triangulated graphs and the elimination process. *J. Math. Anal. Appl.* 32, 3 (1970), 597–609.
[26] Azriel Rosenfeld. 1959. Specializations in differential algebra. *Trans. Amer. Math. Soc.* 90, 3 (1959), 394–407.
[27] Abraham Seidenberg. 1956. An elimination theory for differential algebra. *Univ. Calif. Math. Publ.* 3 (1956), 31–35.
[28] Lieven Vandenberghe and Martin S Andersen. 2015. Chordal graphs and semidefinite optimization. *Foundations and Trends in Optimization* 1, 4 (2015), 241–433.
[29] Hayato Waki, Sunyoung Kim, Masakazu Kojima, and Masakazu Muramatsu. 2006. Sums of squares and semidefinite program relaxations for polynomial optimization problems with structured sparsity. *SIAM J. Optim.* 17, 1 (2006), 218–242.
[30] Hayato Waki and Masakazu Muramatsu. 2010. A facial reduction algorithm for finding sparse SOS representations. *Oper. Res. Lett.* 38, 5 (2010), 361–365.
[31] Dongming Wang. 1993. An elimination method for polynomial systems. *J. Symbolic Comput.* 16, 2 (1993), 83–114.
[32] Dongming Wang. 1996. An elimination method for differential polynomial systems I. *Systems Science and Mathematical Sciences* 9 (1996), 216–228.
[33] Dongming Wang. 1998. Decomposing polynomial systems into simple systems. *J. Symbolic Comput.* 25, 3 (1998), 295–314.
[34] Dongming Wang. 2000. Computing triangular systems and regular systems. *J. Symbolic Comput.* 30, 2 (2000), 221–236.
[35] Dongming Wang. 2001. *Elimination Methods*. Springer-Verlag, Wien.
[36] Jie Wang, Haokun Li, and Bican Xia. 2019. A new sparse SOS decomposition algorithm based on term sparsity. In *Proceedings of ISSAC 2019*. ACM, 347–354.
[37] Wen-Tsun Wu. 1989. On the foundation of algebraic differential geometry. *Systems Science and Mathematical Sciences* 2 (1989), 289–312.

Approximate GCD by Bernstein Basis, and its Applications

Kosaku Nagasaka*
nagasaka@main.h.kobe-u.ac.jp
Kobe University
Kobe, Hyogo, Japan

ABSTRACT

For the given pair of univariate polynomials generated by empirical data hence with a priori error on their coefficients, computing their greatest common divisor can be done by several known approximate GCD algorithms that are usually for polynomials represented by the power polynomial basis (power form). Recently, there are studies on approximate GCD of polynomials represented by not the power polynomial basis, and especially the Bernstein polynomial basis (Bernstein form) is one of them. we are interested in computing approximate GCD of polynomials in the power form but their perturbation is measured by the Euclidean norm of perturbation in the Bernstein form, and we introduce its applications for computing a reduced rational function, the rational function approximation and Padé approximation to get a better approximation in L_2-norm on $[0, 1]$.

CCS CONCEPTS

• **Computing methodologies** → **Hybrid symbolic-numeric methods**.

KEYWORDS

approximate GCD, Bernstein polynomial basis, rational function approximation, Padé approximation

ACM Reference Format:
Kosaku Nagasaka. 2020. Approximate GCD by Bernstein Basis, and its Applications. In *International Symposium on Symbolic and Algebraic Computation (ISSAC '20), July 20–23, 2020, Kalamata, Greece*. ACM, New York, NY, USA, 8 pages. https://doi.org/10.1145/3373207.3403991

1 INTRODUCTION

Computing the greatest common divisor (GCD) of polynomials is one of fundamental computations in computer algebra since it is important to get a reduced rational function, the square-free decomposition and so on, and in general it can be done by the well-known Euclidean algorithm.

*This work was supported by JSPS KAKENHI Grant Number 19K11827.

ISSAC '20, July 20–23, 2020, Kalamata, Greece

ACM ISBN 978-1-4503-7100-1/20/07…$15.00
https://doi.org/10.1145/3373207.3403991

For polynomials with a priori error on their coefficients (e.g. polynomials generated by empirical data), we employ some of approximate polynomial GCD algorithms instead of the Euclidean algorithm since we have to take into account a priori error (see [2, 10, 18] and references therein, for example). Some of common definitions are as follows where $\|\cdot\|$ denotes a polynomial norm defined later.

Definition 1.1 (approximate GCD with degree). For the given polynomials $f(x), g(x) \in \mathbb{R}[x]$ and the degree $k \in \mathbb{N}$, we compute the polynomial $d(x) \in \mathbb{R}[x]$ called "approximate polynomial GCD" of degree k, which minimizes $\| \delta f \|^2 + \| \delta g \|^2$ (called perturbation) and satisfies $f(x)+\delta f(x) = t(x)d(x)$, $g(x)+\delta g(x) = s(x)d(x)$ and $\deg(d) = k$ for some polynomials $\delta f(x), \delta g(x), s(x), t(x) \in \mathbb{R}[x]$ such that $\deg(\delta f) \leq \deg(f)$ and $\deg(\delta g) \leq \deg(g)$.

Definition 1.2 (approximate GCD with tolerance). For the given polynomials $f(x), g(x) \in \mathbb{R}[x]$ and the tolerance $\varepsilon \in \mathbb{R}_{\geq 0}$, we compute the polynomial $d(x) \in \mathbb{R}[x]$ called "approximate polynomial GCD" of tolerance ε, which has the maximum degree and satisfies $f(x) + \delta f(x) = t(x)d(x)$ and $g(x) + \delta g(x) = s(x)d(x)$ for some polynomials $\delta f(x)$, $\delta g(x)$, $s(x)$, $t(x) \in \mathbb{R}[x]$ such that $\deg(\delta f) \leq \deg(f)$, $\deg(\delta g) \leq \deg(g)$, $\|\delta f\| < \varepsilon \|f\|$ and $\|\delta g\| < \varepsilon \|g\|$.

Most of approximate GCD algorithms are for polynomials represented by the power polynomial basis (i.e. $\{1, x, x^2, \ldots\}$ in $\mathbb{R}[x]$). However, there are studies on approximate GCD for polynomials represented by the Bernstein polynomial basis (see [4] for fundamental properties of this basis and the relation to the Bézier curve). For convenience sake, we call polynomials represented by the power polynomial basis "polynomials in the power form", and by the Bernstein polynomial basis "polynomials in the Bernstein form".

For polynomials in the Bernstein form, Sun et al.[14] propose a method to compute a set of perturbation polynomials that makes the given polynomials having a single common zero. For the general case, Winkler et al. introduce the Sylvester resultant and subresultant matrices, and give algorithms for computing approximate GCD in the Bernstein form (see [15–17] and references therein). Their definition of approximate GCD is different from Definitions 1.1 and 1.2. They assume that there exists the correct exact polynomials hidden by the uniformly distributed random noise bounded by the given componentwise signal-to-noise ratio hence by their definition the approximate GCD is the GCD of these correct exact polynomials. Moreover, Corless and Sevyeri [3] propose another algorithm for computing approximate GCD defined by averages of paired close roots, and follows the ideas introduced by Pan [12]. As for exact polynomials (without

a priori error), Minimair [9] proposes the basis-independent polynomial division algorithm hence we can carry out the Euclidean algorithm for polynomials not in the power form without any conversion to the power form.

1.1 Problem to be solved

We are interested in approximately reducible rational functions appeared in several application areas (e.g. rational function approximation, Padé approximation and so on). For the given rational functions with a priori error or computed by the floating-point arithmetics, we have to use some of approximate GCD algorithms to get a reduced rational function. For example, let $r(x)$ be the following rational function.

$$r(x) = \frac{-0.36x^2 - 0.41x - 0.12}{-0.36x^3 - 0.76x^2 - 0.51x - 0.11}.$$

By the UVGCD [18], we can reduce $r(x)$ by an approximate GCD of degree 1, and we note that UVGCD finds an approximate GCD minimizing the 2-norm of changes in coefficients. The resulting reduced rational function is the following $r_1(x)$.

$$r_1(x) = \frac{-0.423811x - 0.286141}{-0.424212x^2 - 0.69792x - 0.268254}.$$

However, there is a better reduced function w.r.t. the function values in $[0, 1]$, and the following $r_2(x)$ is one of them.

$$r_2(x) = \frac{-0.283167x - 0.137336}{-0.282968x^2 - 0.413249x - 0.125889}.$$

In fact, if we measure the residual relative error by the following err_k for $k = 1, 2$, we have $err_1 = 1.06849\texttt{e-2}$ and $err_2 = 7.77569\texttt{e-6}$. Hence $r_2(x)$ is much better than $r_1(x)$ w.r.t. the function values.

$$err_k = \sqrt{\sum_{i=1}^{100} r_k(x_i) - r(x_i)} \Big/ \sqrt{\sum_{i=1}^{100} r(x_i)}\ ,\ x_i = \frac{i-1}{99}.$$

This toy example suggests us a possibility that there exists a better approximate GCD for reducing rational functions than ever. This is our motivation in this paper.

Especially, for this problem, we are interested in the approximate GCD of polynomials in the power form by Definition 1.1 but the perturbation is measured by the Euclidean norm of perturbation polynomials in the Bernstein form. One may think that this is odd. However, as noted in the next subsections, $p(\alpha)$ is bounded by the minimum and maximum values of coefficients in the Bernstein form for any $\alpha \in [0, 1]$ and $p(x) \in \mathbb{R}[x]$. This property is very useful if any change in function values is more important than that in coefficients and one wants to keep the function values as much as possible. Getting such a better reduced rational function in function values (e.g. $r_2(x)$) is the contribution of this paper.

After the preliminary section including some refinements below, we propose the approximate GCD algorithm in Section 2 with our numerical experiment for computing a reduced rational function. Their applications for the rational function approximation and Padé approximation are given in Section 3 with the results of our numerical experiments. In Section 4, we give some remarks.

1.2 Notations and Definitions

Let $\mathbb{R}$ be the field of real numbers and $\mathbb{R}[x]$ be the polynomial ring over $\mathbb{R}$ in x. Throughout this paper, all the polynomials are treated as defined only on $x \in [0, 1]$ (i.e. transformation of the variable is required if interested in outside this interval). The power polynomial basis of degree n is defined by $\{1, x, \ldots, x^n\}$ and we call polynomials represented by this basis "in the power form", and the Bernstein polynomial basis of degree n on $x \in [0, 1]$ is defined by (using binomials)

$$b_k^n(x) = \binom{n}{k}(1-x)^{n-k}x^k \quad (k = 0, 1, \ldots, n)$$

and we call polynomials represented by this basis "in the Bernstein form".

For any polynomial $p(x) \in \mathbb{R}[x]$ of degree n, we denote it in the power form by $p(x)$ (just with an alphabetical letter) and in the Bernstein form by $\hat{p}(x)$ (with "$\hat{}$" symbol), respectively. By $p_i \in \mathbb{R}$ and $\hat{p}_i \in \mathbb{R}$ $(i = 0, 1, \ldots, n)$ we denote their coefficients, respectively. Therefore, we have

$$p(x) = \sum_{i=0}^{n} p_i x^i = \sum_{i=0}^{n} \hat{p}_i b_i^n(x) = \hat{p}(x).$$

For the coefficients of $p(x)$ and $\hat{p}(x)$, we denote their vector representations in the ascending order by $\vec{p}$ and $\vec{\hat{p}}$, respectively. Hence we have

$$\vec{p} = (p_0\ p_1\ \cdots\ p_n)^t,\ \vec{\hat{p}} = (\hat{p}_0\ \hat{p}_1\ \cdots\ \hat{p}_n)^t \in \mathbb{R}^{n+1}.$$

In this paper, we denote the Euclidean norm by $\|\cdot\|_E$ and L_2-norm by $\|\cdot\|_L$. Therefore, we have

$$\|p(x)\|_E^2 = \|\vec{p}\|_E^2 = \sum_{i=0}^{n} |p_i|^2,\ \|\hat{p}(x)\|_E^2 = \|\vec{\hat{p}}\|_E^2 = \sum_{i=0}^{n} |\hat{p}_i|^2,$$

$$\|p(x)\|_L^2 = \int_0^1 |p(x)|^2 dx = \int_0^1 |\hat{p}(x)|^2 dx = \|\hat{p}(x)\|_L^2\ .$$

In the notations and definitions above, $p(x)$ and $\hat{p}(x)$ have the same degree in general. However, in the Bernstein form, even the coefficients of monomials whose degrees are larger than the degree of the polynomial may not be zeros, as in the following simple example.

$$\begin{aligned} 1 &= 1 \cdot b_0^0(x) \\ 0 \cdot x + 1 &= 1 \cdot b_1^1(x) + 1 \cdot b_0^1(x) \\ 0 \cdot x^2 + 0 \cdot x + 1 &= 1 \cdot b_2^2(x) + 1 \cdot b_1^2(x) + 1 \cdot b_0^2(x). \end{aligned}$$

Therefore, we call the conventional degree in the power form "true degree" and the highest degree of the basis of polynomial "formal degree", in case of ambiguous meanings. For example, formal degrees of the above polynomials on the right hand side are 0, 1 and 2, respectively, while the true degree is 0.

1.3 Bernstein polynomial basis properties

There are so many known studies on the Bernstein polynomial basis. The followings are some of its properties that are useful for our discussion (see [4] for their proofs and further information, for example).

Lemma 1.3 (lower and upper bounds).
For $\hat{p}(x) \in \mathbb{R}[x]$ of formal degree n, we have

$$\forall x \in [0,1],\ \min_{0\le i\le n} \hat{p}_i \le \hat{p}(x) \le \max_{0\le i\le n} \hat{p}_i.$$

Lemma 1.4 (product).
For $\hat{p}(x), \hat{q}(x), \hat{r}(x) \in \mathbb{R}[x]$ of formal degrees $n, m, n+m$, respectively, such that $\hat{r}(x) = \hat{p}(x)\hat{q}(x)$, we have

$$\hat{r}_k = \sum_{j=\max(0,k-m)}^{\min(n,k)} \frac{\binom{n}{j}\binom{m}{k-j}}{\binom{n+m}{k}} \hat{p}_j \hat{q}_{k-j} \quad (k = 0, 1, \ldots, n+m).$$

Lemma 1.5 (basis conversion).
For $p(x) \in \mathbb{R}[x]$ of degree n, we have $\vec{p} = \mathcal{T}\vec{\hat{p}}$ and $\vec{\hat{p}} = \mathcal{T}^{-1}\vec{p}$ where $\mathcal{T} = (t_{ij})$, $\mathcal{T}^{-1} = (\bar{t}_{ij}) \in \mathbb{R}^{(n+1)\times(n+1)}$ and

$$t_{ij} = \begin{cases} (-1)^{i-j}\binom{n}{i}\binom{i}{j} & (i \ge j) \\ 0 & (i < j) \end{cases},\ \bar{t}_{ij} = \begin{cases} \binom{i}{j}/\binom{n}{j} & (i \ge j) \\ 0 & (i < j) \end{cases}.$$

The condition numbers of $\mathcal{T}$ in the vector 1-norm and ∞-norm are $\kappa_1(\mathcal{T}) = \kappa_\infty(\mathcal{T}) = (n+1)\binom{n}{\nu}2^\nu$ where $\nu = \lfloor \frac{2(n+1)}{3} \rfloor$.

As in the lemma, the condition number of $\mathcal{T}$ is very large hence any conversion between the bases is very sensitive (see also the figure 10 in [4]) though this sensitivity does not affect our algorithm directly and the conversions can be done by the exact formulas explicitly.

2 APPROXIMATE GCD ALGORITHM

With the notations and definitions in the previous section, we restate our problem to be solved in this section as follows. We note that $\delta f(x)$ and $\delta g(x)$ in Definition 1.1 are replaced with $\hat{\delta f}(x)$ and $\hat{\delta g}(x)$, respectively in the following definition.

Definition 2.1 (approximate GCD by Bernstein form).
For the given polynomials $f(x), g(x) \in \mathbb{R}[x]$ and the degree $k \in \mathbb{N}$, we compute the polynomial $d(x) \in \mathbb{R}[x]$ called "approximate polynomial GCD with perturbation **in the Bernstein form**" of degree k, which minimizes $\|\hat{\delta f}\|_E^2 + \|\hat{\delta g}\|_E^2$ (called perturbation **in the Bernstein form**) and satisfies $f(x) + \delta f(x) = t(x)d(x)$, $g(x) + \delta g(x) = s(x)d(x)$ and $\deg(d) = k$ for some polynomials $\delta f(x), \delta g(x), s(x), t(x) \in \mathbb{R}[x]$ such that $\deg(\delta f) \le \deg(f)$ and $\deg(\delta g) \le \deg(g)$.

When the correctness of function values (i.e. $f(\alpha)$ and $g(\alpha)$ for $\alpha \in [0,1]$) is more important than the correctness of coefficients, this approximate GCD in Definition 2.1 is better than that in Definition 1.1 since by Lemma 1.3 the computed perturbation in the Bernstein form can bound the perturbation of function values (i.e. $\delta f(\alpha)$ and $\delta g(\alpha)$). This in result gives a better approximation in L_2-norm. For example, reducing the given rational function with a priori error on their coefficients by an approximate GCD of its numerator and denominator meets this condition.

One may think that replacing the objective function (in the power form) in the known approximate GCD algorithms with that in the Bernstein form solves this problem. However, as in Lemma 1.5, the condition number of $\mathcal{T}$ is very large hence this simple approach makes the objective function very sensitive and this optimization becomes very difficult to solve. Therefore, we solve this problem as follows.

(1) forward conversion:
compute $\hat{f}(x), \hat{g}(x)$ from $f(x), g(x)$ by Lemma 1.5.
(2) approximate GCD:
compute an approximate GCD of $\hat{f}(x), \hat{g}(x)$.
(i.e. compute $\hat{t}(x), \hat{s}(x), \hat{d}(x)$ in Definition 2.1)
(3) backward conversion:
compute $t(x), s(x), d(x)$ from $\hat{t}(x), \hat{s}(x), \hat{d}(x)$.

Actually this approach is also affected by the sensitivity of the basis conversion, and the resulting $t(x), s(x), d(x)$ in the step (3) are corresponding to some correct result for polynomials that might be far from $f(x)$ and $g(x)$ (i.e. backward unstable). However, for reducing rational functions this may not be a matter.

For example, let $r(x)$ be a rational function $f(x)/g(x)$ whose numerator and denominator have some approximate GCD. In this case, as the result of forward conversion numerically, we have some polynomials in the Bernstein form that might be very far from the exact polynomials $\hat{f}(x)$ and $\hat{g}(x)$. However, this difference is caused by a tiny difference in the power form. Since the given polynomials have a priori error on their coefficients from the beginning, any tiny difference in the power form is not a matter in the step (1). Hence the resulting reduced rational function $\hat{t}(x)/\hat{s}(x)$ based on the resulting approximate GCD in the step (2) must be a well approximation of $f(x)/g(x)$. Finally, in the step (3), the resulting $t(x)$ and $s(x)$ have a large perturbation region of $\hat{t}(x)$ and $\hat{s}(x)$ by the sensitivity of the backward conversion. This largeness of the region is just for the coefficients and is not for the function values by Lemma 1.3. Therefore, the resulting reduced rational function $t(x)/s(x)$ is a well approximation of $f(x)/g(x)$ w.r.t. the function values in $[0,1]$.

2.1 Sylvester matrix in Bernstein form

Our algorithm is based on the following Sylvester resultant and subresultant matrices for polynomials in the Bernstein form [15–17]. Let $f(x), g(x) \in \mathbb{R}[x]$ be of true degrees $n, m \in \mathbb{N}$, respectively, and $r \in \mathbb{N}$. The conventional Sylvester resultant matrix $S_0(f,g)^t$ and subresultant matrix $S_r(f,g) \in \mathbb{R}^{(n+m-r)\times(n+m-2r)}$ of $f(x)$ and $g(x)$ in the power form are defined as follows.

$$S_r(f,g) = \begin{pmatrix} f_0 & & & g_0 & & \\ f_1 & \ddots & & g_1 & \ddots & \\ \vdots & \ddots & f_0 & \vdots & \ddots & g_0 \\ f_{n-1} & \ddots & f_1 & g_{m-1} & \ddots & g_1 \\ f_n & \ddots & \vdots & g_m & \ddots & \vdots \\ & \ddots & f_{n-1} & & \ddots & g_{m-1} \\ & & f_n & & & g_m \end{pmatrix}.$$

Similarly the Sylvester subresultant matrix of $\hat{f}(x)$ and $\hat{g}(x)$ in the Bernstein form is defined as

$$\hat{S}_r(\hat{f}, \hat{g}) = D_{n+m-r}^{-1} T_r(\hat{f}, \hat{g}) \in \mathbb{R}^{(n+m-r)\times(n+m-2r)}$$

where

$$D_\ell^{-1} = \mathrm{diag}\left(\binom{\ell}{0}^{-1} \quad \binom{\ell}{1}^{-1} \quad \cdots \quad \binom{\ell}{\ell-1}^{-1} \quad \binom{\ell}{\ell}^{-1} \right),$$

$$T_r(\hat{f},\hat{g}) = \begin{pmatrix} \hat{f}_0\binom{n}{0} & & & \hat{g}_0\binom{m}{0} & & \\ \hat{f}_1\binom{n}{1} & \ddots & & \hat{g}_1\binom{m}{1} & \ddots & \\ \vdots & \ddots & \hat{f}_0\binom{n}{0} & \vdots & \ddots & \hat{g}_0\binom{m}{0} \\ \hat{f}_{n-1}\binom{n}{n-1} & \ddots & \hat{f}_1\binom{n}{1} & \hat{g}_{m-1}\binom{m}{m-1} & \ddots & \hat{g}_1\binom{m}{1} \\ \hat{f}_n\binom{n}{n} & \ddots & \vdots & \hat{g}_m\binom{m}{m} & \ddots & \vdots \\ & \ddots & \hat{f}_{n-1}\binom{n}{n-1} & & \ddots & \hat{g}_{m-1}\binom{m}{m-1} \\ & & \hat{f}_n\binom{n}{n} & & & \hat{g}_m\binom{m}{m} \end{pmatrix}.$$

We note that $\hat{S}_0(\hat{f},\hat{g})^t$ is the Sylvester resultant matrix of $\hat{f}(x)$ and $\hat{g}(x)$ in the Bernstein form, and these matrices have the following property [17] as in the power form.

Lemma 2.2. *For the largest integer r such that $\hat{S}_r(\hat{f},\hat{g})$ is not column full rank, let $Q_r\vec{\hat{s}} \in \mathbb{R}^{m-r+1}$ be the first $(m-r+1)$ elements and $-Q_r\vec{\hat{t}} \in \mathbb{R}^{n-r+1}$ be the last $(n-r+1)$ elements in any non-zero vector in the column null space of $\hat{S}_r(\hat{f},\hat{g})$ where*

$$Q_r = \mathrm{diag}\left(\binom{m-r}{0} \quad \cdots \quad \binom{m-r}{m-r} \quad \binom{n-r}{0} \quad \cdots \quad \binom{n-r}{n-r} \right).$$

Then, the corresponding polynomials $\hat{t}(x)$ and $\hat{s}(x)$ are the cofactors of GCD of $\hat{f}(x)$ and $\hat{g}(x)$, respectively, and the degree of GCD is $r+1$.

Winkler and Yang [16] reported that computing the null space of $\hat{S}_r(\hat{f},\hat{g})Q_r$ is numerically better than computing the null space of $\hat{S}_r(\hat{f},\hat{g})$ to get the cofactors of $\hat{f}(x)$ and $\hat{g}(x)$, and $\hat{S}_r(\hat{f},\hat{g})Q_r$ includes $\vec{\hat{s}}$ and $-\vec{\hat{t}}$ directly.

2.2 UVGCD Algorithm

Lemma 2.2 allows us to extend several approximate GCD algorithms in the power form to that in the Bernstein form easily. We focus on the UVGCD algorithm [18] since in the power form this algorithm is mostly the best choice (among QRGCD, ExQRGCD, GPGCD, STLNGCD, Fastgcd and GHLGCD) according to the result of numerical experiments [10]. We also note that the STLN (structured total least norm) is used in [17] but their definition of approximate GCD is different from ours.

The original UVGCD algorithm is for Definition 1.2 hence we briefly show its variation for Definition 1.1 as Algorithm 1. The Gauss-Newton method used in UVGCD is as follows. Let $F_h(\vec{s},\vec{t}) \in \mathbb{R}^{(n+m+3)\times(k+1)}$ and $\vec{b} \in \mathbb{R}^{(n+m+3)}$ be the following matrix and vector.

$$F_h(\vec{s},\vec{t}) = \begin{pmatrix} \vec{h}^t \\ C_k(t) \\ C_k(s) \end{pmatrix},\ \vec{b} = \begin{pmatrix} \beta \\ \vec{f} \\ \vec{g} \end{pmatrix}$$

where $\beta \neq 0, \in \mathbb{R}$, $h(x) \in \mathbb{R}[x]$ of degree k and $C_k(p)$ denotes the convolution matrix[1] of k-th order of $p(x) \in \mathbb{R}[x]$ such that $C_k(p)\vec{q} = \vec{r}$ and $r(x) = p(x)q(x)$ for any polynomial $q(x)$ of degree k. Then, the refinement step is to solve the following unconstrained minimization problem.

$$\min_{d(x),t(x),s(x)} \|F_h(\vec{s},\vec{t})\vec{d} - \vec{b}\|_E .$$

The actual iteration is the following least squares.

$$\begin{pmatrix} \vec{d}_{i+1} \\ \vec{t}_{i+1} \\ \vec{s}_{i+1} \end{pmatrix} = \begin{pmatrix} \vec{d}_i - \vec{\Delta}_d \\ \vec{t}_i - \vec{\Delta}_t \\ \vec{s}_i - \vec{\Delta}_s \end{pmatrix},$$

$$\min_{\vec{\Delta}_d,\,\vec{\Delta}_t,\,\vec{\Delta}_s} \left\| J_h(\vec{d}_i,\vec{s}_i,\vec{t}_i) \begin{pmatrix} \vec{\Delta}_d \\ \vec{\Delta}_t \\ \vec{\Delta}_s \end{pmatrix} - \left(F_h(\vec{s}_i,\vec{t}_i)\vec{d}_i - \vec{b} \right) \right\|_E$$

where $J_h(\vec{d}_i,\vec{s}_i,\vec{t}_i)$ is the following Jacobian of $F_h(\vec{s},\vec{t})$.

$$J_h(\vec{d},\vec{s},\vec{t}) = \begin{pmatrix} \vec{h}^T & & \\ C_k(t) & C_{n-k}(d) & \\ C_k(s) & & C_{m-k}(d) \end{pmatrix}.$$

Moreover, in our preliminary implementation, we use $\beta = 1$ and $\vec{h} = \vec{d}_0 \,/\, \|\vec{d}_0\|_E^2$ where $\vec{d}_0$ is the initial value of $\vec{d}$.

Algorithm 1 UVGCD (with degree, brief version)

Require: $f(x), g(x) \in \mathbb{R}[x]$, and degree $k \in \mathbb{N}$.
Ensure: $d(x), t(x), s(x) \in \mathbb{R}[x]$ in Definition 1.1.
1: construct initial cofactors $t(x), s(x)$ from a non-zero vector in the null space of $S_{k-1}(f,g)$.
2: compute initial $d(x)$ by the least squares.
3: refine $d(x), t(x), s(x)$ by the Gauss-Newton method.
4: **return** $d(x), t(x), s(x)$.

2.3 Algorithm by Bernstein basis

Algorithm 1 can be extended to Definition 2.1 simply by replacing everything with their variations in the Bernstein form. Algorithm 2 is the resulting algorithm.

Let $\hat{F}_{\hat{h}}(\vec{\hat{s}},\vec{\hat{t}}) \in \mathbb{R}^{(n+m+3)\times(k+1)}$ and $\vec{\hat{b}} \in \mathbb{R}^{(n+m+3)}$ be the following matrix and vector.

$$\hat{F}_{\hat{h}}(\vec{\hat{s}},\vec{\hat{t}}) = \begin{pmatrix} \vec{\hat{h}}^t \\ \hat{C}_k(\hat{t}) \\ \hat{C}_k(\hat{s}) \end{pmatrix},\ \vec{\hat{b}} = \begin{pmatrix} \beta \\ \vec{\hat{f}} \\ \vec{\hat{g}} \end{pmatrix}$$

where $\hat{C}_k(\hat{p})$ is the matrix such that $\hat{C}_k(\hat{p})\vec{\hat{q}} = \vec{\hat{r}}$ and $\hat{r}(x) = \hat{p}(x)\hat{q}(x)$ for any polynomial $\hat{q}(x)$ of formal degree k, by Lemma 1.4. We note that $\vec{\hat{q}}$ does not include any binomial part of the Bernstein polynomial basis $b_i^k(x)$.

Then, the refinement step is to solve the following unconstrained minimization problem in the Bernstein form.

$$\min_{\hat{d}(x),\hat{t}(x),\hat{s}(x)} \|F_{\hat{h}}(\vec{\hat{s}},\vec{\hat{t}})\vec{\hat{d}} - \vec{\hat{b}}\|_E .$$

The actual iteration becomes the following least squares.

$$\begin{pmatrix} \vec{\hat{d}}_{i+1} \\ \vec{\hat{t}}_{i+1} \\ \vec{\hat{s}}_{i+1} \end{pmatrix} = \begin{pmatrix} \vec{\hat{d}}_i - \vec{\hat{\Delta}}_{\hat{d}} \\ \vec{\hat{t}}_i - \vec{\hat{\Delta}}_{\hat{t}} \\ \vec{\hat{s}}_i - \vec{\hat{\Delta}}_{\hat{s}} \end{pmatrix},$$

[1] The subresultant matrix is also defined by the convolution matrix (cf. [18]).

Algorithm 2 BFGCD (with degree, brief version)

Require: $f(x), g(x) \in \mathbb{R}[x]$, and degree $k \in \mathbb{N}$.
Ensure: $d(x), t(x), s(x) \in \mathbb{R}[x]$ in Definition 2.1.
1: convert $f(x), g(x)$ to $\hat{f}(x), \hat{g}(x)$ by Lemma 1.5.
2: construct initial cofactors $\hat{t}(x), \hat{s}(x)$ from a non-zero vector in the null space of $\hat{S}_{k-1}(\hat{f}, \hat{g})Q_{k-1}$.
3: compute initial $\hat{d}(x)$ by the least squares (Lemma 1.4).
4: refine $\hat{d}(x), \hat{t}(x), \hat{s}(x)$ by the Gauss-Newton method.
5: convert $\hat{d}(x), \hat{t}(x), \hat{s}(x)$ to $d(x), t(x), s(x)$.
6: **return** $d(x), t(x), s(x)$.

$$\min_{\vec{\Delta}_{\hat{d}}, \vec{\Delta}_{\hat{t}}, \vec{\Delta}_{\hat{s}}} \left\| \hat{J}_{\hat{h}}(\vec{\hat{d}}_i, \vec{\hat{s}}_i, \vec{\hat{t}}_i) \begin{pmatrix} \vec{\hat{\Delta}}_{\hat{d}} \\ \vec{\hat{\Delta}}_{\hat{t}} \\ \vec{\hat{\Delta}}_{\hat{s}} \end{pmatrix} - \left(\hat{F}_{\hat{h}}(\vec{\hat{s}}_i, \vec{\hat{t}}_i)\vec{\hat{d}}_i - \vec{\hat{b}} \right) \right\|_E$$

where $\hat{J}_{\hat{h}}(\vec{\hat{d}}_i, \vec{\hat{s}}_i, \vec{\hat{t}}_i)$ is the following Jacobian of $\hat{F}_{\hat{h}}(\vec{\hat{s}}, \vec{\hat{t}})$.

$$\hat{J}_{\hat{h}}(\vec{\hat{d}}, \vec{\hat{s}}, \vec{\hat{t}}) = \begin{pmatrix} \vec{\hat{h}}^T & & \\ \hat{C}_k(\hat{t}) & \hat{C}_{n-k}(\hat{d}) & \\ \hat{C}_k(\hat{s}) & & \hat{C}_{m-k}(\hat{d}) \end{pmatrix}.$$

2.4 Numerical experiments

We define the following sets of polynomials and rational functions for $m, n, \ell \in \mathbb{N}$ and $\epsilon \in \mathbb{R}_{\geq 0}$, whose numerator and denominator are not coprime approximately.

$$\mathcal{P}_m = \{\, p(x) \in \mathbb{R}_1[x] \mid \deg(p) = m \,\},\ \mathbb{R}_1 = [-1, 1] \subset \mathbb{R},$$
$$\mathcal{P}_m^* = \{\, p(x) \in \mathcal{P}_m \mid \forall \alpha \in [0, 1],\ p(\alpha) \neq 0 \,\},$$
$$\mathcal{R}_{m,n,\ell,\epsilon} = \left\{ \frac{t(x)d(x)+\delta p(x)}{s(x)d(x)+\delta q(x)} \,\middle|\, t(x) \in \mathcal{P}_{m-\ell},\ s(x) \in \mathcal{P}_{n-\ell}^*, \right.$$
$$\left. d(x) \in \mathcal{P}_\ell^*,\ \delta p(x) \in \mathcal{P}_m,\ \delta q(x) \in \mathcal{P}_n, \right.$$
$$\left. \|\delta p\|_E = \epsilon \|t(x)d(x)\|_E,\ \|\delta q\|_E = \epsilon \|s(x)d(x)\|_E \right\}.$$

We have randomly generated a couple of subsets (each set has 100 elements) of $\mathcal{R}_{m,n,\ell,\epsilon}$, and for each rational function $r(x)$ and $k = 1, 2, \ldots, \ell$, we have computed a reduced rational function $\bar{r}(x)$ by approximate GCD of degree k of numerator and denominator, and computed a residual norm:

$$\frac{\|(r(x_1) - \bar{r}(x_1),\ \ldots,\ r(x_{100}) - \bar{r}(x_{100}))^t\|_E}{\|(r(x_1),\ \ldots,\ r(x_{100}))^t\|_E}$$

where $x_i = \frac{i-1}{99}$ for $i = 1, 2, \ldots, 100$.

The notable difference between the Sylvester and convolution matrices in the power and Bernstein forms is the binomial terms. This urges each computation of the matrices in the Bernstein form to use much larger precisions. Therefore we have done the experiments with the double precision and 32 decimal precisions[2], by our preliminary implementations on Mathematica 12.0.

Figures 1, 2, 3 and 4 show the results where the vertical axis denotes the averages of the residual norms in the log scale (e.g. -4 denotes 10^{-4}), "UV", "BF", "BB" and "PN" denote the UVGCD, BFGCD, BFGCD (without the backward conversion to the power form) and Pan's algorithms,

[2]We use "`$MaxPrecision=$MinPrecision=32`" and "`SetPrecision[-,32]`" for 32 decimal precisions (that simulates the quad precision).

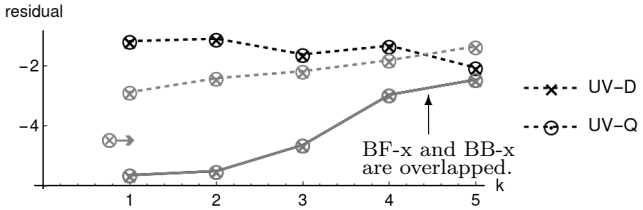


Figure 1: $m = 9,\ n = 10,\ \ell = 5,\ \epsilon =$ 1.0e-3

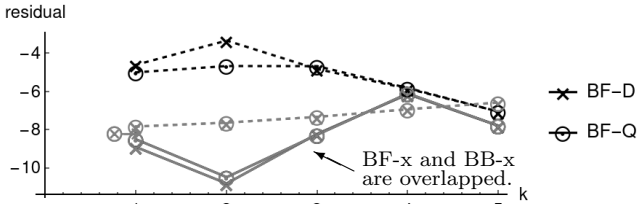


Figure 2: $m = 9,\ n = 10,\ \ell = 5,\ \epsilon =$ 1.0e-8

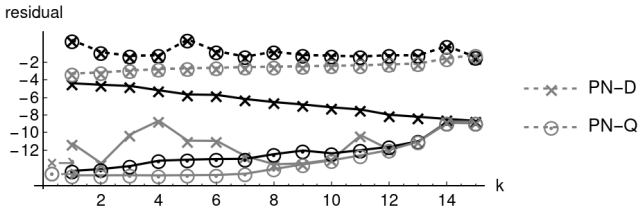


Figure 3: $m = 29,\ n = 30,\ \ell = 15,\ \epsilon =$ 1.0e-3

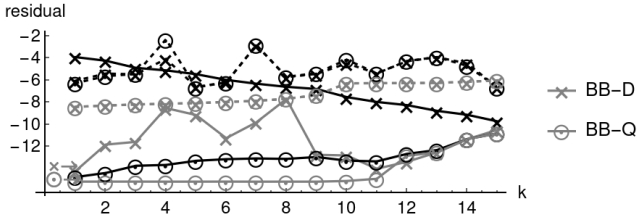


Figure 4: $m = 29,\ n = 30,\ \ell = 15,\ \epsilon =$ 1.0e-8

respectively, and "D" and "Q" denote the double precision and 32 decimal precisions, respectively. The algorithm by Sun et al.[14] can be used for $k = 1$ (reducing by a linear GCD) hence "$\times\!\rightarrow$" and "$\circ\!\rightarrow$" denote its results with the double precision and 32 decimal precisions, respectively, as just a reference.

Our algorithm (BFGCD) works well and is effective for reducing the given rational function by an approximate GCD of its numerator and denominator, even if we use the double precision. However, this also suggests that higher precisions may be required for higher degree polynomials. Moreover, the results by the BFGCD without the backward conversion are better than the BFGCD. This could be caused by numerical instability of the ill-posedness (the large condition number) of the basis conversion.

We note that our implementation of the Pan's algorithm [12] (hence the method by Kai and Noda [8]) is different from the original one that seeks a set of common zeros within the given distance. Our implementation seeks an approximate GCD of the specified degree by increasing the distance (by the quadratic order) until we get the desired one. Moreover, the algorithm by Sun et al.[14] computes only perturbations hence our implementation solves the least squares in the

power form to compute the reduced rational function by the linear factor with the resulting common zero.

3 APPLICATIONS

3.1 Rational function approximation

We focus on rational interpolation that is one of methods for rational function approximation, and its aim is to find a rational function $r(x) = \frac{p(x)}{q(x)}$ that fits the given data points $\{(x_1, v_1), \ldots, (x_\ell, v_\ell)\}$ (i.e. $r(x_i) \approx v_i$ for $i = 1, \ldots, \ell$). Hence we are to find a minimizer of the following optimization problem where $m = \deg(p)$ and $n = \deg(q)$ are given.

$$\min_{p(x),q(x)} \left\| \left(v_1 - \frac{p(x_1)}{q(x_1)}, \ \ldots, \ v_\ell - \frac{p(x_\ell)}{q(x_\ell)} \right)^t \right\|_E . \tag{3.1}$$

Especially we focus on $r(x)$ in the power form though there are recent studies in other representations (rational Krylov spaces [1] and rational barycentric representations [11] for example), and follow some simple algorithms based on the least squares. However, the optimization above is a non-convex problem hence we solve the following linearized one.

$$\min_{p(x),q(x)} \left\| (v_1 q(x_1) - p(x_1), \ \ldots, \ v_\ell q(x_\ell) - p(x_\ell))^t \right\|_E . \tag{3.2}$$

Since the optimization problems (3.1) and (3.2) are different, we use the iterative reweighting method [13] to get better rational function approximations.

One of fundamental problems of rational function approximation with floating-point arithmetics is that the numerator and denominator of resulting rational function may have approximately common zeros if m, n are larger than necessary. For this problem, Kai and Noda [8] proposed to use approximate GCD to reduce the numerator and denominator, and called the method "Hybrid rational function approximation". Their algorithm basically uses the approximate GCD algorithm by Pan [12] with their theoretical upper bound of distance between zeros. In the following experiments, we use UVGCD (Algorithm 1) and BFGCD (Algorithm 2) for detecting an approximate GCD to get a reduced rational function (as Algorithm 3), and compare them with other methods including our implementation of Pan's algorithm mentioned at end of Section 2.

3.1.1 Numerical Experiments.

We define the following set (i.e. sampling data) for $m, n, \ell \in \mathbb{N}$ and $\epsilon_1, \epsilon_2 \in \mathbb{R}_{\geq 0}$, with a priori error.

$$\mathcal{S}_{m,n,\ell,\epsilon_1,\epsilon_2} = \big\{ \{(x_1, v_1), \ldots, (x_{100}, v_{100})\} \subset \mathbb{R}^2 \big| \\ r(x) \in \mathcal{R}_{m,n,\ell,\epsilon_1}, \ x_i = \tfrac{i-1}{99}, \ v_i = r(x_i) + \delta_i, \\ \delta_i \in \mathbb{R}, \ \textstyle\sum_{i=1}^{100} \delta_i^2 = \epsilon_2^2 \sum_{i=1}^{100} r(x_i)^2 \big\} .$$

We have randomly generated a couple of subsets (each set has 100 elements) of $\mathcal{S}_{m,n,\ell,\epsilon_1,\epsilon_2}$. The sets of the erroneous sampling data case are $\mathcal{E}_1, \mathcal{E}_2$ with $m = 19$, $n = 19$, $\ell = 0$, $\epsilon_1 = 0$, $\epsilon_2 = 1.0\texttt{e-}3$ for $\mathcal{E}_1$, and $m = 19$, $n = 19$, $\ell = 0$, $\epsilon_1 = 0$, $\epsilon_2 = 1.0\texttt{e-}8$ for $\mathcal{E}_2$, and the sets of the approximately reducible case are $\mathcal{A}_1, \mathcal{A}_2$ with $m = 19$, $n = 19$, $\ell = 9$, $\epsilon_1 = 1.0\texttt{e-}3$, $\epsilon_2 = 0$ for $\mathcal{A}_1$, and $m = 19$, $n = 19$, $\ell = 9$, $\epsilon_1 = 1.0\texttt{e-}8$, $\epsilon_2 = 0$ for $\mathcal{A}_2$.

Algorithm 3 HRFA (with UVGCD, BFGCD or Pan's)

Require: $\{(x_1, v_1), \ldots, (x_\ell, v_\ell)\} \subset \mathbb{R}^2$, $m, n \in \mathbb{N}$, $\varepsilon \in \mathbb{R}_{\geq 0}$
Ensure: rational function approximation $r(x) = p(x)/q(x)$ such that $\deg(p) \leq m$, $\deg(q) \leq n$ are minimized and the residual norm w.r.t. (3.1) is less than or equal to ε.
1: solve (3.2) by the iterative reweighting method, and let $p(x)/q(x)$ be the resulting rational function.
2: **for** $k = \min(m, n)$ **to** 1 **by** -1 **do**
3: compute cofactors $t(x)$ and $s(x)$ by approximate GCD of degree k, of $p(x)$ and $q(x)$, respectively.
4: **if** the residual norm of $t(x)/s(x) \leq \varepsilon$ **then**
5: $p(x) := t(x)$, $q(x) := s(x)$ and **break**.
6: **end if**
7: **end for**
8: **return** $p(x)/q(x)$.

For each sampling data, by the following methods, we have computed a reduced rational function $\bar{r}(x)$ with $m_{max} = m + 10$ and $n_{max} = n + 10$, whose residual norm (3.1) is less than or equal to $\varepsilon = 10 \max\{\epsilon_1, \epsilon_2\}$, by our preliminary implementations with the double precision on Mathematica 12.0 (but "AAA" is done by Octave 5.1.0).

- **LS:** this solves (3.2) with $\deg(p) = \bar{m}$ and $\deg(q) = \bar{n}$ by the iterative reweighting method for $\bar{n} = \bar{m} = 0, 1, \ldots, n_{max}$ with $\bar{m} \leq m_{max}$ until the residual norm condition is satisfied, and reduces $\bar{m}$ one by one while the residual norm condition is satisfied.
- **LS+BF:** this follows the "LS" and reduce the resulting rational function by approximate GCD (by BFGCD) of degree $k = \min\{\bar{m}, \bar{n}\}, \ldots, 1, 0$ until the residual norm condition is satisfied.
- **UV:** this solves (3.2) with $\deg(p) = m_{max}$ and $\deg(q) = n_{max}$ by the iterative reweighting method and reduce the resulting rational function by approximate GCD (by UVGCD) of degree $k = \min\{m_{max}, n_{max}\}, \ldots, 1, 0$ until the residual norm condition is satisfied (i.e. Algorithm 3).
- **BF:** this follows the "UV" but using BFGCD instead of UVGCD (i.e. Algorithm 3 with BFGCD).
- **PN:** this follows the "UV" but using our implementation of the Pan's algorithm instead of UVGCD.
- **AAA:** this follows the algorithm in [11] but using the Euclidean norm instead of the ∞-norm.

Table 1 shows the results where each value denotes the average of $\deg(\bar{p}) + \deg(\bar{q})$ of detected rational functions $\bar{r}(x) = \bar{p}(x)/\bar{q}(x)$. Our method (with BFGCD) is not better than the methods based on the least squares, however, it is better than that with UVGCD, Pan's and AAA algorithm. Moreover, we note that "LS" and "LS+BF" do not have any difference in this experiments, however, there is a randomly generated example that "LS+BF" is a little bit better than "LS" though it is a very rare case. We have also done the same experiments by "BF" without the backward conversion to the power form, and the result is slightly better than "BF" but not so different.

data set	$\mathcal{E}_1$	$\mathcal{E}_2$	$\mathcal{A}_1$	$\mathcal{A}_2$
LS	5.50	14.55	4.58	12.48
LS+BF	5.50	14.55	4.58	12.48
UV	38.42	49.40	33.20	56.92
BF	10.78	21.54	6.18	14.14
PN	37.62	49.40	50.36	57.64
AAA	31.52	38.24	8.20	14.60

Table 1: degrees of reduced rational functions

3.2 Padé approximation

Padé approximation [6] for the given $f(x)$ and $m, n \in \mathbb{N}$ is to find the rational function $r(x) = p(x)/q(x)$ called "type (m, n) Padé approximant" to $f(x)$ that satisfies

$$r(x) - f(x) = O(x^{m+n+1}),\ \deg(p) = m,\ \deg(q) = n.$$

In general this rational function can be found by solving the following linear equation

$$\begin{pmatrix} f_{m+1} & f_m & \cdots & f_{m-n+1} \\ f_{m+2} & f_{m+1} & \cdots & f_{m-n+2} \\ \vdots & \vdots & \ddots & \vdots \\ f_{m+n} & f_{m+n-1} & \cdots & f_m \end{pmatrix} \begin{pmatrix} q_0 \\ q_1 \\ \vdots \\ q_n \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ \vdots \\ 0 \end{pmatrix}, \tag{3.3}$$

and computing the numerator as

$$\begin{pmatrix} p_0 \\ p_1 \\ \vdots \\ p_m \end{pmatrix} = \begin{pmatrix} f_0 & 0 & \cdots & 0 \\ f_1 & f_0 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ f_m & f_{m-1} & \cdots & f_{m-n} \end{pmatrix} \begin{pmatrix} q_0 \\ q_1 \\ \vdots \\ q_n \end{pmatrix}. \tag{3.4}$$

With exact computations and without a priori error, for large enough $M, N \in \mathbb{N}$ we have the unique reduced representation of rational functions for all $m, n \in \mathbb{N}$ $(m \geq M,\ n \geq N)$.

However, with numerical computations or with a priori error, it is difficult to find a such rational function of the maximum order since we can not reduce the computed rational function due to their numerical error. To overcome this situation, there are some known studies [5, 7] to find a Padé approximant with numerator and denominator of the minimum degrees by basically using the singular values (i.e. numerical matrix ranks). In the following experiments, we use UVGCD (Algorithm 1) and BFGCD (Algorithm 2) for detecting an approximate GCD to determine a Padé approximant of the minimum degree (as Algorithm 4), and compare them with other methods. We note that the lines **1** to **13** are to find the number of possible common zeros of numerator and denominator, and based on Algorithm 2 in [5].

3.2.1 *Numerical Experiments.*

We define the following sets (i.e. Taylor coefficients) for $m, n, \ell \in \mathbb{N}$ and $\epsilon_1, \epsilon_2 \in \mathbb{R}_{\geq 0}$, with a priori error.

$$\mathcal{R}^*_{m,n,\ell,\epsilon_1} = \{r(x) \in \mathcal{R}_{m,n,\ell,\epsilon_1} \mid \{f_0, f_1, \ldots, f_{m+n}\} = \mathrm{Taylor}(r),\ \tfrac{\max_i |f_i|}{\min_i |f_i|} \leq 2^{m+n}\Big\},$$

$$\mathcal{C}_{m,n,\ell,\epsilon_1,\epsilon_2} = \{\{f_0 + \delta_0, f_1 + \delta_1, \ldots, f_{m+n} + \delta_{m+n}\} \mid r(x) \in \mathcal{R}^*_{m,n,\ell,\epsilon_1},\ \{f_0, f_1, \ldots, f_{m+n}\} = \mathrm{Taylor}(r),\ \delta_i \in \mathbb{R},\ \textstyle\sum_{i=0}^{m+n} \delta_i^2 = \epsilon_2^2 \sum_{i=0}^{m+n} f_i^2\}$$

Algorithm 4 Padé app. (with UVGCD, BFGCD or Pan's)

Require: $\{f_0, f_1, \ldots, f_\ell\} \subset \mathbb{R}$, $m, n \in \mathbb{N}$ $(\ell \geq m+n)$, $\varepsilon \in \mathbb{R}_{\geq 0}$
Ensure: Padé approximant $r(x) = p(x)/q(x)$ such that $r(x) - f(x) = O(x^{m+n+1})$ and $\deg(p) \leq m$, $\deg(q) \leq n$ are minimized by approximate GCD.

```
1:  k := 0, τ := ε ‖(f_0, f_1, ..., f_ℓ)^t‖_E.
2:  if |f_i| ≤ τ (i = 0, 1, ..., m) then
3:    k := min{m, n}.
4:  else
5:    while n ≠ 0 do
6:      compute the singular values of the n × (n+1) matrix
        in (3.3), and let ρ be the number of singular values
        that are greater than τ.
7:      if ρ < n and n − ρ ≤ m then
8:        n := ρ, m := m − (n − ρ), k := k + (n − ρ).
9:      else
10:       break.
11:     end if
12:   end while
13: end if
14: solve (3.3) and compute (3.4) with m, n,
      and let p(x)/q(x) be the resulting rational function.
15: compute cofactors t(x) and s(x) by approximate GCD
                of degree k, of p(x) and q(x), respectively.
16: return p(x)/q(x).
```

where Taylor(r) denotes the coefficients $\{f_0, f_1, \ldots, f_{m+n}\}$ of Taylor series of $r(x)$ at $x = 0$.

We have randomly generated a couple of subsets (each set has 100 elements) of $\mathcal{C}_{m,n,\ell,\epsilon_1,\epsilon_2}$. The sets of the erroneous coefficients case are $\mathcal{E}_3, \mathcal{E}_4$ with $m = 19$, $n = 20$, $\ell = 0$, $\epsilon_1 = 0$, $\epsilon_2 = 1.0\mathtt{e}{-3}$ for $\mathcal{E}_3$, and $m = 19$, $n = 20$, $\ell = 0$, $\epsilon_1 = 0$, $\epsilon_2 = 1.0\mathtt{e}{-8}$ for $\mathcal{E}_4$, and the sets of the approximately reducible case are $\mathcal{A}_3, \mathcal{A}_4$ with $m = 19$, $n = 20$, $\ell = 10$, $\epsilon_1 = 1.0\mathtt{e}{-3}$, $\epsilon_2 = 0$ for $\mathcal{A}_3$, and $m = 19$, $n = 20$, $\ell = 10$, $\epsilon_1 = 1.0\mathtt{e}{-8}$, $\epsilon_2 = 0$ for $\mathcal{A}_4$.

For each coefficients data, by the following methods, we have computed a reduced Padé approximant $\bar{r}(x)$ with $m_{max} = m$, $n_{max} = n$ and $\varepsilon = 1.0\mathtt{e}{-12}$, by our preliminary implementations with the double precision on Mathematica 12.0.

- **DI:** this solves (3.3) and compute (3.4) with m_{max}, n_{max}.
- **GGT:** this follows the algorithm in [5].
- **GGT_WO8:** this follows the algorithm in [5] but without the step 8 (this step reduces the degrees of numerator and denominator, independently).
- **UV:** this follows the "GGT_WO8" to determine m, n, solves (3.3) and compute (3.4) with m_{max}, n_{max}, and reduce the resulting rational function by approximate GCD (by UVGCD) of degree $k = m_{max} - m$ (i.e. Algorithm 4 with UVGCD).
- **BF:** this follows the "UV" but using BFGCD instead of UVGCD (i.e. Algorithm 4 with BFGCD).
- **PN:** this follows the "UV" but using our implementation of the Pan's algorithm instead of UVGCD.

data set	$\mathcal{E}_3$	$\mathcal{E}_4$
DI	0.845069 (39.0)	0.0627637 (39.00)
GGT	0.845069 (39.0)	0.065353 (38.41)
GGT_WO8	0.845069 (39.0)	0.0675395 (38.48)
UV	0.845069 (39.0)	0.143128 (38.48)
BF	0.845069 (39.0)	0.0627698 (38.48)
PN	0.845069 (39.0)	0.0688095 (38.48)

Table 2: residual norm of function values of Padé

data set	$\mathcal{A}_3$	$\mathcal{A}_4$
DI	3.12831e-7 (39.00)	2.72861e-8 (39.00)
GGT	2.18353e-4 (37.46)	4.73386e-6 (26.16)
GGT_WO8	2.18353e-4 (37.46)	4.73386e-6 (26.16)
UV	1.01356e-2 (37.46)	5.01329e-6 (26.16)
BF	3.12942e-7 (37.46)	5.73857e-8 (26.16)
PN	5.51157e-4 (37.46)	1.45860e-3 (26.04)

Table 3: residual norm of function values of Padé

Tables 2 and 3 show the results where each cell denotes the average of the following residual norms of function values and the sums of degrees $\deg(\bar{p}) + \deg(\bar{q})$ for each $r(x)$ and its computed Padé approximant $\bar{r}(x) = \bar{p}(x)/\bar{q}(x)$.

$$\frac{\|(r(x_1) - \bar{r}(x_1),\ \ldots,\ r(x_{100}) - \bar{r}(x_{100}))^t\|_E}{\|(r(x_1),\ \ldots,\ r(x_{100}))^t\|_E}$$

where $x_i = \frac{i-1}{99}$ for $i = 1, 2, \ldots, 100$. Our algorithm (BFGCD) works well and is much effective for the approximately reducible cases. Moreover, we have also done the same experiments by "BF" without the backward conversion to the power form, and the result is slightly better than "BF" but not so different.

4 REMARKS

In this paper, we have proposed the BFGCD algorithm that works very effective according to our numerical experiments. Especially, it is better than the UVGCD and Pan's algorithms w.r.t. the function values in $[0, 1]$. Though the basis conversion between the power and the Bernstein forms is ill-conditioned, the property in Lemma 1.3 is very useful and may be extensible for other applications.

For example, in our numerical experiments in Section 3.1.1, only "LS" and "LS+BF" try to reduce the degree of numerator independently of its denominator, and our methods ("UV" and "BF") only try to reduce the degrees of numerator and denominator together since they are based on approximate GCD. However, in "BF" the given polynomials are converted hence in the Bernstein form, it may be possible to decrease the degrees of numerator and denominator independently, by the following useful properties.

Lemma 4.1 (degree elevation).
For $\hat{p}(x), \hat{q}(x) \in \mathbb{R}[x]$ of formal degrees $n, n+1$, respectively, such that $\hat{p}(x) = \hat{q}(x)$, we have $\hat{q}_0 = \hat{p}_0$, $\hat{q}_{n+1} = \hat{p}_n$ and

$$\hat{q}_k = \frac{k}{n+1}\hat{p}_{k-1} + \left(1 - \frac{k}{n+1}\right)\hat{p}_k \quad (k = 1, 2, \ldots, n).$$

Lemma 4.2 (best degree reduction).
For $\hat{p}(x) \in \mathbb{R}[x]$ of formal degree n, let $\hat{q}(x), \hat{r}(x) \in \mathbb{R}[x]$ be of formal degrees m, n with $m < n$, respectively, such that $\hat{r}(x)$ is the result of degree elevation of $\hat{q}(x)$ by Lemma 4.1. Then, the followings have the same minimizer $\hat{q}(x)$.

$$\min_{\hat{q}(x)} \|\hat{p}(x) - \hat{q}(x)\|_L \quad \textit{and} \quad \min_{\hat{q}(x)} \|\vec{\hat{p}} - \vec{\hat{r}}\|_E\ .$$

This approach and other possible applications for polynomials in the Bernstein form will be the future works.

REFERENCES

[1] Mario Berljafa and Stefan Güttel. 2015. Generalized rational Krylov decompositions with an application to rational approximation. *SIAM J. Matrix Anal. Appl.* 36, 2 (2015), 894–916.

[2] Paola Boito. 2011. *Structured matrix based methods for approximate polynomial GCD.* Tesi. Scuola Normale Superiore di Pisa (Nuova Series) [Theses of Scuola Normale Superiore di Pisa (New Series)], Vol. 15. Edizioni della Normale, Pisa. xvi+199 pages.

[3] Robert M. Corless and Leili R. Sevyeri. 2019. Approximate GCD in Bernstein basis. *ACM Commun. Comput. Algebra* 53, 3 (2019), 103–106.

[4] Rida T. Farouki. 2012. The Bernstein polynomial basis: a centennial retrospective. *Comput. Aided Geom. Design* 29, 6 (2012), 379–419.

[5] Pedro Gonnet, Stefan Güttel, and Lloyd N. Trefethen. 2013. Robust Padé approximation via SVD. *SIAM Rev.* 55, 1 (2013), 101–117.

[6] W. B. Gragg. 1972. The Padé table and its relation to certain algorithms of numerical analysis. *SIAM Rev.* 14 (1972), 1–16.

[7] O. L. Ibryaeva and V. M. Adukov. 2013. An algorithm for computing a Padé approximant with minimal degree denominator. *J. Comput. Appl. Math.* 237, 1 (2013), 529–541.

[8] Hiroshi Kai and Matu-Tarow Noda. 2000. Hybrid rational function approximation and its accuracy analysis, In Proceedings of the International Conference on Rational Approximation, ICRA 99 (Antwerp). *Reliab. Comput.* 6, 4, 429–438.

[9] Manfred Minimair. 2008. Basis-independent polynomial division algorithm applied to division in Lagrange and Bernstein basis. In *Computer Mathematics. ASCM 2007.* Lecture Notes in Comput. Sci., Vol. 5081. Springer, Berlin, 72–86.

[10] Kosaku Nagasaka. 2020. Toward the best algorithm for approximate GCD of univariate polynomials. *J. Symbolic Comput.* (2020). Special issue on MICA 2016. (in press).

[11] Yuji Nakatsukasa, Olivier Sète, and Lloyd N. Trefethen. 2018. The AAA algorithm for rational approximation. *SIAM J. Sci. Comput.* 40, 3 (2018), A1494–A1522.

[12] Victor Y. Pan. 2001. Computation of approximate polynomial GCDs and an extension. *Inform. and Comput.* 167, 2 (2001), 71–85.

[13] C. Sanathanan and J. Koerner. 1963. Transfer function synthesis as a ratio of two complex polynomials. *IEEE Trans. Autom. Control* 8, 1 (1963), 56–58.

[14] Jianzhong Sun, Falai Chen, and Yongming Qu. 1998. Approximate common divisors of polynomials and degree reduction for rational curves. *Appl. Math. J. Chinese Univ. Ser. B* 13, 4 (1998), 437–444. A Chinese summary appears in Gaoxiao Yingyong Shuxue Xuebao Ser. A **13** (1998), no. 4, 486.

[15] Joab R. Winkler and Ronald N. Goldman. 2003. The Sylvester resultant matrix for Bernstein polynomials. In *Curve and surface design (Saint-Malo, 2002).* Nashboro Press, Brentwood, TN, 407–416.

[16] Joab R. Winkler and Ning Yang. 2013. Resultant matrices and the computation of the degree of an approximate greatest common divisor of two inexact Bernstein basis polynomials. *Comput. Aided Geom. Design* 30, 4 (2013), 410–429.

[17] Ning Yang. 2013. *Structured matrix methods for computations on Bernstein basis polynomials.* Ph.D. Dissertation. University of Sheffield, England.

[18] Zhonggang Zeng. 2011. The numerical greatest common divisor of univariate polynomials. In *Randomization, relaxation, and complexity in polynomial equation solving.* Contemp. Math., Vol. 556. Amer. Math. Soc., Providence, RI, 187–217.

Our preliminary implementations are available: https://wwwmain.h.kobe-u.ac.jp/~nagasaka/research/snap/issac20.nb

A Divide-and-conquer Algorithm for Computing Gröbner Bases of Syzygies in Finite Dimension

Simone Naldi
Univ. Limoges, CNRS, XLIM, UMR 7252
F-87000 Limoges, France

Vincent Neiger
Univ. Limoges, CNRS, XLIM, UMR 7252
F-87000 Limoges, France

ABSTRACT

Let $f_1, \ldots, f_m$ be elements in a quotient $\mathcal{R}^n/\mathcal{N}$ which has finite dimension as a $\mathbb{K}$-vector space, where $\mathcal{R} = \mathbb{K}[X_1, \ldots, X_r]$ and $\mathcal{N}$ is an $\mathcal{R}$-submodule of $\mathcal{R}^n$. We address the problem of computing a Gröbner basis of the module of syzygies of $(f_1, \ldots, f_m)$, that is, of vectors $(p_1, \ldots, p_m) \in \mathcal{R}^m$ such that $p_1 f_1 + \cdots + p_m f_m = 0$.

An iterative algorithm for this problem was given by Marinari, Möller, and Mora (1993) using a dual representation of $\mathcal{R}^n/\mathcal{N}$ as the kernel of a collection of linear functionals. Following this viewpoint, we design a divide-and-conquer algorithm, which can be interpreted as a generalization to several variables of Beckermann and Labahn's recursive approach for matrix Padé and rational interpolation problems. To highlight the interest of this method, we focus on the specific case of bivariate Padé approximation and show that it improves upon the best known complexity bounds.

KEYWORDS

Syzygies; Gröbner basis; Padé approximation; divide and conquer

ACM Reference Format:
Simone Naldi and Vincent Neiger. 2020. A Divide-and-conquer Algorithm for Computing Gröbner Bases of Syzygies in Finite Dimension. In *International Symposium on Symbolic and Algebraic Computation (ISSAC '20), July 20–23, 2020, Kalamata, Greece.* ACM, New York, NY, USA, 8 pages. https://doi.org/10.1145/3373207.3404059

1 INTRODUCTION

Context. Hereafter, $\mathcal{R} = \mathbb{K}[X_1, \ldots, X_r]$ is the ring of r-variate polynomials over a field $\mathbb{K}$. Given an $\mathcal{R}$-submodule $\mathcal{N} \subset \mathcal{R}^n$ such that $\mathcal{R}^n/\mathcal{N}$ has finite dimension D as a $\mathbb{K}$-vector space, as well as a matrix $\boldsymbol{F} \in \mathcal{R}^{m \times n}$ with rows $\boldsymbol{f}_1, \ldots, \boldsymbol{f}_m \in \mathcal{R}^n$, this paper studies the computation of a Gröbner basis of the module of syzygies

$$\mathrm{Syz}_{\mathcal{N}}(\boldsymbol{F}) = \{\boldsymbol{p} = (p_i)_{1 \le i \le m} \in \mathcal{R}^m \mid \boldsymbol{p}\boldsymbol{F} = \textstyle\sum_{1 \le i \le m} p_i \boldsymbol{f}_i \in \mathcal{N}\},$$

where $\boldsymbol{p}$ is seen as a $1 \times m$ row vector. Note that $\mathcal{R}^m/\mathrm{Syz}_{\mathcal{N}}(\boldsymbol{F})$ also has finite dimension, at most D, as a $\mathbb{K}$-vector space.

Following a path of work pioneered by Marinari, Möller and Mora [1, 25, 27], we focus on a specific situation where $\mathcal{N}$ is described using duality. That is, $\mathcal{N}$ is known through D linear functionals $\varphi_j : \mathcal{R}^n \to \mathbb{K}$ such that $\mathcal{N} = \cap_{1 \le j \le D} \ker(\varphi_j)$. In this context, it is customary to make an assumption equivalent to the following: $\mathcal{N}_i = \cap_{1 \le j \le i} \ker(\varphi_i)$ is an $\mathcal{R}$-module, for $1 \le i \le D$; see e.g. [25, Algo. 2] [16, Eqn. (4.1)] [30, Eqn. (5)] for such assumptions and related algorithms. Namely, this assumption allows one to design iterative algorithms which compute bases of $\mathrm{Syz}_{\mathcal{N}_i}(\boldsymbol{F})$ iteratively for increasing i, until reaching $i = D$ and obtaining the sought basis of $\mathrm{Syz}_{\mathcal{N}}(\boldsymbol{F})$. An efficient such iterative procedure is given in [25], specifically in Algorithm 2 (variant in Section 9 therein); note that it is written for $m = n = 1$ and $\boldsymbol{F} = [1]$, in which case $\mathrm{Syz}_{\mathcal{N}_i}(\boldsymbol{F}) = \mathcal{N}_i$, but directly extends to the case $m \ge 1$ and $\boldsymbol{F} \in \mathcal{R}^{m \times n}$.

Ideal of points and Padé approximation. One particular case of interest is when $\mathcal{N}$ is the vanishing ideal of a given set of points: $n = 1$, and $\mathcal{N}$ is the ideal of all polynomials in $\mathcal{R}$ which vanish at distinct points $\boldsymbol{\alpha}_1, \ldots, \boldsymbol{\alpha}_D \in \mathbb{K}^r$. Here, one takes the linear functionals for evaluation: $\varphi_j : f \in \mathcal{R} \mapsto f(\boldsymbol{\alpha}_j) \in \mathbb{K}$. The question is, given the points, m polynomials as $\boldsymbol{F} \in \mathcal{R}^{m \times 1}$, and a monomial order $\preccurlyeq$, to compute a $\preccurlyeq$-Gröbner basis of the set of vectors $\boldsymbol{p}$ such that $\boldsymbol{p}\boldsymbol{F}$ vanishes at all the points. When $m = 1$ and $\boldsymbol{F} = [1]$, this means computing a $\preccurlyeq$-Gröbner basis of the ideal of the points, as studied in [25, 26].

Another case is that of (multivariate) Padé approximation and its extensions, as studied in [14, 16, 17, 30], as well as in [6] in the context of the computation of multidimensional linear recurrence relations. The basic setting is for $n = 1$, with $\mathcal{N}$ an ideal of the form $\langle X_1^{d_1}, \ldots, X_r^{d_r} \rangle$, and $\boldsymbol{F} = \left[\begin{smallmatrix} f \\ -1 \end{smallmatrix}\right]$ for some given $f \in \mathcal{R}$. Then, elements of $\mathrm{Syz}_{\mathcal{N}}(\boldsymbol{F})$ are vectors $(q, p) \in \mathcal{R}^2$ such that $f = p/q \bmod X_1^{d_1}, \ldots, X_r^{d_r}$. Here, the $D = d_1 \cdots d_r$ linear functionals correspond to the coefficients of multidegree less than $(d_1, \ldots, d_r)$; note that not all orderings of these functionals satisfy the assumption above.

For these two situations, as well as some extensions of them, the fastest known algorithms rely on linear algebra and have a cost bound of $O(mD^2 + rD^3)$ operations in $\mathbb{K}$ [16, 25]; this was recently improved in [28, Thm. 2.13] and [29] to $O(mD^{\omega-1} + rD^{\omega} \log(D))$ where $\omega < 2.38$ is the exponent of matrix multiplication [10, 24].

Based on work in [9, 15], in the specific case of an ideal of points $\mathcal{N}$ and the lexicographic order, Ceria and Mora gave a combinatorial algorithm to compute the $\preccurlyeq_{\mathrm{lex}}$-monomial basis of $\mathcal{R}/\mathcal{N}$, the Cerlienco-Mureddu correspondence, and squarefree separators for the points using $O(rD^2 \log(D))$ operations [8].

The univariate case. This problem has received attention in the case of a single variable ($r = 1$) notably thanks to the numerous applications of matrix rational interpolation and Hermite-Padé approximation, which are the two situations described above. Iterative algorithms were first given for Padé approximation in [18, 34] and then for Hermite-Padé approximation in [2, 4, 33]; the latter can be seen as univariate analogues of [25, Algo. 2] and [16, Algo. 4.7].

ISSAC '20, July 20–23, 2020, Kalamata, Greece

ACM ISBN 978-1-4503-7100-1/20/07…$15.00
https://doi.org/10.1145/3373207.3404059

A breakthrough divide and conquer approach was designed by Beckermann and Labahn in [3, Algo. SPHPS], allowing one to take advantage of univariate polynomial matrix multiplication while previous iterative algorithms only relied on naive linear algebra operations. This led to a line of work [19, 21, 22, 32, 35] which consistently improved the incorporation of fast linear algebra and fast polynomial multiplication in this divide and conquer framework, culminating in cost bounds for rational interpolation and Hermite-Padé approximation which are close asymptotically to the size of the problem (if $\omega = 2$, these cost bounds are quasi-linear in the size of the input). To the best of our knowledge, no similar divide and conquer technique has been developed in multivariate settings prior to this work.

Contribution. We propose a divide and conquer algorithm for the problem of computing a $\preccurlyeq$-Gröbner basis of $\mathrm{Syz}_{\mathcal{N}}(\boldsymbol{F})$ in the multivariate case. This is based on the iterative algorithm [25, Algo. 2], observing that each step of the iteration can be interpreted as a left multiplication by a matrix which has a specific shape, which we call elementary Gröbner basis (see Section 3). The new algorithm reorganizes these matrix products through a divide and conquer strategy, and thus groups several products by elementary Gröbner bases into a single multivariate polynomial matrix multiplication.

Thus, both the existing iterative and the new divide and conquer approaches compute the same elementary Gröbner bases, but unlike the former, our algorithm does not explicitly compute Gröbner bases for all intermediate syzygy modules $\mathrm{Syz}_{\mathcal{N}_i}(\boldsymbol{F})$. By computing less, we expect to achieve better computational complexity. To illustrate this, we specialize our approach to multivariate matrix Padé approximation and derive complexity bounds for this case; we obtain the next result, which is a particular case of Proposition 5.5.

Theorem 1.1. *For $\mathcal{R} = \mathbb{K}[X, Y]$, let $f_1, \ldots, f_m \in \mathcal{R}$, and let $\preccurlyeq$ be a monomial order on $\mathcal{R}$. Then one can compute a minimal $\preccurlyeq$-Gröbner basis of the module of Hermite-Padé approximants*

$$\{(p_1, \ldots, p_m) \in \mathcal{R}^m \mid p_1 f_1 + \cdots + p_m f_m = 0 \bmod \langle X^d, Y^d \rangle\}$$

using $O\tilde{}(m^\omega d^{\omega+2})$ operations in $\mathbb{K}$, where $O\tilde{}(\cdot)$ means that polylogarithmic factors are omitted.

In this case the vector space dimension is $D = d^2$. Thus, as noted above and to the best of our knowledge, the fastest previously known algorithm for this task has a cost of $O\tilde{}(md^{2(\omega-1)} + d^{2\omega})$ operations in $\mathbb{K}$ and does not exploit fast polynomial multiplication.

Perspectives. The base case of our divide and conquer algorithm concerns the case $\mathcal{N} = \ker(\varphi)$ of a single linear functional, detailed in Section 3; we thus work in a vector space $\mathcal{R}^n/\mathcal{N}$ of dimension 1. A natural perspective is to improve the efficiency of our algorithm thanks to a better exploitation of fast linear algebra by grouping several base cases together; using fast linear algebra to accelerate the base case was a key strategy in obtaining efficient univariate algorithms [19, 22]. In the context of Padé approximation, where one can introduce the variables one after another, one could also try to incorporate known algorithms for the univariate case.

One reason why these improvements are not straightforward to do in the multivariate case is that there is no direct generalization of a property at the core of the correctness of univariate algorithms. This property (see [23, Lem. 2.4]) states that if $\boldsymbol{P}_1$ is a $\preccurlyeq_1$-Gröbner basis of $\mathcal{N}_1 \supset \mathcal{N}$ and $\boldsymbol{P}_2$ is a $\preccurlyeq_2$-Gröbner basis of $\mathrm{Syz}_{\mathcal{N}}(\boldsymbol{P}_1)$, then $\boldsymbol{P}_2\boldsymbol{P}_1$ is a $\preccurlyeq_1$-Gröbner basis of $\mathcal{N}$, provided that the order $\preccurlyeq_2$ is well chosen (a Schreyer order for $\boldsymbol{P}_1$ and $\preccurlyeq_1$, see Section 2.4). We give a counterexample to such a property in Example 3.6. It remains open to find a similar general property that would help to design algorithms based on matrix multiplication in the multivariate case.

Another difficulty arises in analyzing the complexity of our divide and conquer scheme in contexts where the number of elements in the sought Gröbner basis is not well controlled, such as rational interpolation. Indeed, this number corresponds to the size of the matrices used in the algorithm, and therefore is directly related to the cost of the matrix multiplication. In fact, the worst-case number of elements depends on the monomial order and is often pessimistic compared to what is observed in a generic situation. Thus, future work involves investigating complexity bounds for generic input and for interesting particular cases other than Padé approximation.

2 PRELIMINARIES

2.1 Notation

Here and hereafter, the coordinate vector with 1 at index i is denoted by $\boldsymbol{e}_i$; its dimension is inferred from the context. A monomial in $\mathcal{R}^m$ is an element of the form $v\boldsymbol{e}_i$ for some $1 \le i \le m$ and some monomial v in $\mathcal{R}$; i is called the support of $v\boldsymbol{e}_i$. We denote by $\mathrm{Mon}(\mathcal{R}^m)$ the set of all monomials in $\mathcal{R}^m$. A term is a monomial multiplied by a nonzero constant from $\mathbb{K}$. The elements of $\mathcal{R}^m$ are $\mathbb{K}$-linear combinations of elements of $\mathrm{Mon}(\mathcal{R}^m)$ and are called polynomials.

Elements in $\mathcal{R}$ are written in regular font (e.g. monomials μ and ν and polynomials f and p), while elements in $\mathcal{R}^m$ are boldfaced (e.g. monomials $\boldsymbol{\mu}$ and $\boldsymbol{\nu}$ and polynomials $\boldsymbol{f}$ and $\boldsymbol{p}$). Vectors or (ordered) lists of polynomials in $\mathcal{R}^m$ are seen as matrices, written in boldfaced capital letters; precisely, $(\boldsymbol{p}_1, \ldots, \boldsymbol{p}_k) \in (\mathcal{R}^m)^k$ is seen as a matrix $\boldsymbol{P} \in \mathcal{R}^{k \times m}$ whose ith row is $\boldsymbol{p}_i$. In particular, in what follows the default orientation is to see an element of $\mathcal{R}^m$ as a *row* vector in $\mathcal{R}^{1\times m}$.

For the sake of completeness, we recall below in Sections 2.2 to 2.4 some classical definitions from commutative algebra concerning submodules of $\mathcal{R}^m$; we assume familiarity with the corresponding notions concerning ideals of $\mathcal{R}$. For a more detailed introduction the reader may refer to [11–13].

2.2 Monomial orders for modules

A monomial order on $\mathcal{R}^m$ is a total order $\preccurlyeq$ on $\mathrm{Mon}(\mathcal{R}^m)$ such that, for $\nu \in \mathrm{Mon}(\mathcal{R})$ and $\boldsymbol{\mu}_1, \boldsymbol{\mu}_2 \in \mathrm{Mon}(\mathcal{R}^m)$ with $\boldsymbol{\mu}_1 \preccurlyeq \boldsymbol{\mu}_2$, one has $\boldsymbol{\mu}_1 \preccurlyeq \nu\boldsymbol{\mu}_1 \preccurlyeq \nu\boldsymbol{\mu}_2$; hereafter $\boldsymbol{\mu}_1 \prec \boldsymbol{\mu}_2$ means that $\boldsymbol{\mu}_1 \preccurlyeq \boldsymbol{\mu}_2$ and $\boldsymbol{\mu}_1 \neq \boldsymbol{\mu}_2$. For $\boldsymbol{p} \in \mathcal{R}^m$, its $\preccurlyeq$-leading monomial is denoted by $\mathrm{lm}_{\preccurlyeq}(\boldsymbol{p})$ and is the largest of its monomials with respect to the order $\preccurlyeq$ (we take the convention $\mathrm{lm}_{\preccurlyeq}(\boldsymbol{0}) = \boldsymbol{0}$ for $\boldsymbol{0} \in \mathcal{R}^m$ the zero element). We extend this notation to collections of polynomials $\mathcal{P} \subset \mathcal{R}^m$ with $\mathrm{lm}_{\preccurlyeq}(\mathcal{P}) = \{\mathrm{lm}_{\preccurlyeq}(\boldsymbol{p}) : \boldsymbol{p} \in \mathcal{P}\}$, and to matrices $\boldsymbol{P} \in \mathcal{R}^{k\times m}$ with $\mathrm{lm}_{\preccurlyeq}(\boldsymbol{P})$ the $k \times m$ matrix whose ith row is the $\preccurlyeq$-leading monomial of the ith row of $\boldsymbol{P}$.

Example 2.1. The usual lexicographic comparison is a monomial order on $\mathbb{K}[X, Y]$: $X^a Y^b \preccurlyeq_{\mathrm{lex}} X^{a'} Y^{b'}$ if and only if $a < a'$ or ($a = a'$ and $b < b'$). It can be used to define a monomial order on

$\mathbb{K}[X,Y]^2$, called the term-over-position lexicographic order: for μ, ν in $\mathrm{Mon}(\mathbb{K}[X,Y])$ and i, j in $\{1,2\}$, $\mu \boldsymbol{e}_i \preccurlyeq^{\mathrm{top}}_{\mathrm{lex}} \nu \boldsymbol{e}_j$ if and only if $\mu \preccurlyeq_{\mathrm{lex}} \nu$ or $(\mu = \nu$ and $i < j)$.

We refer to [11, Sec. 1.§2 and 5.§2] for other classical monomial orders, such as the degree reverse lexicographical order on $\mathcal{R}$, and the construction of term-over-position and position-over-term orders on $\mathcal{R}^m$ from monomial orders on $\mathcal{R}$.

A monomial order $\preccurlyeq$ on $\mathcal{R}^m$ induces a monomial order $\preccurlyeq_i$ on $\mathcal{R}$ for each $1 \le i \le m$, by restricting to the ith coordinate: for $\nu_1, \nu_2 \in \mathrm{Mon}(\mathcal{R})$, $\nu_1 \preccurlyeq_i \nu_2$ if and only if $\nu_1 \boldsymbol{e}_i \preccurlyeq \nu_2 \boldsymbol{e}_i$. In particular, $\mathrm{lm}_{\preccurlyeq}(q\boldsymbol{p})$ is a multiple of $\mathrm{lm}_{\preccurlyeq}(\boldsymbol{p})$ for $q \in \mathcal{R}$ and $\boldsymbol{p} \in \mathcal{R}^m$:

Lemma 2.2. *Let $\bar{\imath}$ be the support of $\mathrm{lm}_{\preccurlyeq}(\boldsymbol{p})$. Then $\mathrm{lm}_{\preccurlyeq}(q\boldsymbol{p}) = \mathrm{lm}_{\preccurlyeq_{\bar{\imath}}}(q)\mathrm{lm}_{\preccurlyeq}(\boldsymbol{p})$.*

Proof. Write $q = \sum_\ell \nu_\ell$ and $\boldsymbol{p} = \sum_{i,j} \mu_{ij} \boldsymbol{e}_i$ for terms μ_{ij}, ν_ℓ in $\mathcal{R}$. Then $q\boldsymbol{p} = \sum_{\ell,i,j} \nu_\ell \mu_{ij} \boldsymbol{e}_i$, i.e. the terms of $q\boldsymbol{p}$ are all those of the form $\nu_\ell \mu_{ij} \boldsymbol{e}_i$. Now let $\bar{\ell}$ and $\bar{\jmath}$ be such that $\mathrm{lm}_{\preccurlyeq_{\bar{\imath}}}(q) = \nu_{\bar{\ell}}$ and $\mathrm{lm}_{\preccurlyeq}(\boldsymbol{p}) = \mu_{\bar{\imath}\bar{\jmath}} \boldsymbol{e}_{\bar{\imath}}$. Then $\nu_\ell \prec_{\bar{\imath}} \nu_{\bar{\ell}}$ for all $\ell \neq \bar{\ell}$, which implies that $\nu_\ell \mu_{\bar{\imath}\bar{\jmath}} \prec_{\bar{\imath}} \nu_{\bar{\ell}} \mu_{\bar{\imath}\bar{\jmath}}$ and thus, by definition of $\preccurlyeq_{\bar{\imath}}$, that $\nu_\ell \mu_{\bar{\imath}\bar{\jmath}} \boldsymbol{e}_{\bar{\imath}} \prec \nu_{\bar{\ell}} \mu_{\bar{\imath}\bar{\jmath}} \boldsymbol{e}_{\bar{\imath}}$. On the other hand, $\mu_{ij} \boldsymbol{e}_i \prec \mu_{\bar{\imath}\bar{\jmath}} \boldsymbol{e}_{\bar{\imath}}$ holds for all $(i,j) \neq (\bar{\imath}, \bar{\jmath})$, hence $\nu_\ell \mu_{ij} \boldsymbol{e}_i \prec \nu_\ell \mu_{\bar{\imath}\bar{\jmath}} \boldsymbol{e}_{\bar{\imath}}$. Therefore we obtain $\nu_\ell \mu_{ij} \boldsymbol{e}_i \preccurlyeq \nu_{\bar{\ell}} \mu_{\bar{\imath}\bar{\jmath}} \boldsymbol{e}_{\bar{\imath}}$ for all (i,j,ℓ), with equality only if $(i,j,\ell) = (\bar{\imath}, \bar{\jmath}, \bar{\ell})$. This proves that $\mathrm{lm}_{\preccurlyeq}(q\boldsymbol{p}) = \nu_{\bar{\ell}} \mu_{\bar{\imath}\bar{\jmath}} \boldsymbol{e}_{\bar{\imath}} = \mathrm{lm}_{\preccurlyeq_{\bar{\imath}}}(q)\mathrm{lm}_{\preccurlyeq}(\boldsymbol{p})$. □

2.3 Gröbner bases

As a consequence of Hilbert's Basis Theorem, any $\mathcal{R}$-submodule of $\mathcal{R}^m$ is finitely generated [13, Prop. 1.4]. For a (possibly infinite) collection of polynomials $\mathcal{P} \subset \mathcal{R}^m$, we denote by $\langle \mathcal{P} \rangle$ the $\mathcal{R}$-submodule of $\mathcal{R}^m$ generated by the elements of $\mathcal{P}$. Similarly, for a matrix $\boldsymbol{P}$ in $\mathcal{R}^{k \times m}$, $\langle \boldsymbol{P} \rangle$ stands for the $\mathcal{R}$-submodule of $\mathcal{R}^m$ generated by its rows, that is, $\langle \boldsymbol{P} \rangle = \{\boldsymbol{q}\boldsymbol{P} \mid \boldsymbol{q} \in \mathcal{R}^k\}$.

For a given submodule $\mathcal{M} \subset \mathcal{R}^m$, the $\preccurlyeq$-leading module of $\mathcal{M}$ is the module $\langle \mathrm{lm}_{\preccurlyeq}(\mathcal{M}) \rangle$ generated by the leading monomials of the elements of $\mathcal{M}$. Then, a matrix $\boldsymbol{P}$ in $\mathcal{R}^{k \times m}$ whose rows are in $\mathcal{M}$ is said to be a $\preccurlyeq$-Gröbner basis of $\mathcal{M}$ if

$$\langle \mathrm{lm}_{\preccurlyeq}(\mathcal{M}) \rangle = \langle \mathrm{lm}_{\preccurlyeq}(\boldsymbol{P}) \rangle.$$

In this case we have $\langle \boldsymbol{P} \rangle = \mathcal{M}$ (see [11, Ch.5, Prop.2.7]), hence we will often omit the reference to the module $\mathcal{M}$ and just say that $\boldsymbol{P}$ is a $\preccurlyeq$-Gröbner basis.

A $\preccurlyeq$-Gröbner basis $\boldsymbol{P}$, whose rows are $(\boldsymbol{p}_1, \ldots, \boldsymbol{p}_k)$, is said to be minimal if $\mathrm{lm}_{\preccurlyeq}(\boldsymbol{p}_i)$ is not divisible by $\mathrm{lm}_{\preccurlyeq}(\boldsymbol{p}_j)$, for any $j \neq i$. It is said to be reduced if it is minimal and, for all $1 \le i \le k$, $\mathrm{lm}_{\preccurlyeq}(\boldsymbol{p}_i)$ is monic and none of the terms of $\boldsymbol{p}_i$ is divisible by any of $\{\mathrm{lm}_{\preccurlyeq}(\boldsymbol{p}_j) \mid j \neq i\}$. Given a monomial order $\preccurlyeq$ and an $\mathcal{R}$-submodule $\mathcal{M} \subset \mathcal{R}^m$, there is a reduced $\preccurlyeq$-Gröbner basis of $\mathcal{M}$ and it is unique (up to permutation of its elements) [13, Sec. 15.2].

Example 2.3. The syzygy module

$$\mathcal{M} = \{(p_1, p_2) \in \mathbb{K}[X,Y]^2 \mid p_1 - p_2 \in \langle X, Y \rangle\} = \mathrm{Syz}_{\langle X,Y \rangle}([\begin{smallmatrix} 1 \\ -1 \end{smallmatrix}])$$

is generated by $(X\boldsymbol{e}_1, Y\boldsymbol{e}_1, \boldsymbol{e}_1 + \boldsymbol{e}_2)$, that is, by the rows of

$$\boldsymbol{P} = \begin{bmatrix} X & 0 \\ Y & 0 \\ 1 & 1 \end{bmatrix} \in \mathbb{K}[X,Y]^{3\times 2}.$$

Furthermore, $\boldsymbol{P}$ is the reduced $\preccurlyeq^{\mathrm{top}}_{\mathrm{lex}}$-Gröbner basis of $\mathcal{M}$.

2.4 Schreyer orders

In the context of the computation of bases of syzygies it is generally beneficial to use a specific construction of monomial orders, as first highlighted by Schreyer [20, 31] (see also [13, Th. 15.10] and [5]).

In the univariate case, the notion of shifted degree plays the same role as Schreyer orders and is ubiquitous in the computation of bases of modules of syzygies [19, 21, 35]; an equivalent notion of defects was also used earlier for M-Padé and Hermite-Padé approximation algorithms [2, 3]. Specifically, this provides a monomial order on $\mathcal{R}^k$ constructed from a monomial order $\preccurlyeq$ on $\mathcal{R}^m$ and from the leading monomials of a $\preccurlyeq$-Gröbner basis in $\mathcal{R}^m$ of cardinality k.

Definition 2.4. Let $\preccurlyeq$ be a monomial order on $\mathcal{R}^m$, and let $\boldsymbol{L} = (\boldsymbol{\mu}_1, \ldots, \boldsymbol{\mu}_k)$ be a list of monomials of $\mathcal{R}^m$. A *Schreyer order for $\preccurlyeq$ and $\boldsymbol{L}$* is any monomial order on $\mathcal{R}^k$, denoted by $\preccurlyeq_{\boldsymbol{L}}$, such that for $\nu_1 \boldsymbol{e}_i, \nu_2 \boldsymbol{e}_j \in \mathrm{Mon}(\mathcal{R}^k)$, if $\nu_1 \boldsymbol{\mu}_i \prec \nu_2 \boldsymbol{\mu}_j$ then $\nu_1 \boldsymbol{e}_i \preccurlyeq_{\boldsymbol{L}} \nu_2 \boldsymbol{e}_j$.

As noted above, this notion is often used with $\boldsymbol{L} = \mathrm{lm}_{\preccurlyeq}(\boldsymbol{P})$ for a list of polynomials $\boldsymbol{P} \in \mathcal{R}^{k \times m}$, which is typically a $\preccurlyeq$-Gröbner basis.

Remark that Definition 2.4 uses a strict inequality, and implies that if $\nu_1 \boldsymbol{e}_i \preccurlyeq_{\boldsymbol{L}} \nu_2 \boldsymbol{e}_j$, then $\nu_1 \boldsymbol{\mu}_i \prec \nu_2 \boldsymbol{\mu}_j$ or $\nu_1 \boldsymbol{\mu}_i = \nu_2 \boldsymbol{\mu}_j$. In particular, for $\nu_1 = \nu_2 = 1$ and assuming $\boldsymbol{\mu}_i \neq \boldsymbol{\mu}_j$ for all $i \neq j$ (for instance, if $\boldsymbol{L} = \mathrm{lm}_{\preccurlyeq}(\boldsymbol{P})$ for a minimal $\preccurlyeq$-Gröbner basis $\boldsymbol{P}$), then $\boldsymbol{e}_i \preccurlyeq_{\boldsymbol{L}} \boldsymbol{e}_j$ if and only if $\boldsymbol{\mu}_i \prec \boldsymbol{\mu}_j$.

Furthermore, for every $\preccurlyeq$ and $\boldsymbol{L}$, a corresponding Schreyer order exists and can be constructed explicitly: for example, $\nu_1 \boldsymbol{e}_i \preccurlyeq_{\boldsymbol{L}} \nu_2 \boldsymbol{e}_j$ if and only if

$$\nu_1 \boldsymbol{\mu}_i \prec \nu_2 \boldsymbol{\mu}_j \text{ or } (\nu_1 \boldsymbol{\mu}_i = \nu_2 \boldsymbol{\mu}_j \text{ and } i < j).$$

This specific Schreyer order is the one used in the algorithms in this paper, where we write

$$\preccurlyeq_{\boldsymbol{L}} \leftarrow \textsc{SchreyerOrder}(\preccurlyeq, \boldsymbol{L})$$

to mean that the algorithm constructs it from $\preccurlyeq$ and $\boldsymbol{L}$.

3 BASE CASE OF THE DIVIDE AND CONQUER SCHEME

In this section we present the base case of our main algorithm. It constructs Gröbner bases for syzygies modulo the kernel of a single linear functional, which we call elementary Gröbner bases and describe in Section 3.1. Further in Section 3.2 we state properties that are useful to prove the correctness of the base case algorithm given in Section 3.3. Precisely, this correctness is written having in mind the design of an algorithm handling several functionals iteratively by repeating this basic procedure and multiplying the elementary bases together.

3.1 Elementary Gröbner basis

If $\mathcal{I} \subset \mathcal{R}$ is an ideal such that $\mathcal{R}/\mathcal{I}$ has dimension 1 as a $\mathbb{K}$-vector space, then $\mathcal{I}$ is maximal: it is of the form $\langle X_1 - \alpha_1, \ldots, X_r - \alpha_r \rangle$ for some point $(\alpha_1, \ldots, \alpha_r) \in \mathbb{K}^r$, which directly yields the reduced Gröbner basis of $\mathcal{I}$, for any monomial order. In this paper, we will make use of a similar property for submodules of $\mathcal{R}^m$; such submodules have Gröbner bases of the form

$$\boldsymbol{E} = \begin{bmatrix} \boldsymbol{I}_{\pi-1} & \boldsymbol{\lambda}_1 & \\ & \boldsymbol{X} - \boldsymbol{\alpha} & \\ & \boldsymbol{\lambda}_2 & \boldsymbol{I}_{m-\pi} \end{bmatrix} \in \mathcal{R}^{(m+r-1)\times m}, \tag{1}$$

for the vector of variables $X = [X_1 \ \cdots \ X_r]^T$ and vectors of values $\boldsymbol{\alpha} = [\alpha_1 \ \cdots \ \alpha_r]^T \in \mathbb{K}^{r\times 1}$, $\boldsymbol{\lambda}_1 = [\lambda_1 \ \cdots \ \lambda_{\pi-1}]^T \in \mathbb{K}^{(\pi-1)\times 1}$, and $\boldsymbol{\lambda}_2 = [\lambda_{\pi+1} \ \cdots \ \lambda_m]^T \in \mathbb{K}^{(m-\pi)\times 1}$. In what follows, such matrices are called elementary Gröbner bases.

Theorem 3.1. *Let $\mathcal{M}$ be an $\mathcal{R}$-submodule of $\mathcal{R}^m$ such that $\mathcal{R}^m/\mathcal{M}$ has dimension 1 as a $\mathbb{K}$-vector space, then for any monomial order $\preccurlyeq$ on $\mathcal{R}^m$, the reduced $\preccurlyeq$-Gröbner basis $\boldsymbol{E}$ of $\mathcal{M}$ is as in Eq. (1) with $\lambda_i = 0$ if $\boldsymbol{e}_i \prec \boldsymbol{e}_\pi$ for all $i \neq \pi$. Conversely, any matrix $\boldsymbol{E}$ as in Eq. (1) defines a submodule $\mathcal{M} = \langle \boldsymbol{E} \rangle$ such that $\mathcal{R}^m/\mathcal{M}$ has dimension 1 as a $\mathbb{K}$-vector space, and $\boldsymbol{E}$ is a reduced $\preccurlyeq$-Gröbner basis for any monomial order $\preccurlyeq$ such that $\lambda_i = 0$ if $\boldsymbol{e}_i \prec \boldsymbol{e}_\pi$ for all $i \neq \pi$.*

Proof. By [13, Thm. 15.3], a basis of $\mathcal{R}^m/\mathcal{M}$ as a $\mathbb{K}$-vector space is given by the monomials not in $\mathrm{lm}_\preccurlyeq(\mathcal{M})$; since the dimension of $\mathcal{R}^m/\mathcal{M}$ as a $\mathbb{K}$-vector space is 1, there exists a unique monomial which is not in $\mathrm{lm}_\preccurlyeq(\mathcal{M})$. Thus there is a unique $\pi \in \{1, \ldots, m\}$ such that

$$\mathrm{lm}_\preccurlyeq(\boldsymbol{E}) = (\boldsymbol{e}_1, \ldots, \boldsymbol{e}_{\pi-1}, X_1\boldsymbol{e}_\pi, \ldots, X_r\boldsymbol{e}_\pi, \boldsymbol{e}_{\pi+1}, \ldots, \boldsymbol{e}_m). \quad (2)$$

By definition of reduced Gröbner bases, the jth polynomial in $\boldsymbol{E}$ is the sum of the jth element of $\mathrm{lm}_\preccurlyeq(\boldsymbol{E})$ and a constant multiple of $\boldsymbol{e}_\pi$; hence $\boldsymbol{E}$ has the form in Eq. (1). In addition, for $i \neq \pi$, the equality $\mathrm{lm}_\preccurlyeq(\boldsymbol{e}_i + \lambda_i \boldsymbol{e}_\pi) = \boldsymbol{e}_i$ implies that $\lambda_i = 0$ whenever $\boldsymbol{e}_i \prec \boldsymbol{e}_\pi$.

For the converse, let $\preccurlyeq$ be such that $\lambda_i = 0$ if $\boldsymbol{e}_i \prec \boldsymbol{e}_\pi$ for all $i \neq \pi$ (such an order exists since there are orders for which $\boldsymbol{e}_\pi$ is the smallest coordinate vector). Then $\mathrm{lm}_\preccurlyeq(\boldsymbol{E})$ is as in Eq. (2); in particular, the monomials in $\langle \mathrm{lm}_\preccurlyeq(\boldsymbol{E}) \rangle$ are precisely $\mathrm{Mon}(\mathcal{R}^m) \setminus \{\boldsymbol{e}_\pi\}$. It follows that either $\boldsymbol{e}_\pi \in \mathrm{lm}_\preccurlyeq(\mathcal{M})$ and $\langle \mathrm{lm}_\preccurlyeq(\mathcal{M}) \rangle = \mathcal{R}^m$, or $\boldsymbol{e}_\pi \notin \mathrm{lm}_\preccurlyeq(\mathcal{M})$ and $\langle \mathrm{lm}_\preccurlyeq(\mathcal{M}) \rangle = \langle \mathrm{lm}_\preccurlyeq(\boldsymbol{E}) \rangle$. In the second case $\boldsymbol{E}$ is a reduced $\preccurlyeq$-Gröbner-basis and $\mathcal{R}^m/\mathcal{M}$ has dimension 1 by [13, Thm. 15.3]. To conclude the proof, we show that $\boldsymbol{e}_\pi \in \mathrm{lm}_\preccurlyeq(\mathcal{M})$ cannot occur; by contradiction, suppose there exists $\boldsymbol{q} \in \mathcal{M}$ such that $\mathrm{lm}_\preccurlyeq(\boldsymbol{q}) = \boldsymbol{e}_\pi$. Since the rows of $\boldsymbol{E}$ generate $\mathcal{M}$, we can write

$$\begin{aligned}\boldsymbol{q} &= (q_1, \ldots, q_{\pi-1}, p_1, \ldots, p_r, q_{\pi+1}, \ldots, q_m)\,\boldsymbol{E} \\ &= \left(q_1, \ldots, q_{\pi-1}, \sum_{i\neq\pi} q_i\lambda_i + \sum_{j=1}^{r}(X_j - \alpha_j)p_j, q_{\pi+1}, \ldots, q_m\right).\end{aligned}$$

For $i \neq \pi$ such that $\boldsymbol{e}_\pi \prec \boldsymbol{e}_i$, any nonzero term of $q_i\boldsymbol{e}_i$ would appear in $\boldsymbol{q}$ and be greater than $\boldsymbol{e}_\pi$, hence $q_i = 0$. Moreover, for $i \neq \pi$ such that $\boldsymbol{e}_i \prec \boldsymbol{e}_\pi$ we have $\lambda_i = 0$. Thus, considering the πth component of $\boldsymbol{q}$ yields the equality

$$1 = \sum_{i\neq\pi} q_i\lambda_i + \sum_{j=1}^{r}(X_j - \alpha_j)p_j = \sum_{j=1}^{r}(X_j - \alpha_j)p_j$$

which is a contradiction since $1 \notin \langle X_1 - \alpha_1, \ldots, X_r - \alpha_r \rangle$. □

Remark that in the module case ($m \geq 2$) the reduced $\preccurlyeq$-Gröbner basis depends on the order $\preccurlyeq$, more precisely on how the $\boldsymbol{e}_i$'s are ordered by $\preccurlyeq$. For instance, the matrix in Example 2.3 is a reduced $\preccurlyeq$-Gröbner basis for every order such that $\boldsymbol{e}_1 \preccurlyeq \boldsymbol{e}_2$, whereas for orders such that $\boldsymbol{e}_2 \preccurlyeq \boldsymbol{e}_1$ the reduced $\preccurlyeq$-Gröbner basis of the same module is

$$\boldsymbol{E} = \begin{bmatrix} 1 & 1 \\ 0 & X \\ 0 & Y \end{bmatrix} \in \mathbb{K}[X, Y]^{3\times 2}.$$

3.2 Multiplying by elementary Gröbner bases

Let $\preccurlyeq$ be a monomial order on $\mathcal{R}^m$ and let $\boldsymbol{P} = (\boldsymbol{p}_1, \ldots, \boldsymbol{p}_k) \in \mathcal{R}^{k\times m}$ be a $\preccurlyeq$-Gröbner basis. In this section, we show conditions on an elementary Gröbner basis $\boldsymbol{E}$ to ensure that $\boldsymbol{EP}$ is a $\preccurlyeq$-Gröbner basis.

We write $\boldsymbol{L} = (\boldsymbol{\mu}_1, \ldots, \boldsymbol{\mu}_k)$ for $\mathrm{lm}_\preccurlyeq(\boldsymbol{P})$, that is, $\boldsymbol{\mu}_i = \mathrm{lm}_\preccurlyeq(\boldsymbol{p}_i)$ for $1 \leq i \leq k$. Let $\preccurlyeq_{\boldsymbol{L}}$ be a Schreyer order for $\preccurlyeq$ and $\boldsymbol{P}$, and consider a reduced $\preccurlyeq_{\boldsymbol{L}}$-Gröbner basis $\boldsymbol{E} \in \mathcal{R}^{(k+r-1)\times k}$ which has the form in Eq. (1); thus

$$\begin{aligned}\boldsymbol{EP} = (&\boldsymbol{p}_1 + \lambda_1\boldsymbol{p}_\pi, \ldots, \boldsymbol{p}_{\pi-1} + \lambda_{\pi-1}\boldsymbol{p}_\pi, \\ &(X_1 - \alpha_1)\boldsymbol{p}_\pi, \ldots, (X_r - \alpha_r)\boldsymbol{p}_\pi, \\ &\lambda_{\pi+1}\boldsymbol{p}_\pi + \boldsymbol{p}_{\pi+1}, \ldots, \lambda_k\boldsymbol{p}_\pi + \boldsymbol{p}_k)\end{aligned}$$

which is in $\mathcal{R}^{(k+r-1)\times m}$. We will show that, under suitable assumptions, $\boldsymbol{EP}$ is a $\preccurlyeq$-Gröbner basis; the next lemmas use the above notation. We start by describing the leading terms of $\boldsymbol{EP}$.

Lemma 3.2. *If $\boldsymbol{\mu}_i \neq \boldsymbol{\mu}_\pi$ for all $i \neq \pi$, then*

$$\begin{aligned}\mathrm{lm}_\preccurlyeq(\boldsymbol{EP}) &= \mathrm{lm}_{\preccurlyeq_{\boldsymbol{L}}}(\boldsymbol{E})\,\boldsymbol{L} \\ &= (\boldsymbol{\mu}_1, \ldots, \boldsymbol{\mu}_{\pi-1}, X_1\boldsymbol{\mu}_\pi, \ldots, X_r\boldsymbol{\mu}_\pi, \boldsymbol{\mu}_{\pi+1}, \ldots, \boldsymbol{\mu}_k).\end{aligned}$$

Proof. First, $\mathrm{lm}_\preccurlyeq((X_j - \alpha_j)\boldsymbol{p}_\pi) = X_j\boldsymbol{\mu}_\pi$ for $1 \leq j \leq r$. Next we claim that $\mathrm{lm}_\preccurlyeq(\boldsymbol{p}_i + \lambda_i\boldsymbol{p}_\pi) = \boldsymbol{\mu}_i$ for all $i \neq \pi$. If $\lambda_i = 0$, the identity is obvious. If $\lambda_i \neq 0$, then $\boldsymbol{e}_\pi \preccurlyeq_{\boldsymbol{L}} \boldsymbol{e}_i$ (see Section 3.1), and from the definition of a Schreyer order and the assumption $\boldsymbol{\mu}_\pi \neq \boldsymbol{\mu}_i$, we deduce $\boldsymbol{\mu}_\pi \prec \boldsymbol{\mu}_i$ and hence $\mathrm{lm}_\preccurlyeq(\boldsymbol{p}_i + \lambda_i\boldsymbol{p}_\pi) = \boldsymbol{\mu}_i$. □

Next, we characterize the fact that $\boldsymbol{EP}$ generates a submodule which differs from the one generated by $\boldsymbol{P}$.

Lemma 3.3. *If $\boldsymbol{\mu}_i \neq \boldsymbol{\mu}_\pi$ for all $i \neq \pi$, then*

$$\langle \boldsymbol{EP} \rangle \neq \langle \boldsymbol{P} \rangle \ \Leftrightarrow\ \boldsymbol{p}_\pi \notin \langle \boldsymbol{EP} \rangle \ \Leftrightarrow\ \boldsymbol{\mu}_\pi \notin \langle \mathrm{lm}_\preccurlyeq(\langle \boldsymbol{EP} \rangle) \rangle.$$

Proof. First, remark that $\langle \boldsymbol{EP} \rangle = \langle \boldsymbol{P} \rangle \Rightarrow \boldsymbol{p}_\pi \in \langle \boldsymbol{EP} \rangle \Rightarrow \boldsymbol{\mu}_\pi \in \langle \mathrm{lm}_\preccurlyeq(\langle \boldsymbol{EP} \rangle) \rangle$ is obvious; thus, to conclude the proof it remains to show that $\langle \boldsymbol{EP} \rangle = \langle \boldsymbol{P} \rangle \Leftarrow \boldsymbol{\mu}_\pi \in \langle \mathrm{lm}_\preccurlyeq(\langle \boldsymbol{EP} \rangle) \rangle$. Suppose that $\boldsymbol{\mu}_\pi \in \langle \mathrm{lm}_\preccurlyeq(\langle \boldsymbol{EP} \rangle) \rangle$. Then, since $\boldsymbol{\mu}_i \in \langle \mathrm{lm}_\preccurlyeq(\langle \boldsymbol{EP} \rangle) \rangle$ for all $i \neq \pi$ by Lemma 3.2, we have $\mathrm{lm}_\preccurlyeq(\boldsymbol{P}) \subset \langle \mathrm{lm}_\preccurlyeq(\langle \boldsymbol{EP} \rangle) \rangle$, hence $\langle \mathrm{lm}_\preccurlyeq(\boldsymbol{P}) \rangle \subset \langle \mathrm{lm}_\preccurlyeq(\langle \boldsymbol{EP} \rangle) \rangle$. Furthermore, recall that $\langle \mathrm{lm}_\preccurlyeq(\boldsymbol{P}) \rangle = \langle \mathrm{lm}_\preccurlyeq(\langle \boldsymbol{P} \rangle) \rangle$ since $\boldsymbol{P}$ is a $\preccurlyeq$-Gröbner basis, and that $\langle \mathrm{lm}_\preccurlyeq(\langle \boldsymbol{EP} \rangle) \rangle \subset \langle \mathrm{lm}_\preccurlyeq(\langle \boldsymbol{P} \rangle) \rangle$ since $\langle \boldsymbol{EP} \rangle \subset \langle \boldsymbol{P} \rangle$: we obtain $\langle \mathrm{lm}_\preccurlyeq(\langle \boldsymbol{P} \rangle) \rangle = \langle \mathrm{lm}_\preccurlyeq(\langle \boldsymbol{EP} \rangle) \rangle$. Then, [13, Lemma 15.5] shows that $\langle \boldsymbol{EP} \rangle = \langle \boldsymbol{P} \rangle$. □

For example, if $\boldsymbol{P}$ is a *minimal* $\preccurlyeq$-Gröbner basis, then the assumption in the previous lemma is satisfied. Example 3.6 below exhibits a case where $\boldsymbol{P}$ is a minimal $\preccurlyeq$-Gröbner basis and $\boldsymbol{p}_\pi$ does belong to $\langle \boldsymbol{EP} \rangle$. In that case, $\langle \boldsymbol{EP} \rangle = \langle \boldsymbol{P} \rangle$ and $\boldsymbol{EP}$ is not a Gröbner basis since $\boldsymbol{\mu}_\pi$ is in $\langle \mathrm{lm}_\preccurlyeq(\langle \boldsymbol{EP} \rangle) \rangle$ but not in $\langle \mathrm{lm}_\preccurlyeq(\boldsymbol{EP}) \rangle$.

Lemma 3.4. *If $\boldsymbol{\mu}_i \neq \boldsymbol{\mu}_\pi$ for all $i \neq \pi$ and $\langle \boldsymbol{EP} \rangle \neq \langle \boldsymbol{P} \rangle$, then $\boldsymbol{EP}$ is a $\preccurlyeq$-Gröbner basis.*

Proof. Suppose by contradiction that $\boldsymbol{EP}$ is not a $\preccurlyeq$-Gröbner basis. Then there exists a nonzero $\boldsymbol{h} \in \langle \boldsymbol{EP} \rangle$ such that $\mathrm{lm}_\preccurlyeq(\boldsymbol{h}) \notin \langle \mathrm{lm}_\preccurlyeq(\boldsymbol{EP}) \rangle$, that is, by Lemma 3.2, $\mathrm{lm}_\preccurlyeq(\boldsymbol{h})$ is not divisible by any of the elements $\boldsymbol{\mu}_i$ for $i \neq \pi$ and $X_j\boldsymbol{\mu}_\pi$ for $1 \leq j \leq r$. On the other hand, $\mathrm{lm}_\preccurlyeq(\boldsymbol{h})$ is in $\langle \mathrm{lm}_\preccurlyeq(\langle \boldsymbol{EP} \rangle) \rangle$ and therefore in $\langle \mathrm{lm}_\preccurlyeq(\boldsymbol{P}) \rangle$, hence $\mathrm{lm}_\preccurlyeq(\boldsymbol{h})$ is divisible by at least one $\boldsymbol{\mu}_i$, $1 \leq i \leq k$. These divisibility constraints lead to $\mathrm{lm}_\preccurlyeq(\boldsymbol{h}) = \boldsymbol{\mu}_\pi$, which implies $\boldsymbol{\mu}_\pi \in \langle \mathrm{lm}_\preccurlyeq(\langle \boldsymbol{EP} \rangle) \rangle$. From Lemma 3.3 one deduces $\langle \boldsymbol{EP} \rangle = \langle \boldsymbol{P} \rangle$, which is absurd. □

Corollary 3.5. *Assume that $\langle EP\rangle \neq \langle P\rangle$ and that P is a minimal $\preccurlyeq$-Gröbner basis. Let $j_1 < \cdots < j_\ell$ be the indices $j \in \{1,\dots,r\}$ such that $X_j\boldsymbol{\mu}_\pi \notin \langle \boldsymbol{\mu}_i, i \neq \pi\rangle$. Then, the submatrix*

$$Q = \begin{bmatrix} I_{\pi-1} & & \\ & X_{j_1} - \alpha_{j_1} & \\ & \vdots & \\ & X_{j_\ell} - \alpha_{j_\ell} & \\ & \boldsymbol{\lambda}_2 & I_{m-\pi} \end{bmatrix} \in \mathcal{R}^{(k+\ell-1)\times k} \tag{3}$$

of E is such that QP is a minimal $\preccurlyeq$-Gröbner basis of $\langle EP\rangle$.

Proof. Since P is minimal, $\boldsymbol{\mu}_i \neq \boldsymbol{\mu}_\pi$ for all $i \neq \pi$; then Lemma 3.4 ensures that EP is a $\preccurlyeq$-Gröbner basis and Lemma 3.2 gives

$$\mathrm{lm}_\preccurlyeq(QP) = (\boldsymbol{\mu}_1,\dots,\boldsymbol{\mu}_{\pi-1}, X_{j_1}\boldsymbol{\mu}_\pi,\dots,X_{j_\ell}\boldsymbol{\mu}_\pi,\boldsymbol{\mu}_{\pi+1},\dots,\boldsymbol{\mu}_k).$$

By construction of $j_1,\dots,j_\ell$, one has $\langle \mathrm{lm}_\preccurlyeq(QP)\rangle = \langle \mathrm{lm}_\preccurlyeq(EP)\rangle$, which implies

$$\begin{aligned}\langle \mathrm{lm}_\preccurlyeq(\langle EP\rangle)\rangle = \langle \mathrm{lm}_\preccurlyeq(EP)\rangle &= \langle \mathrm{lm}_\preccurlyeq(QP)\rangle \\ &\subset \langle \mathrm{lm}_\preccurlyeq(\langle QP\rangle)\rangle \subset \langle \mathrm{lm}_\preccurlyeq(\langle EP\rangle)\rangle.\end{aligned}$$

Hence $\langle \mathrm{lm}_\preccurlyeq(QP)\rangle = \langle \mathrm{lm}_\preccurlyeq(\langle QP\rangle)\rangle$, and QP is a minimal $\preccurlyeq$-Gröbner basis. We conclude using [13, Lem. 15.5], which shows that $\langle QP\rangle \subset \langle EP\rangle$ and $\langle \mathrm{lm}_\preccurlyeq(\langle QP\rangle)\rangle = \langle \mathrm{lm}_\preccurlyeq(\langle EP\rangle)\rangle$ imply $\langle QP\rangle = \langle EP\rangle$. □

Example 3.6. Consider the case $\mathcal{R} = \mathbb{K}[X,Y]$ and $m = 1$. Let $P = \left[\begin{smallmatrix} X \\ Y+1 \end{smallmatrix}\right] \in \mathcal{R}^{2\times 1}$, which is the reduced $\preccurlyeq_1$-Gröbner basis of $\langle X, Y+1\rangle$ for any monomial order $\preccurlyeq_1$ on $\mathrm{Mon}(\mathcal{R})$. Let also $E \in \mathcal{R}^{3\times 2}$ whose rows are (Xe_1, Ye_1, e_2); according to Theorem 3.1, E is a reduced $\preccurlyeq_2$-Gröbner basis for any monomial order $\preccurlyeq_2$ on $\mathrm{Mon}(\mathcal{R}^2)$. Now, the product $EP \in \mathcal{R}^{3\times 1}$ has entries X^2, XY, and $Y+1$. Thus, $\langle \mathrm{lm}_{\preccurlyeq_3}(EP)\rangle = \langle X^2, XY, Y\rangle = \langle X^2, Y\rangle$ for any monomial order $\preccurlyeq_3$ on $\mathrm{Mon}(\mathcal{R})$. On the other hand, $\langle EP\rangle$ contains $X = X(Y+1) - XY$, hence $\langle \mathrm{lm}_{\preccurlyeq_3}(EP)\rangle \neq \langle \mathrm{lm}_{\preccurlyeq_3}(\langle EP\rangle)\rangle$, which means that EP is not a $\preccurlyeq_3$-Gröbner basis.

3.3 Algorithm

We now describe Algorithm Syzygy_BaseCase, which will serve as the base case of the divide and conquer scheme.

Theorem 3.7. *Let $\mathcal{N} \subset \mathcal{R}^n$ be an $\mathcal{R}$-submodule, let $F \in \mathcal{R}^{m\times n}$, and let $P \in \mathcal{R}^{k\times m}$ be a minimal $\preccurlyeq$-Gröbner basis of $\mathrm{Syz}_{\mathcal{N}}(F)$ for some monomial order $\preccurlyeq$ on $\mathcal{R}^m$. Assume that the input of Algorithm 1 is such that $\ker(\varphi) \cap \mathcal{N}$ is an $\mathcal{R}$-module, $G = PF$, and $\mathrm{lm}_\preccurlyeq(P) = (\boldsymbol{\mu}_1,\dots,\boldsymbol{\mu}_k)$. Then Algorithm 1 returns (Q, L) such that QP is a minimal $\preccurlyeq$-Gröbner basis of $\mathrm{Syz}_{\ker(\varphi)\cap\mathcal{N}}(F)$ and $L = \mathrm{lm}_\preccurlyeq(QP)$.*

Proof. If $(\varphi(g_1),\dots,\varphi(g_k)) = (0,\dots,0)$, then Algorithm 1 stops at Line 2 and returns $Q = I_k$ and K. Thus $QP = P$, hence by assumption $L = K = \mathrm{lm}_\preccurlyeq(P) = \mathrm{lm}_\preccurlyeq(QP)$, and QP is a minimal $\preccurlyeq$-Gröbner basis of $\mathrm{Syz}_{\mathcal{N}}(F)$; besides, the identity $\mathrm{Syz}_{\mathcal{N}}(F) = \mathrm{Syz}_{\ker(\varphi)\cap\mathcal{N}}(F)$ is easily deduced from $(\varphi(g_1),\dots,\varphi(g_k)) = (0,\dots,0)$.

In the rest of the proof, assume $(\varphi(g_1),\dots,\varphi(g_k)) \neq (0,\dots,0)$. Define $E \in \mathcal{R}^{(k+r-1)\times k}$ as in Eq. (1) with π and λ_i as in Algorithm 1 and $\alpha_j = \varphi(X_j g_\pi)/v_\pi$ for $1 \le j \le r$; in particular, Q computed at Line 8 is formed by a subset of the rows of E.

First, E is a $\preccurlyeq_K$-Gröbner basis according to Theorem 3.1, since by definition of π and λ_i one gets the implications $e_i \preccurlyeq_K e_\pi \Rightarrow v_i = 0 \Rightarrow \lambda_i = 0$, for $i \neq \pi$.

Algorithm 1 Syzygy_BaseCase$(\varphi, G, \preccurlyeq, L)$

Input:
- a linear functional $\varphi : \mathcal{R}^n \to \mathbb{K}$,
- a matrix G in $\mathcal{R}^{k\times n}$ with rows $g_1,\dots,g_k \in \mathcal{R}^n$,
- a monomial order $\preccurlyeq$ on $\mathcal{R}^m$,
- a list $K = (\boldsymbol{\mu}_1,\dots,\boldsymbol{\mu}_k)$ of elements of $\mathrm{Mon}(\mathcal{R}^m)$.

Output:
- a matrix Q in $\mathcal{R}^{(k+\ell-1)\times k}$ for some $\ell \in \{0,\dots,r\}$,
- a list L of $k+\ell-1$ elements of $\mathrm{Mon}(\mathcal{R}^k)$.

1: $(v_1,\dots,v_k) \leftarrow (\varphi(g_1),\dots,\varphi(g_k)) \in \mathbb{K}^k$
2: **if** $(v_1,\dots,v_k) = (0,\dots,0)$ **then return** (I_k, K)
3: $\preccurlyeq_K \leftarrow$ SchreyerOrder$(\preccurlyeq, K)$
4: $\pi \leftarrow \arg\min_{\preccurlyeq_K}\{e_i \mid 1 \le i \le k, v_i \neq 0\}$ ▷ *the index i such that $v_i \neq 0$ which minimizes e_i with respect to $\preccurlyeq_K$*
5: $\{j_1 < \cdots < j_\ell\} \leftarrow \{j \in \{1,\dots,r\} \mid X_j\boldsymbol{\mu}_\pi \notin \langle \boldsymbol{\mu}_i, i \neq \pi\rangle\}$
6: $\alpha_{j_s} \leftarrow \varphi(X_{j_s} g_\pi)/v_\pi$ for $1 \le s \le \ell$
7: $\lambda_i \leftarrow -v_i/v_\pi$ for $1 \le i < \pi$ and $\pi < i \le k$
8: $Q \leftarrow$ matrix in $\mathcal{R}^{(k+\ell-1)\times k}$ as in Eq. (3)
9: $L \leftarrow (\boldsymbol{\mu}_1,\dots,\boldsymbol{\mu}_{\pi-1}, X_{j_1}\boldsymbol{\mu}_\pi,\dots,X_{j_\ell}\boldsymbol{\mu}_\pi,\boldsymbol{\mu}_{\pi+1},\dots,\boldsymbol{\mu}_k)$
10: **return** (Q, L)

Next, we claim that $\langle E\rangle = \mathrm{Syz}_{\ker(\varphi)\cap\mathcal{N}}(G)$. Indeed, the rows of PF are in $\mathcal{N}$, and thus so are those of $EG = EPF$. Moreover, by choice of π and λ_i the rows of EG are in $\ker(\varphi)$, since for $i \neq \pi$ one has $\varphi((p_i + \lambda_i p_\pi)F) = \varphi(g_i + \lambda_i g_\pi) = v_i + \lambda_i v_\pi = 0$ and for $1 \le j \le r$ one has $\varphi((X_j - \alpha_j)p_\pi F) = \varphi((X_j - \alpha_j)g_\pi) = \varphi(X_j g_\pi) - \alpha_j v_\pi = 0$. Therefore the rows of EG are in $\ker(\varphi)\cap\mathcal{N}$, that is, $\langle E\rangle \subset \mathrm{Syz}_{\ker(\varphi)\cap\mathcal{N}}(G)$. To prove the reverse inclusion, recall from Theorem 3.1 that $\langle E\rangle$ has codimension 1 in $\mathcal{R}^k$ and hence $\mathrm{Syz}_{\ker(\varphi)\cap\mathcal{N}}(G)$ is either $\langle E\rangle$ or $\mathcal{R}^k$. Since

$$0 \neq v_\pi = \varphi(g_\pi) = \varphi(p_\pi F) = \varphi(e_\pi PF) = \varphi(e_\pi G)$$

one has that $e_\pi \notin \mathrm{Syz}_{\ker(\varphi)\cap\mathcal{N}}(G)$, hence $\mathrm{Syz}_{\ker(\varphi)\cap\mathcal{N}}(G) = \langle E\rangle$.

It follows that $\langle EP\rangle = \mathrm{Syz}_{\ker(\varphi)\cap\mathcal{N}}(F)$. Indeed, the rows of EPF are in $\ker(\varphi)\cap\mathcal{N}$ as noted above, and thus $\langle EP\rangle \subset \mathrm{Syz}_{\ker(\varphi)\cap\mathcal{N}}(F)$. Now let $p \in \mathrm{Syz}_{\ker(\varphi)\cap\mathcal{N}}(F)$; thus in particular $p \in \mathrm{Syz}_{\mathcal{N}}(F)$, and $p = qP$ for some $q \in \mathcal{R}^k$. Then $pF = qPF = qG \in \ker(\varphi)\cap\mathcal{N}$, hence $q \in \mathrm{Syz}_{\ker(\varphi)\cap\mathcal{N}}(G) = \langle E\rangle$, and therefore $p \in \langle EP\rangle$.

Now, $\varphi(p_\pi F) \neq 0$ implies $p_\pi \notin \mathrm{Syz}_{\ker(\varphi)\cap\mathcal{N}}(F) = \langle EP\rangle$. Thus Lemma 3.3 ensures $\langle EP\rangle \neq \langle P\rangle$, and finally Corollary 3.5 states that QP is a minimal $\preccurlyeq$-Gröbner basis of $\langle EP\rangle = \mathrm{Syz}_{\ker(\varphi)\cap\mathcal{N}}(F)$. Besides Lemma 3.2 yields $\mathrm{lm}_\preccurlyeq(QP) = \mathrm{lm}_{\preccurlyeq_K}(Q)K = L$. □

4 DIVIDE AND CONQUER ALGORITHM

Repeating the basic procedure described in Section 3.3 iteratively, we obtain an algorithm for syzygy basis computation when $\mathcal{N}$ is an intersection of kernels of linear functionals with a specific property (see Eq. (4)). This algorithm is similar to [25, Algo. 2] and [30, Algo. 3.2], apart from differences in the input description. Here, the input consists of linear functionals $\varphi_1,\dots,\varphi_D : \mathcal{R}^n \to \mathbb{K}$, with the assumption that

$$\mathcal{N}_i = \cap_{1\le j\le i} \ker(\varphi_j) \text{ is an } \mathcal{R}\text{-module for } 1 \le i \le D. \tag{4}$$

Then we consider the $\mathcal{R}$-module $\mathcal{N} = \mathcal{N}_D = \cap_{1\le j\le D}\ker(\varphi_j)$, which is such that $\mathcal{R}^n/\mathcal{N}$ has dimension at most D as a $\mathbb{K}$-vector

space. For F in $\mathcal{R}^{m\times n}$, the following algorithm computes a minimal $\preccurlyeq$-Gröbner basis of the syzygy module $\mathrm{Syz}_{\mathcal{N}}(F)$. Note that we do not specify the representation of F since it may depend on the specific functionals φ_i; typically, one considers F to be known modulo $\mathcal{N}$, via the images of its rows by the functionals φ_i.

Algorithm 2 SYZYGY_ITER($\varphi_1, \ldots, \varphi_D, F, \preccurlyeq$)

Input:
- linear functionals $\varphi_1, \ldots, \varphi_D : \mathcal{R}^n \to \mathbb{K}$ such that Eq. (4),
- a matrix F in $\mathcal{R}^{m\times n}$,
- a monomial order $\preccurlyeq$ on $\mathcal{R}^m$.

Output:
- a minimal $\preccurlyeq$-Gröbner basis $P \in \mathcal{R}^{k\times m}$ of $\mathrm{Syz}_{\mathcal{N}}(F)$.

1: $P \leftarrow I_m \in \mathcal{R}^{m\times m}$; $G \leftarrow F$; $L \leftarrow (e_1, \ldots, e_m) = \mathrm{lm}_{\preccurlyeq}(P)$
2: **for** $i = 1, \ldots, D$ **do**
3: $\quad (Q, L) \leftarrow$ SYZYGY_BASECASE($\varphi_i, G, \preccurlyeq, L$)
4: $\quad P \leftarrow QP$; $G \leftarrow QG$
5: **return** P

COROLLARY 4.1. *At the end of the ith iteration of Algorithm 2, P is a minimal $\preccurlyeq$-Gröbner basis of $\mathrm{Syz}_{\mathcal{N}_i}(F)$, and one has $G = PF$ as well as $L = \mathrm{lm}_{\preccurlyeq}(P)$. In particular, Algorithm 2 is correct.*

PROOF. Note that at Line 1 of Algorithm 2, $P = I_m$ is the reduced $\preccurlyeq$-Gröbner basis of $\mathcal{R}^m = \mathrm{Syz}_{\mathcal{N}_0}(F)$ with $\mathcal{N}_0 = \mathcal{R}^n$, and both $G = PF = F$ and $L = (e_1, \ldots, e_m) = \mathrm{lm}_{\preccurlyeq}(P)$ hold. We conclude that if $D = 0$, Algorithm 2 is correct.

The rest of the proof is by induction on D. We claim that the properties in the statement are preserved across the D iterations. Precisely, we assume that at the beginning of the ith iteration, P is a minimal $\preccurlyeq$-Gröbner basis of $\mathrm{Syz}_{\mathcal{N}_i}(F)$, $G = PF$, and $L = \mathrm{lm}_{\preccurlyeq}(P)$.

Since $\mathcal{N}_{i+1} = \ker(\varphi_{i+1}) \cap \mathcal{N}_i$ is an $\mathcal{R}$-module, applying Theorem 3.7 shows that (Q, L) computed during the iteration are such that $L = \mathrm{lm}_{\preccurlyeq}(QP)$ and that QP is a minimal $\preccurlyeq$-Gröbner basis of $\mathrm{Syz}_{\mathcal{N}_{i+1}}(F)$. □

This allows us to deduce bounds on the size of a minimal $\preccurlyeq$-Gröbner basis of $\mathrm{Syz}_{\mathcal{N}}(F)$.

LEMMA 4.2. *Let $P \in \mathcal{R}^{k\times m}$ be the output of Algorithm 2. Then, $m \le k \le m + (r-1)D$, and thus the same holds for any minimal $\preccurlyeq$-Gröbner basis of $\mathrm{Syz}_{\mathcal{N}}(F)$. Furthermore, at the end of the iteration i of Algorithm 2, the basis Q has at most $k + D - i$ elements.*

PROOF. Remark that all minimal $\preccurlyeq$-Gröbner bases of the same module have the same number of rows. Before the first iteration, the basis is I_m which has m rows, and each iteration of the for loop adds $\ell - 1$ rows to the basis for some ℓ in $\{0, \ldots, r\}$. Therefore $k \le m + (r-1)D$, and the last claim follows from $\ell - 1 \ge -1$. The lower bound $m \le k$ comes from the fact that $\mathcal{R}^m/\mathrm{Syz}_{\mathcal{N}}(F)$ has finite dimension as a $\mathbb{K}$-vector space. □

This iterative algorithm can be turned into a divide and conquer one (Algorithm 3), by reorganizing how the products are performed. It computes a minimal $\preccurlyeq$-Gröbner basis of $\mathrm{Syz}_{\mathcal{N}}(F)$, if one takes as input $G = F$ and $K = (e_1, \ldots, e_m)$.

Algorithm 3 SYZYGY_DAC($\varphi_1, \ldots, \varphi_D, G, \preccurlyeq, K$)

Input:
- linear functionals $\varphi_1, \ldots, \varphi_D : \mathcal{R}^n \to \mathbb{K}$,
- a matrix G in $\mathcal{R}^{k\times n}$,
- a monomial order $\preccurlyeq$ on $\mathcal{R}^m$,
- a list $K = (\mu_1, \ldots, \mu_k)$ of elements of $\mathrm{Mon}(\mathcal{R}^m)$.

Output:
- a matrix Q in $\mathcal{R}^{\ell\times m}$ for some $\ell \ge 0$,
- a list L of ℓ elements of $\mathrm{Mon}(\mathcal{R}^m)$.

1: **if** $D = 1$ **then return** SYZYGY_BASECASE($\varphi_i, G, \preccurlyeq, K$)
2: $(Q_1, L_1) \leftarrow$ SYZYGY_DAC($\varphi_1, \ldots, \varphi_{\lfloor D/2 \rfloor}, G, \preccurlyeq, K$)
3: $(Q_2, L_2) \leftarrow$ SYZYGY_DAC($\varphi_{\lfloor D/2 \rfloor + 1}, \ldots, \varphi_D, Q_1 G, \preccurlyeq, L_1$)
4: **return** $(Q_2 Q_1, L_2)$

THEOREM 4.3. *Let $\mathcal{N} \subset \mathcal{R}^n$ be an $\mathcal{R}$-submodule, let $F \in \mathcal{R}^{m\times n}$, and let $P \in \mathcal{R}^{k\times m}$ be a minimal $\preccurlyeq$-Gröbner basis of $\mathrm{Syz}_{\mathcal{N}}(F)$ for some monomial order $\preccurlyeq$ on $\mathcal{R}^m$. Assume that the input of Algorithm 3 is such that $G = PF$, and $\mathrm{lm}_{\preccurlyeq}(P) = (\mu_1, \ldots, \mu_k)$, and*

$$\mathcal{N}_i \cap \mathcal{N} \text{ is an } \mathcal{R}\text{-module for } 1 \le i \le D, \tag{5}$$

where $\mathcal{N}_i = \cap_{1\le j\le i} \ker(\varphi_j)$. Then Algorithm 3 outputs (Q, L) such that QP is a minimal $\preccurlyeq$-Gröbner basis of $\mathrm{Syz}_{\mathcal{N}_D \cap \mathcal{N}}(F)$ and $L = \mathrm{lm}_{\preccurlyeq}(QP)$.

PROOF. If $D = 1$ the output returned by Algorithm 1 is correct, since by Theorem 3.7, QP is a minimal $\preccurlyeq$-Gröbner basis of $\mathrm{Syz}_{\ker(\varphi_1)\cap\mathcal{N}}(F)$ and $L = \mathrm{lm}_{\preccurlyeq}(QP)$. We assume by induction hypothesis that Algorithm 3 returns the output foreseen by Theorem 4.3 when the number of input linear functionals is $< D$, and when the assumptions of the theorem are satisfied.

By such a hypothesis, since $G = PF$ and $K = \mathrm{lm}_{\preccurlyeq}(P)$, one deduces that (Q_1, L_1) are such that $Q_1 P$ is a $\preccurlyeq$-Gröbner basis of $\mathrm{Syz}_{\mathcal{M}}(F)$, with $\mathcal{M} = \mathcal{N}_{\lfloor D/2 \rfloor} \cap \mathcal{N}$, and $L_1 = \mathrm{lm}_{\preccurlyeq}(Q_1 P)$.

Let $\mathcal{K}_i = \cap_{\lfloor D/2 \rfloor + 1 \le j \le i} \ker(\varphi_j)$, for each $i = \lfloor D/2 \rfloor + 1, \ldots, D$. By hypothesis $\mathcal{K}_i \cap \mathcal{M} = \mathcal{N}_i \cap \mathcal{N}$ is a module, for $i = \lfloor D/2 \rfloor + 1, \ldots, i = D$. Since $Q_1 G = Q_1 PF$ and $Q_1 P$ is a $\preccurlyeq$-Gröbner basis of $\mathrm{Syz}_{\mathcal{M}}(F)$, and $L_1 = \mathrm{lm}_{\preccurlyeq}(Q_1 P)$, we can apply again the induction hypothesis, and conclude that (Q_2, L_2) is such that $Q_2 Q_1 P$ is a minimal $\preccurlyeq$-Gröbner basis of $\mathrm{Syz}_{\mathcal{K}_D \cap \mathcal{M}}(F) = \mathrm{Syz}_{\mathcal{N}_D \cap \mathcal{N}}(F)$, and $L_2 = \mathrm{lm}_{\preccurlyeq}(Q_2 Q_1 P)$. We conclude that the global output $(Q_2 Q_1, L_2)$ satisfies the claimed properties. □

5 MULTIVARIATE PADÉ APPROXIMATION

The algorithm in the previous section gives a general framework, which can be refined when applied to a particular context. Here, we consider the context of multivariate Padé approximation, where

$$\mathcal{N} = \langle X_1^{d_1}, \ldots, X_r^{d_r} \rangle \times \cdots \times \langle X_1^{d_1}, \ldots, X_r^{d_r} \rangle \subseteq \mathcal{R}^n, \tag{6}$$

for some $d_1, \ldots, d_r \in \mathbb{Z}_{>0}$. We begin with some remarks on the degrees and sizes of Gröbner bases of syzygy modules $\mathrm{Syz}_{\mathcal{N}}(F)$.

To express this context in the framework of Section 4, we take for the D linear functionals φ_i the dual basis of the canonical monomial basis of $\mathcal{R}^n/\mathcal{N}$. Precisely, the linear functionals are $\varphi_{\mu,j} : \mathcal{R}^n \to \mathbb{K}$ for $1 \le j \le n$ and all monomials $\mu \in \mathrm{Mon}(\mathcal{R})$ with $\deg_{X_i}(\mu) < d_i$ for $1 \le i \le r$, defined as follows: for $f = (f_1, \ldots, f_n) \in \mathcal{R}^n$, $\varphi_{\mu,j}(f)$ is the coefficient of the monomial μ in f_j. These linear functionals

can be ordered in several ways to ensure that Eq. (4) is satisfied. Here we design our algorithm by ordering the functionals $\varphi_{\mu,j}$ according to the term-over-position lexicographic order on the monomials $\mu \boldsymbol{e}_j \in \mathrm{Mon}(\mathcal{R}^n)$.

Example 5.1. Consider the case of $r = 2$ variables X, Y with $d_1 = 2$, $d_2 = 4$, and $n = 2$. Then the functionals are

$$\varphi_{1,1},\ \varphi_{1,2},\ \varphi_{Y,1},\ \varphi_{Y,2},\ \varphi_{Y^2,1},\ \varphi_{Y^2,2},\ \varphi_{Y^3,1},\ \varphi_{Y^3,2},$$
$$\varphi_{X,1},\ \varphi_{X,2},\ \varphi_{XY,1},\ \varphi_{XY,2},\ \varphi_{XY^2,1},\ \varphi_{XY^2,2},\ \varphi_{XY^3,1},\ \varphi_{XY^3,2},$$

in this specific order.

Lemma 5.2. *Let $\mathcal{N}$ be as in Eq. (6), let $\boldsymbol{F} \in \mathcal{R}^{m\times n}$, and let $\preccurlyeq$ be a monomial order on $\mathcal{R}^m$. Then, for $1 \le i \le r$, each polynomial in the reduced $\preccurlyeq$-Gröbner basis of $\mathrm{Syz}_{\mathcal{N}}(\boldsymbol{F})$ either has degree in X_i less than d_i or has the form $X_i^{d_i}\boldsymbol{e}_j$ for some $1 \le j \le m$.*

Proof. Let $\boldsymbol{P}$ be the reduced $\preccurlyeq$-Gröbner basis of $\mathrm{Syz}_{\mathcal{N}}(\boldsymbol{F})$ and let $i \in \{1,\dots,r\}$. Since $\mathcal{R}^m/\mathrm{Syz}_{\mathcal{N}}(\boldsymbol{F})$ has finite dimension as a $\mathbb{K}$-vector space, for each $j \in \{1,\dots,n\}$ there is a polynomial in $\boldsymbol{P}$ whose $\preccurlyeq$-leading monomial has the form $X_i^d\boldsymbol{e}_j$ for some $d \ge 0$. Since $\boldsymbol{P}$ is reduced, any other $(p_1,\dots,p_m)$ in $\boldsymbol{P}$ whose $\preccurlyeq$-leading monomial has support j is such that $\deg_{X_i}(p_j) < d \le d_i$; the last inequality follows from the fact that the monomial $X_i^{d_i}\boldsymbol{e}_j$ is in $\mathrm{Syz}_{\mathcal{N}}(\boldsymbol{F})$ and thus is a multiple of $X_i^d\boldsymbol{e}_j$. It follows that all polynomials in $\boldsymbol{P}$ whose $\preccurlyeq$-leading monomial is not among $\{X_i^{d_i}\boldsymbol{e}_j, 1 \le j \le n\}$ must have degree in X_i less than d_i. On the other hand, any polynomial in $\boldsymbol{P}$ whose $\preccurlyeq$-leading monomial is $X_i^{d_i}\boldsymbol{e}_j$ for some j must be equal to this monomial, since it belongs to $\mathrm{Syz}_{\mathcal{N}}(\boldsymbol{F})$ and $\boldsymbol{P}$ is reduced. □

In the context of Algorithm 3, Lemma 5.2 allows us to truncate the product $\boldsymbol{Q}_2\boldsymbol{Q}_1$ while preserving a $\preccurlyeq$-Gröbner basis.

Corollary 5.3. *Let $\mathcal{N}$ be as in Eq. (6), let $\boldsymbol{F} \in \mathcal{R}^{m\times n}$, let $\preccurlyeq$ be a monomial order on $\mathcal{R}^m$, and let $\boldsymbol{P} \in \mathcal{R}^{k\times m}$ be a minimal $\preccurlyeq$-Gröbner basis of $\mathrm{Syz}_{\mathcal{N}}(\boldsymbol{F})$. If $\boldsymbol{P}$ is modified by truncating each of its polynomials modulo $\langle X_1^{d_1+1},\dots,X_r^{d_r+1}\rangle$, then $\boldsymbol{P}$ is still a minimal $\preccurlyeq$-Gröbner basis of $\mathrm{Syz}_{\mathcal{N}}(\boldsymbol{F})$.*

Proof. On the first hand, this modification of $\boldsymbol{P}$ does not affect the $\preccurlyeq$-leading terms since they all have X_i-degree less than $d_i + 1$ according to Lemma 5.2, hence after modification we still have $\langle \mathrm{lm}_{\preccurlyeq}(\boldsymbol{P})\rangle = \langle \mathrm{lm}_{\preccurlyeq}(\mathrm{Syz}_{\mathcal{N}}(\boldsymbol{F}))\rangle$. On the other hand, after this modification we also have $\langle \boldsymbol{P}\rangle \subseteq \mathrm{Syz}_{\mathcal{N}}(\boldsymbol{F})$ since we started from a basis of $\mathrm{Syz}_{\mathcal{N}}(\boldsymbol{F})$ and added to each of its elements some multiples of $\langle X_1^{d_1+1},\dots,X_r^{d_r+1}\rangle$, which are contained in $\mathrm{Syz}_{\mathcal{N}}(\boldsymbol{F})$. Then [13, Lem. 15.5] yields $\langle \boldsymbol{P}\rangle = \mathrm{Syz}_{\mathcal{N}}(\boldsymbol{F})$, hence the conclusion. □

Then, the divide and conquer approach can be refined as described in Algorithm 4. The correctness of this algorithm can be shown by following the proof of Theorem 4.3 and with the following considerations. By induction hypothesis, $\boldsymbol{Q}_1$ is such that each component of the rows of $\boldsymbol{Q}_1\boldsymbol{G}$ is an element of

$$\langle X_1^{d_1},\dots,X_{j-1}^{d_{j-1}},X_j^{\lfloor d_j/2\rfloor},X_{j+1},\dots,X_r\rangle,$$

hence its truncation modulo

$$\langle X_1^{d_1},\dots,X_j^{d_j},X_{j+1},\dots,X_r\rangle$$

Algorithm 4 Padé$(d_1,\dots,d_r,\boldsymbol{G},\preccurlyeq,\boldsymbol{K})$

Input:
- integers $d_1,\dots,d_r \in \mathbb{Z}_{>0}$,
- a matrix $\boldsymbol{G}$ in $\mathcal{R}^{k\times n}$,
- a monomial order $\preccurlyeq$ on $\mathcal{R}^m$,
- a list $\boldsymbol{K} = (\boldsymbol{\mu}_1,\dots,\boldsymbol{\mu}_k)$ of elements of $\mathrm{Mon}(\mathcal{R}^m)$.

Output:
- a matrix $\boldsymbol{Q}$ in $\mathcal{R}^{\ell\times m}$ for some $\ell \ge 0$,
- a list $\boldsymbol{L}$ of ℓ elements of $\mathrm{Mon}(\mathcal{R}^m)$.

1: **if** $d_1 = \cdots = d_r = 1$ **then**
2: $\quad\boldsymbol{Q} \in \mathcal{R}^{k\times k} \leftarrow \boldsymbol{I}_k$; $\boldsymbol{H} \leftarrow \boldsymbol{G} \bmod X_1,\dots,X_r$; $\boldsymbol{L} \leftarrow \boldsymbol{K}$
3: $\quad$**for** $i = 1,\dots,n$ **do**
4: $\quad\quad\varphi \leftarrow$ linear functional $\mathcal{R}^n \to \mathbb{K}$ defined by $\varphi(\boldsymbol{f}) = f_i(\boldsymbol{0})$
5: $\quad\quad(\boldsymbol{Q}_i,\boldsymbol{L}) \leftarrow$ Syzygy_BaseCase$(\varphi,\boldsymbol{H},\preccurlyeq,\boldsymbol{L})$
6: $\quad\quad\boldsymbol{Q} \leftarrow \boldsymbol{Q}_i\boldsymbol{Q} \bmod X_1^2,\dots,X_r^2$
7: $\quad\quad\boldsymbol{H} \leftarrow \boldsymbol{Q}_i\boldsymbol{H} \bmod X_1,\dots,X_r$
8: $\quad$**return** $(\boldsymbol{Q},\boldsymbol{L})$
9: $j \leftarrow \max\{i \in \{1,\dots,r\} \mid d_i > 1\}$
10: $(\boldsymbol{Q}_1,\boldsymbol{L}_1) \leftarrow$ Padé$(d_1,\dots,d_{j-1},\lfloor d_j/2\rfloor,1,\dots,1,\boldsymbol{G},\preccurlyeq,\boldsymbol{K})$
11: $\boldsymbol{G}_2 \leftarrow X_j^{-\lfloor d_j/2\rfloor}(\boldsymbol{Q}_1\boldsymbol{G} \bmod X_1^{d_1},\dots,X_j^{d_j},X_{j+1},\dots,X_r)$
12: $(\boldsymbol{Q}_2,\boldsymbol{L}_2) \leftarrow$ Padé$(d_1,\dots,d_{j-1},\lceil d_j/2\rceil,1,\dots,1,\boldsymbol{G}_2,\preccurlyeq,\boldsymbol{L}_1)$
13: $\boldsymbol{Q} \leftarrow \boldsymbol{Q}_2\boldsymbol{Q}_1 \bmod X_1^{d_1+1},\dots,X_r^{d_r+1}$
14: **return** $(\boldsymbol{Q},\boldsymbol{L}_2)$

is an $\mathcal{R}$-multiple of $X_j^{\lfloor d_j/2\rfloor}$. It follows that on Line 11, $\boldsymbol{G}_2$ is well defined. Moreover, for $\boldsymbol{p} \in \mathcal{R}^m$ the next equations are equivalent:

$$\boldsymbol{p}\boldsymbol{Q}_1\boldsymbol{G} = 0 \mod X_1^{d_1},\dots,X_{j-1}^{d_{j-1}},X_j^{d_j}$$
$$\boldsymbol{p}\boldsymbol{G}_2 = \boldsymbol{p}X_j^{-\lfloor d_j/2\rfloor}\boldsymbol{Q}_1\boldsymbol{G} = 0 \mod X_1^{d_1},\dots,X_{j-1}^{d_{j-1}},X_j^{\lceil d_j/2\rceil}$$

This justifies the division by $X_j^{\lfloor d_j/2\rfloor}$ at Line 11 and the fact that the second call is done with $\lceil d_j/2\rceil$ instead of d_j at Line 12.

For the complexity analysis, we use Lemma 5.2 to give a bound on the size of the computed Gröbner bases, which differs from the general bound in Lemma 4.2.

Corollary 5.4 (of Lemma 5.2). *Let $\mathcal{N}$ be as in Eq. (6), let $\boldsymbol{F} \in \mathcal{R}^{m\times n}$, let $\preccurlyeq$ be a monomial order on $\mathcal{R}^m$, and let $\boldsymbol{P} \in \mathcal{R}^{k\times m}$ be a minimal $\preccurlyeq$-Gröbner basis of $\mathrm{Syz}_{\mathcal{N}}(\boldsymbol{F})$. Then,*

$$k \le md_1\cdots d_r/(\max_{1\le i\le r} d_i).$$

Proof. Let $\boldsymbol{L} = \mathrm{lm}_{\preccurlyeq}(\boldsymbol{P}) \in \mathcal{R}^{k\times m}$ and let $\bar{\imath}$ be such that $d_{\bar{\imath}} = \max_{1\le i\le r} d_i$. It is enough to prove that $\boldsymbol{L}$ has at most $d_1\cdots d_r/d_{\bar{\imath}}$ rows of the form $\mu\boldsymbol{e}_j$ for each $j \in \{1,\dots,m\}$; by Lemma 5.2, the monomial $\mu \in \mathrm{Mon}(\mathcal{R})$ has X_i-degree at most d_i for $1 \le i \le r$. Now, for each monomial $\nu = X_1^{e_1}\cdots X_{\bar{\imath}-1}^{e_{\bar{\imath}-1}}X_{\bar{\imath}+1}^{e_{\bar{\imath}+1}}\cdots X_r^{e_r}$ with $e_i \le d_i$ for all $i \ne \bar{\imath}$, there is at most one row $\mu\boldsymbol{e}_j$ in $\boldsymbol{L}$ such that $\mu = \nu X_{\bar{\imath}}^e$ for some $e \ge 0$: otherwise, one of two such rows would divide the other, which would contradict the minimality of $\boldsymbol{P}$. The number of such monomials ν is precisely $d_1\cdots d_r/d_{\bar{\imath}}$. □

Here we have $D = nd_1\cdots d_r$, hence the above bound on the cardinality of minimal $\preccurlyeq$-Gröbner bases refines the bound in Lemma 4.2 as soon as $m \le n(r-1)(\max_{1\le i\le r} d_i)$.

Proposition 5.5. *For $\mathcal{R} = \mathbb{K}[X, Y]$, let*

$$\mathcal{N} = \langle X^d, Y^e \rangle \times \cdots \times \langle X^d, Y^e \rangle \subset \mathcal{R}^n,$$

let $\boldsymbol{F} \in \mathcal{R}^{m \times n}$ with $\deg_X(\boldsymbol{F}) < d$ and $\deg_Y(\boldsymbol{F}) < e$, and let $\preccurlyeq$ be a monomial order on $\mathcal{R}^m$. Algorithm 4 computes a minimal $\preccurlyeq$-Gröbner basis of $\mathrm{Syz}_{\mathcal{N}}(\boldsymbol{F})$ using $\tilde{O}((M^{\omega-1} + Mn)(M+n)de)$ operations in $\mathbb{K}$, where $M = m \min(d, e)$.

Proof. According to Corollary 5.4, the number of rows of the matrices $\boldsymbol{Q}$ computed in Algorithm 4 is at most $M = m \min(d, e)$. It follows that all matrices $\boldsymbol{Q}_i, \boldsymbol{Q}_1, \boldsymbol{Q}_2, \boldsymbol{Q}$ in the algorithm have at most M rows and at most M columns, and that the matrices $\boldsymbol{G}, \boldsymbol{H}, \boldsymbol{G}_1, \boldsymbol{G}_2$ have at most M rows and exactly n columns. Besides, by Kronecker substitution [7, Chap. 1 Sec. 8], multiplying two bivariate matrices of dimensions $M \times M$ (resp. $M \times n$) and bidegree at most (d, e) costs $\tilde{O}(M^\omega de)$ (resp. $\tilde{O}(M^\omega(1 + n/M)de)$) operations in $\mathbb{K}$.

Let $C(m, n, d, e)$ denote the number of field operations used by Algorithm 4; we have $C(m, n, d, e) \leq C(M, n, d, e)$. First, for $e > 1$, $C(M, n, d, e)$ is bounded by $C(M, n, d, \lfloor e/2 \rfloor) + C(M, n, d, \lceil e/2 \rceil) + \tilde{O}(M^\omega(1 + n/M)de)$. Indeed, there are two recursive calls with parameters $(d, \lfloor e/2 \rfloor)$ and $(d, \lceil e/2 \rceil)$, and two matrix products $\boldsymbol{Q}_1\boldsymbol{G}$ and $\boldsymbol{Q}_2\boldsymbol{Q}_1$ to perform; as noted above, the latter products cost $\tilde{O}(M^\omega(1 + n/M)de)$ operations in $\mathbb{K}$. The same analysis for $d > 1$ and $e = 1$ shows that $C(M, n, d, 1)$ is bounded by $C(M, n, \lfloor d/2 \rfloor, 1) + C(M, n, \lceil d/2 \rceil, 1) + \tilde{O}(M^\omega(1 + n/M)d)$.

Finally, for $d = e = 1$, we show that $C(M, n, 1, 1) \in O(M(M+n)n)$. In this case, there are n iterations of the loop. Each of them makes one call to Syzygy_BaseCase, which uses $O(M)$ field operations for computing the λ_i's at Line 7; note that the α_j's are zero in the present context where the linear functional φ corresponds to the constant coefficient. The computed basis $\boldsymbol{Q}_i$ has a single nontrivial column (it has the form in Eq. (3)), so that computing $\boldsymbol{Q}_i\boldsymbol{Q}$ mod $\langle X_1^2, \ldots, X_r^2 \rangle$ (resp. $\boldsymbol{Q}_i\boldsymbol{H}$ mod $\langle X_1, \ldots, X_r \rangle$) can be done naively at a cost of $O(M^2)$ (resp. $O(M(M+n))$) operations in $\mathbb{K}$.

Based on the previous inequalities, unrolling the recursion by following the divide-and-conquer scheme leads to the announced complexity bound. □

ACKNOWLEDGMENTS

Acknowledgements. The first author acknowledges support from the Fondation Mathématique Jacques Hadamard through the Programme PGMO, project number 2018-0061H.

REFERENCES

[1] M.E. Alonso, M.G. Marinari, and T. Mora. 2003. The Big Mother of all Dualities: Möller Algorithm. *Communications in Algebra* 31, 2 (2003), 783–818. https://doi.org/10.1081/AGB-120017343

[2] B. Beckermann. 1992. A reliable method for computing M-Padé approximants on arbitrary staircases. *J. Comput. Appl. Math.* 40, 1 (1992), 19–42. https://doi.org/10.1016/0377-0427(92)90039-Z

[3] B. Beckermann and G. Labahn. 1994. A Uniform Approach for the Fast Computation of Matrix-Type Padé Approximants. *SIAM J. Matrix Anal. Appl.* 15, 3 (1994), 804–823. https://doi.org/10.1137/S0895479892230031

[4] B. Beckermann and G. Labahn. 1997. Recursiveness in matrix rational interpolation problems. *J. Comput. Appl. Math.* 77, 1 (1997), 5–34. https://doi.org/10.1016/S0377-0427(96)00120-3

[5] C. Berkesch and F.-O. Schreyer. 2015. Syzygies, finite length modules, and random curves. In *Commutative Algebra and Noncommutative Algebraic Geometry*. Mathematical Sciences Research Institute Publications (Vol. 67), pp. 25–52.

[6] J. Berthomieu and J.-C. Faugère. 2018. A Polynomial-Division-Based Algorithm for Computing Linear Recurrence Relations. In *Proceedings ISSAC 2018*. 79–86. https://doi.org/10.1145/3208976.3209017

[7] D. Bini and V. Y. Pan. 1994. *Polynomial and Matrix Computations (Vol. 1): Fundamental Algorithms*. Birkhauser Verlag.

[8] M. Ceria and T. Mora. 2018. Combinatorics of ideals of points: a Cerlienco-Mureddu-like approach for an iterative lex game. *Preprint arXiv:1805.09165*.

[9] L. Cerlienco and M. Mureddu. 1995. From algebraic sets to monomial linear bases by means of combinatorial algorithms. *Discrete Mathematics* 139, 1-3 (1995), 73–87. https://doi.org/10.1016/0012-365X(94)00126-4

[10] D. Coppersmith and S. Winograd. 1990. Matrix multiplication via arithmetic progressions. *J. Symb. Comput.* 9, 3 (1990), 251–280. https://doi.org/10.1016/S0747-7171(08)80013-2

[11] D. A. Cox, J. Little, and D. O'Shea. 2005. *Using Algebraic Geometry (second edition)*. Springer-Verlag New-York, New York, NY. https://doi.org/10.1007/b138611

[12] D. A. Cox, J. Little, and D. O'Shea. 2007. *Ideals, Varieties, and Algorithms (third edition)*. Springer-Verlag New-York, New York, NY. https://doi.org/10.1007/978-0-387-35651-8

[13] D. Eisenbud. 1995. *Commutative Algebra: with a View Toward Algebraic Geometry*. Springer, New York, Berlin, Heildelberg. https://doi.org/10.1007/978-1-4612-5350-1

[14] J.B. Farr and S. Gao. 2006. Computing Gröbner bases for vanishing ideals of finite sets of points. In *International Symposium on Applied Algebra, Algebraic Algorithms, and Error-Correcting Codes*. Springer, 118–127.

[15] B. Felszeghy, B. Ráth, and L. Rónyai. 2006. The lex game and some applications. *J. Symb. Comput.* 41, 6 (2006), 663–681. https://doi.org/10.1016/j.jsc.2005.11.003

[16] P. Fitzpatrick. 1997. Solving a Multivariable Congruence by Change of Term Order. *J. Symb. Comput.* 24, 5 (1997), 575–589. https://doi.org/10.1006/jsco.1997.0153

[17] P. Fitzpatrick and J. Flynn. 1992. A Gröbner basis technique for Padé approximation. *J. Symb. Comput.* 13, 2 (1992), 133–138. https://doi.org/10.1016/S0747-7171(08)80087-9

[18] K. O. Geddes. 1973. *Algorithms for Analytic Approximation (to a Formal Power-series)*. Ph.D. Dissertation. University of Toronto, Canada.

[19] P. Giorgi, C.-P. Jeannerod, and G. Villard. 2003. On the complexity of polynomial matrix computations. In *ISSAC'03* (Philadelphia, PA, USA). ACM, 135–142. https://doi.org/10.1145/860854.860889

[20] M. Janet. 1920. Sur les systèmes d'équations aux dérivées partielles. *J. Math. Pures Appl.* 170 (1920), 65–152.

[21] C.-P. Jeannerod, V. Neiger, É. Schost, and G. Villard. 2016. Fast computation of minimal interpolation bases in Popov form for arbitrary shifts. In *ISSAC'16* (Waterloo, ON, Canada). ACM, 295–302. https://doi.org/10.1145/2930889.2930928

[22] C.-P. Jeannerod, V. Neiger, É. Schost, and G. Villard. 2017. Computing minimal interpolation bases. *J. Symb. Comput.* 83 (2017), 272–314. https://doi.org/10.1016/j.jsc.2016.11.015

[23] C.-P. Jeannerod, V. Neiger, and G. Villard. 2020. Fast computation of approximant bases in canonical form. *J. Symb. Comput.* 98 (2020), 192–224. https://doi.org/10.1016/j.jsc.2019.07.011

[24] F. Le Gall. 2014. Powers of Tensors and Fast Matrix Multiplication. In *ISSAC'14* (Kobe, Japan). ACM, 296–303. https://doi.org/10.1145/2608628.2608664

[25] M. G. Marinari, H. M. Möller, and T. Mora. 1993. Gröbner bases of ideals defined by functionals with an application to ideals of projective points. *Appl. Algebra Engrg. Comm. Comput.* 4, 2 (1993), 103–145. https://doi.org/10.1007/BF01386834

[26] H. M. Möller and B. Buchberger. 1982. The Construction of Multivariate Polynomials with Preassigned Zeros. In *EUROCAM'82 (LNCS)*, Vol. 144. Springer, 24–31. https://doi.org/10.1007/3-540-11607-9_3

[27] T. Mora. 2009. The FGLM Problem and Möller's Algorithm on Zero-dimensional Ideals. In *Gröbner Bases, Coding, and Cryptography*, M. Sala, S. Sakata, T. Mora, C. Traverso, and L. Perret (Eds.). Springer Berlin Heidelberg, Berlin, Heidelberg, 27–45. https://doi.org/10.1007/978-3-540-93806-4_3

[28] V. Neiger. 2016. *Bases of relations in one or several variables: fast algorithms and applications*. Ph.D. Dissertation. École Normale Supérieure de Lyon. https://tel.archives-ouvertes.fr/tel-01431413/

[29] V. Neiger and É. Schost. 2019. Computing syzygies in finite dimension using fast linear algebra. *Preprint arXiv:1912.01848*.

[30] H. O'Keeffe and P. Fitzpatrick. 2002. Gröbner basis solutions of constrained interpolation problems. *Linear Algebra Appl.* 351 (2002), 533–551. https://doi.org/10.1016/S0024-3795(01)00509-2

[31] F-O. Schreyer. 1980. *Die Berechnung von Syzygien mit dem verallgemeinerten Weierstraßschen Divisionssatz*. Ph.D. Dissertation. Master's thesis, Fakultät für Mathematik, Universität Hamburg.

[32] A. Storjohann. 2006. Notes on computing minimal approximant bases. In *Challenges in Symbolic Computation Software (Dagstuhl Seminar Proceedings)*. http://drops.dagstuhl.de/opus/volltexte/2006/776

[33] M. Van Barel and A. Bultheel. 1992. A general module theoretic framework for vector M-Padé and matrix rational interpolation. *Numer. Algorithms* 3 (1992), 451–462. https://doi.org/10.1007/BF02141952

[34] P. Wynn. 1960. The Rational Approximation of Functions which are Formally Defined by a Power Series Expansion. *Math. Comp.* 14, 70 (1960), 147–186.

[35] W. Zhou and G. Labahn. 2012. Efficient Algorithms for Order Basis Computation. *J. Symb. Comput.* 47, 7 (2012), 793–819. https://doi.org/10.1016/j.jsc.2011.12.009

Generic Bivariate Multi-point Evaluation, Interpolation and Modular Composition with Precomputation

Vincent Neiger
Univ. Limoges, CNRS, XLIM, UMR 7252
F-87000 Limoges, France

Johan Rosenkilde
Technical University of Denmark
Kgs. Lyngby, Denmark

Grigory Solomatov
Technical University of Denmark
Kgs. Lyngby, Denmark

ABSTRACT

Suppose $\mathbb{K}$ is a large enough field and $\mathcal{P} \subset \mathbb{K}^2$ is a fixed, generic set of points which is available for precomputation. We introduce a technique called *reshaping* which allows us to design quasi-linear algorithms for both: computing the evaluations of an input polynomial $f \in \mathbb{K}[x, y]$ at all points of $\mathcal{P}$; and computing an interpolant $f \in \mathbb{K}[x, y]$ which takes prescribed values on $\mathcal{P}$ and satisfies an input y-degree bound. Our genericity assumption is explicit and we prove that it holds for most point sets over a large enough field. If $\mathcal{P}$ violates the assumption, our algorithms still work and the performance degrades smoothly according to a distance from being generic. To show that the reshaping technique may have an impact on other related problems, we apply it to modular composition: suppose generic polynomials $M \in \mathbb{K}[x]$ and $A \in \mathbb{K}[x]$ are available for precomputation, then given an input $f \in \mathbb{K}[x, y]$ we show how to compute $f(x, A(x)) \operatorname{rem} M(x)$ in quasi-linear time.

KEYWORDS

Multi-point evaluation, interpolation, modular composition, bivariate polynomials, precomputation.

ACM Reference Format:
Vincent Neiger, Johan Rosenkilde, and Grigory Solomatov. 2020. Generic Bivariate Multi-point Evaluation, Interpolation and Modular Composition with Precomputation. In *International Symposium on Symbolic and Algebraic Computation (ISSAC '20), July 20–23, 2020, Kalamata, Greece.* ACM, New York, NY, USA, 8 pages. https://doi.org/10.1145/3373207.3404032

1 INTRODUCTION

Outline. Let $\mathbb{K}$ be an effective field. We consider the three classical problems for bivariate polynomials $\mathbb{K}[x, y]$ mentioned in the title. We assume a model where part of the input is given early as *preinput* which is available for heavier computation, and the primary goal is to keep the complexity of the *online phase*, once the remaining part of the input is given, to a minimum.

Multi-point evaluation (MPE): with preinput a point set $\mathcal{P} = \{(\alpha_i, \beta_i)\}_{i=1}^n \subseteq \mathbb{K}^2$ and input $f \in \mathbb{K}[x, y]$, compute $\big(f(\alpha_i, \beta_i)\big)_{i=1}^n$. We give two algorithms: the first requires pairwise distinct α_i's and has online complexity $\tilde{O}(\deg_x f \deg_y f + n)$ as long as $\mathcal{P}$ is *balanced*, a notion described below; the second accepts repeated x-coordinates with online complexity $\tilde{O}(\deg_x f(\deg_x f + \deg_y f) + n)$ as long as a certain "shearing" of $\mathcal{P}$ is balanced. "*soft-O*" ignores logarithmic terms: $O(f(n)(\log f(n))^c) \subset \tilde{O}(f(n))$ for any $c \in \mathbb{Z}_{\geq 0}$.

Interpolation: with preinput a point set $\mathcal{P}$ as before, and input values $\boldsymbol{\gamma} \in \mathbb{K}^n$, compute $f \in \mathbb{K}[x, y]$ such that $\big(f(\alpha_i, \beta_i)\big)_{i=1}^n = \boldsymbol{\gamma}$, satisfying some constraints on the monomial support. We give an algorithm which preinputs a degree bound d and outputs f such that $\deg_y f < d$ and $\deg_x f \in O(n/d)$. The online complexity is $\tilde{O}(n)$ if $\mathcal{P}$ and a shearing of $\mathcal{P}$ are both balanced; d should exceed the *x-valency of $\mathcal{P}$*, i.e. the maximal number of y-coordinates for any given x-coordinate.

Modular composition: with preinput $M, A \in \mathbb{K}[x]$, we input $f \in \mathbb{K}[x, y]$ and compute $f(x, A) \operatorname{rem} M$. Our algorithm has online complexity $\tilde{O}(\deg_x f \deg_y f + \deg A + \deg M)$, as long as the bivariate ideal $\langle M, y - A \rangle$ is balanced.

We prove that if $\mathcal{P} \subseteq \mathbb{K}^2$ is random of fixed cardinality n, and if $|\mathbb{K}| \gg n^2 \log(n)$ then $\mathcal{P}$ is balanced with high probability. Similarly, if M is square-free and A is uniformly random of degree less than $\deg M$, then $\langle M, y - A \rangle$ is balanced with high probability. Our proof techniques currently do not extend to proving that sheared point sets are balanced. A few trials we conducted suggest that this may often be the case if the x-valency of $\mathcal{P}$ is not too high. The cost of the second MPE algorithm is not symmetric in the x- and y-degree, so whenever $\deg_x f < \deg_y f$ one should consider transposing the input, i.e. evaluating $f(y, x)$ on $\{(\beta_i, \alpha_i)\}_{i=1}^n$. In this case, the balancedness assumption is on the transposed point set.

Our algorithms are deterministic, and once the preinput has been processed, the user knows whether it is balanced and hence whether the algorithms will perform well. Further, the performance of our algorithms deteriorates smoothly with how "unbalanced" the preinput is, in the sense of certain polynomials, which depend only on preinput, having sufficiently well behaved degrees. In a toolbox one might therefore apply our algorithms whenever the preinput turns out to be sufficiently balanced and reverting to other algorithms on very unbalanced preinput.

A typical use of precomputation is if we compute e.g. MPEs on the same point set for many different polynomials. This occurs in coding theory, where bivariate MPE corresponds to the encoding stage of certain families of codes such as some Reed-Muller codes [1, Chap. 5] and some algebraic-geometric codes [14]: here $\mathcal{P}$ is fixed and communication consists of a long series of bivariate MPEs on $\mathcal{P}$. In these applications, $\mathcal{P}$ is often not random but chosen carefully, and so our genericity assumptions might not apply.

Techniques. We introduce a tool we call *reshaping* for achieving the following: given an ideal $I \subseteq \mathbb{K}[x, y]$ and $f \in \mathbb{K}[x, y]$, compute $\hat{f} \in f + I$ with smaller y-degree. For instance in MPE, we let

ISSAC '20, July 20–23, 2020, Kalamata, Greece

ACM ISBN 978-1-4503-7100-1/20/07…$15.00
https://doi.org/10.1145/3373207.3404032

$\Gamma \subset \mathbb{K}[x,y]$ be the ideal of polynomials which vanish on all the points $\mathcal{P}$. Then all elements of $f+\Gamma$ have the same evaluations on $\mathcal{P}$, so we compute a $\hat{f} \in f+\Gamma$ of y-degree 0 (it exists if $\mathcal{P}$ has distinct x-coordinates), and then apply fast univariate MPE.

An obvious idea to accomplish this iteratively is to find some $g \in \Gamma$ of lower y-degree than f and whose leading y-term is 1, and then compute $\tilde{f} = f \operatorname{rem} g$. The problem is that the x-degree of $\tilde{f}$ may now be as large as $\deg_x f + (\deg_y f - \deg_y g)\deg_x g$. Our idea is to seek polynomials g that we call *reshapers*, which have the form

$$g = y^{2d/3} - \hat{g}\,,$$

where $\deg_y \hat{g} < d/3$ and $d = \deg_y f + 1$ (for simplicity, here 3 divides d). Writing $f = f_1 y^{2d/3} + f_0$ with $\deg_y f_0 < 2d/3$, then $\tilde{f} = f_1\hat{g} + f_0$ is easy to compute, has y-degree less than $2d/3$, and x-degree only $\deg_x f + \deg_x g$. Repeating such a reduction $O(\log(d))$ times with reshapers of progressively smaller y-degree, we eventually reach y-degree 0.

For efficiency, we therefore need the x-degrees of all these reshapers g to be small. For MPE, stating that $g \in \Gamma$ specifies n linear contraints on the coefficients of $\hat{g}$, so we look for g with about n monomials. Generically, since $\deg_y \hat{g} \approx d/3$, one may expect to find g with $\deg_x g \approx 3n/d$. Informally, $\mathcal{P}$ is *balanced* if all the reshapers needed in the above process satisfy this degree constraint.

Above, we assumed the point set has distinct x-coordinates. To handle repetitions, we shear the points by $(\alpha,\beta) \mapsto (\alpha+\theta\beta,\beta)$, where θ generates an extension field of $\mathbb{K}$ of degree 2. The resulting point set has distinct x-coordinates. This replaces $f(x,y)$ with $f(x-\theta y, y)$, and whenever $\deg_x f < \deg_y f$ we stay within quasi-linear complexity if the sheared point set is balanced.

Previous work. Quasi-linear complexity has been achieved for multivariate MPE and interpolation on special point sets and monomial support: Pan [18] gave an algorithm on grids, and van der Hoeven and Schost [26] (see also [5, Sec. 2]) generalised this to certain types of subsets of grids, constraining both the points and the monomial support. See [26] for references to earlier work on interpolation, not achieving quasi-linear complexity.

In classical univariate modular composition, we are given f, M, A in $\mathbb{K}[x]$ and seek $f(A) \operatorname{rem} M$. Brent and Kung's baby-step giant-step algorithm [2, 19] performs this operation in $\tilde{O}(n^{(\omega+1)/2})$, where ω is the matrix multiplication exponent with best known bound $\omega < 2.373$ [13]. Nüsken and Ziegler [17] extended this to a bivariate f, computing $f(x,A) \operatorname{rem} M$ in complexity $O(\deg_x f(\deg_y f)^{(\omega+1)/2})$, assuming that A and M have degree at most $\deg_x f \deg_y f$. They applied this to solve MPE in the same cost; in this paper, we use essentially the same link between these problems. To the best of our knowledge, this is currently the best known cost bound for these problems, in the algebraic complexity model.

In a breakthough, Kedlaya and Umans [11] achieved "almost linear" time for modular composition and MPE, for specific types of fields $\mathbb{K}$ and in the bit complexity model. For modular composition, the cost is $O(n^{1+\epsilon})$ bit operations for any $\epsilon > 0$, while for MPE it is $O((n + (\deg_x f)^2)^{1+\epsilon})$, assuming $\deg_y f < \deg_x f$ (the algorithm also supports multivariate MPE). Unfortunately, these algorithms have so far resisted attempts at a practical implementation [25].

Our quasi-linear complexities improve upon the above results (including Kedlaya and Umans' ones since quasi-linear compares favorably to almost linear); however we stress that none of the latter have the two constraints of our work: allowing precomputation, and genericity assumption. For modular composition, precomputation on M was suggested in [24] to leverage its factorisation structure. Except for slight benefits of precomputation in Brent and Kung's modular composition (used in the Flint and NTL libraries [8, 22]), we are unaware of previous work focusing on the use of precomputation for MPE, Interpolation, and Modular Composition.

Genericity has recently been used by Villard [27], who showed how to efficiently compute the resultant of two generic bivariate polynomials; a specific case computes, for given univariate M and A, the characteristic polynomial of A in $\mathbb{K}[x]/\langle M\rangle$, with direct links to the modular composition $f(A) \operatorname{rem} M$ [27, 28]. This led to an ongoing work on achieving exponent $(\omega+2)/3$ for modular composition [15]. In that line, the main benefit from genericity is that $\langle M, y-A\rangle$ admits bases formed by m polynomials of y-degree $< m$ and x-degree at most $\deg(M)/m$, for a given parameter $2 \le m \le \deg(M)$. Such a basis is represented as an $m \times m$ matrix over $\mathbb{K}[x]$ with all entries of degree at most $\deg(M)/m$, and one can then rely on fast univariate polynomial matrix algorithms. In this paper, genericity serves a purpose similar to that in [15, 27]: it ensures the existence of such bases for several parameters m, and also of the reshapers g mentioned above; besides we make use of these bases to precompute these reshapers. Whereas an important contribution of [27] is the efficient computation of such bases, here they are only used to find reshapers in the precomputation stage and the speed of computing them is not a main concern. Once the reshapers are known, our algorithms work without requiring any other genericity property.

Organisation. After some preliminaries in Section 2, we describe the reshaping strategy for an arbitrary ideal in Section 3. Then Sections 4 to 6 give algorithms for each of the three problems. We discuss precomputation in Section 7 and genericity in Section 8.

2 PRELIMINARIES

For complexity estimates, we use the algebraic RAM model and count arithmetic operations in $\mathbb{K}$. By $\mathsf{M}(n)$ we denote the cost of multiplying two univariate polynomials over $\mathbb{K}$ of degree at most n; one may take $\mathsf{M}(n) \in O(n\log n\log\log n) \subset \tilde{O}(n)$ [3]. Division with remainder in $\mathbb{K}[x]$ also costs $O(\mathsf{M}(n))$ [30, Thm. 9.6]. When degrees of a polynomial, say $f \in \mathbb{K}[x,y]$, appear in complexity estimates, we abuse notation and let $\deg_x f$ denote $\max(\deg_x f, 1)$.

It is well-known that univariate interpolation and multi-point evaluation can be done in quasi-linear time [30, Cor. 10.8 and 10.12]: given $f \in \mathbb{K}[x]$ and $\alpha_1,\dots,\alpha_n \in \mathbb{K}$, we may compute $\big(f(\alpha_i)\big)_{i=1}^n$ in time $O(\mathsf{M}(\deg_x f + n)\log n) \subseteq \tilde{O}(\deg_x f + n)$; given $\alpha_1,\dots,\alpha_n$ and $\beta_1,\dots,\beta_n$ in $\mathbb{K}$ with the α_i's pairwise distinct, we may compute the unique corresponding interpolant in time $O(\mathsf{M}(n)\log n) \subseteq \tilde{O}(n)$. We will also use the fact that two bivariate $f, g \in \mathbb{K}[x,y]$ can be multiplied in time $O(\mathsf{M}(d_x d_y)) \subset \tilde{O}(d_x d_y)$, where $d_x = \max(\deg_x f, \deg_x g)$ and $d_y = \max(\deg_y f, \deg_y g)$ [30, Cor. 8.28].

For a bivariate polynomial $f = \sum_{i=0}^k f_i(x) y^i \in \mathbb{K}[x,y]$ such that $f_k \neq 0$, we define its y-leading coefficient as $\mathsf{LC}_y(f) = f_k \in \mathbb{K}[x]$.

For our genericity results, we will invoke the following staple:

Lemma 2.1 (DeMillo-Lipton-Schwartz-Zippel [7, 21, 31]). *Let $f \in \mathbb{K}[x_1,\dots,x_n]$ be non-zero of total degree d, and $\mathcal{T} \subseteq \mathbb{K}$ be finite.*

For $\alpha_1, \ldots, \alpha_k \in \mathcal{T}$ chosen independently and uniformly at random, the probability that $f(\alpha_1, \ldots, \alpha_k) = 0$ is at most $d/|\mathcal{T}|$.

For a point set $\mathcal{P} \subseteq \mathbb{K}^2$, the x-valency of $\mathcal{P}$, denoted by $v_x(\mathcal{P})$, is the largest number of y-coordinates for any given x-coordinate, i.e.

$$v_x(\mathcal{P}) = \max_{\alpha \in \mathbb{K}} |\{\beta \in \mathbb{K} \mid (\alpha, \beta) \in \mathcal{P}\}| .$$

When $v_x(\mathcal{P}) = 1$, the x-coordinates of $\mathcal{P}$ are pairwise distinct.

The vanishing ideal of $\mathcal{P}$ is the bivariate ideal

$$\Gamma(\mathcal{P}) = \{f \in \mathbb{K}[x, y] \mid f(\alpha, \beta) = 0 \text{ for all } (\alpha, \beta) \in \mathcal{P}\} ,$$

Hereafter, $\prec_{\text{lex}}$ stands for the lexicographic order on $\mathbb{K}[x, y]$ with $x \prec_{\text{lex}} y$, and $\mathsf{LT}_{\text{lex}}(f)$ is the $\prec_{\text{lex}}$-leading term of $f \in \mathbb{K}[x, y]$. The following is folklore and follows e.g. from [12] and [6, Thm. 3].

Lemma 2.2. *Let $\mathcal{P} \subset \mathbb{K}^2$ be a point set of cardinality n and let $G = \{g_1, \ldots, g_s\}$ be the reduced $\prec_{\text{lex}}$-Gröbner basis of $\Gamma(\mathcal{P})$, ordered by $\prec_{\text{lex}}$. Then $g_1 \in \mathbb{K}[x]$, and g_s is y-monic with $\deg_y g_s = v_x(\mathcal{P})$.*

3 RESHAPE

We first describe our algorithm Reshape which takes $f \in \mathbb{K}[x, y]$ and an ideal I and finds $\hat{f} \in f + I$ whose y-degree is below some target. This will pass through several intermediate elements of $f + I$ of progressively smaller y-degree. This sequence of y-degrees has the following form:

Definition 3.1. We say $\boldsymbol{\eta} = (\eta_i)_{i=0}^k \in \mathbb{Z}_{>0}^{k+1}$ is a (η_0, η_k)-*reshaping sequence* if $\eta_{i-1} > \eta_i \geq \lfloor \frac{2}{3}\eta_{i-1} \rfloor$ for $i = 1, \ldots, k$. For $I \subseteq \mathbb{K}[x, y]$ an ideal and $\boldsymbol{\eta} = (\eta_i)_{i=0}^k$ a reshaping sequence, we say $\boldsymbol{g} = (g_i)_{i=1}^k \in I^k$ is an $\boldsymbol{\eta}$-*reshaper for* I if $g_i = y^{\eta_i} + \hat{g}_i$ where $\deg_y \hat{g}_i \leq 2\eta_i - \eta_{i-1}$, for each $i = 1, \ldots, k$.

Our algorithms are faster with short reshaping sequences, so we should choose $\eta_i \approx \frac{2}{3}\eta_{i-1}$, and hence $2\eta_i - \eta_{i-1} \approx \frac{1}{3}\eta_i$. It is easy to see that for any $a, b \in \mathbb{Z}_{>0}$, there is an (a, b)-reshaping sequence of length less than $\log_{3/2}(a) + 2$. Observe that for any (a, b)-reshaping sequence we have $\eta_i \geq \frac{2}{3}(\eta_{i-1} - 1)$ for $i = 1, \ldots, k$ and therefore

$$2\eta_i - \eta_{i-1} \geq \tfrac{\eta_{i-1}-4}{3} \geq \tfrac{\eta_i}{3} - 1 . \tag{1}$$

By considering the cases $\eta_i \geq 3$ and $\eta_i = 1, 2$, we get $2\eta_i - \eta_{i-1} \geq 0$.

Theorem 3.2. *Algorithm 1 is correct and has complexity*

$$\tilde{O}(\textstyle\sum_{i=i_0}^k \eta_i(\deg_x f + \sum_{j=i_0}^i \deg_x g_j))$$
$$\subseteq \tilde{O}(k \deg_y f \deg_x f + k \textstyle\sum_{i=i_0}^k \eta_i \deg_x g_i) ,$$

for the smallest i_0 such that $\eta_{i_0} \leq \deg_y f$.

Proof. Let $\hat{f}_i, \hat{f}_{i,0}, \hat{f}_{i,1}$ be the values of $\hat{f}, \hat{f}_0, \hat{f}_1$ at the end of iteration i. First, the iterations for $i < i_0$ perform no operation and keep $\hat{f}_i = f$, since $\eta_i > \deg_y \hat{f}_{i-1}$ implies $\hat{f}_{i,1} = 0$ and $\hat{f}_i = \hat{f}_{i-1}$. In particular, if $\eta_i > \deg_y f$ for all i then the algorithm is correct and returns f without using any arithmetic operation. Now for $i \geq i_0$, observe that $\hat{f}_i = \hat{f}_{i,1}\hat{g}_i + \hat{f}_{i,0} = \hat{f}_{i-1} - \hat{f}_{i,1}g_i$; thus in the end $\hat{f} \in f + I$ since each g_i belongs to I. We show the following loop invariants, which imply the degree bounds on the output:

$$\deg_x \hat{f}_i \leq \deg_x f + \textstyle\sum_{j=i_0}^i \deg_x g_j, \text{ and } \deg_y \hat{f}_i < \eta_i .$$

Both are true for $i = i_0 - 1$ (just before the loop, if $i_0 = 1$). For the x-degree, $\hat{f}_i = \hat{f}_{i-1} - \hat{f}_{i,1}g_i$ yields $\deg_x \hat{f}_i \leq \deg_x \hat{f}_{i-1} + \deg_x g_i$,

Algorithm 1 Reshape$(f, \boldsymbol{\eta}, \boldsymbol{g})$

Input: A bivariate polynomial $f \in \mathbb{K}[x, y]$; a reshaping sequence $\boldsymbol{\eta} = (\eta_i)_{i=0}^k \in \mathbb{Z}_{>0}^{k+1}$ with $\deg_y f < \eta_0$; an $\boldsymbol{\eta}$-reshaper $\boldsymbol{g} = (g_i)_{i=1}^k \in I^k$ for some ideal $I \subseteq \mathbb{K}[x, y]$.

Output: a polynomial $\hat{f} \in f + I$ such that $\deg_y \hat{f} < \eta_k$ and $\deg_x \hat{f} \leq \deg_x f + \sum_{i=1}^k \deg_x g_i$.

1: $\hat{f} \leftarrow f$
2: **for** $i = 1, \ldots, k$ **do**
3: Write $g_i = y^{\eta_i} + \hat{g}_i$ where $\deg_y \hat{g}_i \leq 2\eta_i - \eta_{i-1}$
4: Write $\hat{f} = \hat{f}_1 y^{\eta_i} + \hat{f}_0$ where $\deg_y \hat{f}_0 < \eta_i$
5: $\hat{f} \leftarrow \hat{f}_1\hat{g}_i + \hat{f}_0$ ▷ *equivalent to* $\hat{f} \leftarrow \hat{f} - \hat{f}_1 g_i$
6: **return** $\hat{f}$

and the loop invariant follows. For the y-degree, by construction $\deg_y \hat{f}_{i,0} < \eta_i$ and $\deg_y \hat{f}_{i,1} \leq \deg_y \hat{f}_{i-1} - \eta_i$ hold; the assumption $\deg_y \hat{f}_{i-1} < \eta_{i-1}$ then gives $\deg_y \hat{f}_{i,1}\hat{g}_i < \eta_i$, hence $\deg_y \hat{f}_i < \eta_i$.

For complexity, the only costly step is at Line 5 and for iterations $i \geq i_0$. From the above bound $\deg_y \hat{f}_{i,1}\hat{g}_i < \eta_i$, multiplying $\hat{f}_{i,1}$ and $\hat{g}_i$ costs $O(\mathsf{M}((\deg_x \hat{f}_{i,1} + \deg_x \hat{g}_i)\eta_i))$. Since $\deg_x \hat{g}_i = \deg_x g_i$, since both $\hat{f}_{i,0}$ and $\hat{f}_{i,1}$ have x-degree at most $\deg_x \hat{f}_{i-1}$, and since $\deg_y \hat{f}_{i,0} < \eta_i$, the total cost of the ith iteration is in

$$\tilde{O}((\deg_x \hat{f}_{i-1} + \deg_x \hat{g}_i)\eta_i) \subseteq \tilde{O}((\deg_x f + \textstyle\sum_{j=i_0}^i \deg_x g_j)\eta_i).$$

Summing over all iterations, we get the first complexity bound in the theorem; the second one follows from it, using the fact that $\deg_y f \geq \eta_{i_0} > \eta_{i_0+1} > \ldots > \eta_k$ and $i_0 \geq 1$. □

We now define the balancedness of a point set. In Section 8 we prove that this notion captures the *expected* x-degree of reshapers.

Definition 3.3. Let $\mathcal{P} \subseteq \mathbb{K}^2$ be a point set of cardinality n, and let $\boldsymbol{\eta} = (\eta_i)_{i=0}^k$ be a reshaping sequence. Then $\mathcal{P}$ is $\boldsymbol{\eta}$-*balanced* if there exists an $\boldsymbol{\eta}$-reshaper $\boldsymbol{g} = (g_i)_{i=1}^k \in \mathbb{K}[x, y]^k$ for $\Gamma(\mathcal{P})$ such that $\deg_x g_i \leq \lfloor \frac{n}{2\eta_i - \eta_{i-1}+1} \rfloor + 1$ for $i = 1, \ldots, k$.

The next bound is often used below for deriving complexity estimates; it follows directly from Eq. (1).

Lemma 3.4. *Let $\boldsymbol{\eta} = (\eta_i)_{i=0}^k$ be a reshaping sequence, $\mathcal{P} \subseteq \mathbb{K}^2$ be an $\boldsymbol{\eta}$-balanced point set of cardinality n, and $\boldsymbol{g} = (g_i)_{i=1}^k$ be an $\boldsymbol{\eta}$-reshaper for $\Gamma(\mathcal{P})$. Then $\sum_{i=i_0}^k \eta_i \deg_x g_i \leq (3n + \eta_{i_0})k$ for $1 \leq i_0 \leq k$.*

We conclude this section with two results about the existence of $\boldsymbol{\eta}$-reshapers for vanishing ideals of point sets.

Lemma 3.5. *Let $\mathcal{P} \subseteq \mathbb{K}^2$ be a point set and $\boldsymbol{\eta} = (\eta_i)_{i=0}^k$ a reshaping sequence. If $v_x(\mathcal{P}) \leq \min_{1 \leq i \leq k}(2\eta_i - \eta_{i-1} + 1)$, then there exists an $\boldsymbol{\eta}$-reshaper $\boldsymbol{g} \in \mathbb{K}[x, y]^k$ for $\Gamma(\mathcal{P})$.*

Proof. By Lemma 2.2, the reduced $\prec_{\text{lex}}$-Gröbner basis G of $\Gamma(\mathcal{P})$ contains a polynomial with $\prec_{\text{lex}}$-leading term $y^{v_x(\mathcal{P})}$. Thus $\deg_y y^\eta \operatorname{rem} G < v_x(\mathcal{P})$ for any η, and setting $g_i = y^{\eta_i} - (y^{\eta_i} \operatorname{rem} G)$ yields an $\boldsymbol{\eta}$-reshaper as long as $v_x(\mathcal{P}) \leq 2\eta_i - \eta_{i-1} + 1$ for all i. □

Corollary 3.6. *Let $\mathcal{P} \subseteq \mathbb{K}^2$ be a point set of cardinality n and $a, b \in \mathbb{Z}_{>0}$ with $n > a > b \geq v_x(\mathcal{P})$. Then there is an (a, b)-reshaping sequence $\boldsymbol{\eta}$ which satisfies the condition of Lemma 3.5 and has length $k \leq \log_{3/2}(a) + 1 \in O(\log(a))$.*

Proof. Let $v = \nu_x(\mathcal{P}) - 1$ and let $\boldsymbol{\eta}' = (\eta'_0, \dots, \eta'_k)$ be any $(a-v, b-v)$-reshaping sequence with $k \leq \log_{3/2}(a-v)+1$. Now let $\boldsymbol{\eta} = (\eta_0, \dots, \eta_k)$ be defined by $\eta_i = \eta'_i + v$ for $i = 0, \dots, k$. Then, $\boldsymbol{\eta}$ is an (a,b)-reshaping sequence. Indeed, clearly the endpoints are correct and $\eta_{i-1} > \eta_i$ for $i = 1, \dots, k$; moreover,

$$\eta_i = \eta'_i + v \geq \lfloor \tfrac{2}{3}\eta'_{i-1} \rfloor + v = \lfloor \tfrac{2}{3}\eta_{i-1} + \tfrac{1}{3}v \rfloor \geq \lfloor \tfrac{2}{3}\eta_{i-1} \rfloor .$$

To conclude, we use $2\eta'_i - \eta'_{i-1} \geq 0$ as mentioned above to observe that $2\eta_i - \eta_{i-1} + 1 = 2\eta'_i - \eta'_{i-1} + v + 1 \geq v + 1 = \nu_x(\mathcal{P})$. □

4 MULTI-POINT EVALUATION

In this section we use reshaping for MPE with precomputation; i.e. given a point set $\mathcal{P} \subset \mathbb{K}^2$ upon which we are allowed to perform precomputation, and a polynomial $f \in \mathbb{K}[x,y]$ which is assumed to be received at online time, compute $f(P)$ for all $P \in \mathcal{P}$. Algorithm 2 deals with the case $\nu_x(\mathcal{P}) = 1$, which we reduce to an instance of univariate MPE using Reshape. The cost of Algorithm 2 follows directly from Theorem 3.2 and Lemma 3.4.

Algorithm 2 MPE-DistinctX$_{d,\boldsymbol{\eta},\mathcal{P}}(f)$

Preinput: $d \in \mathbb{Z}_{>0}$; a $(d,1)$-reshaping sequence $\boldsymbol{\eta}$; a point set $\mathcal{P} = \{(\alpha_i, \beta_i)\}_{i=1}^n \subset \mathbb{K}^2$ with the α_i's pairwise distinct.
Precomputation:
a: $\boldsymbol{g} \leftarrow \boldsymbol{\eta}$-reshaper for $\Gamma(\mathcal{P})$
Input: $f \in \mathbb{K}[x,y]$ with $\deg_y f < d$.
Output: $\big(f(\alpha_1,\beta_1), \dots, f(\alpha_n,\beta_n)\big) \in \mathbb{K}^n$.
1: $\hat{f} \leftarrow$ Reshape$(f, \boldsymbol{\eta}, \boldsymbol{g}) \in \mathbb{K}[x]$
2: **return** $\big(\hat{f}(\alpha_1), \dots, \hat{f}(\alpha_n)\big) \in \mathbb{K}^n$ ▷ *univariate MPE*

Theorem 4.1. *Algorithm 2 is correct. If $\mathcal{P}$ is $\boldsymbol{\eta}$-balanced and $\boldsymbol{\eta}$ has length in $O(\log(n))$, the complexity is $\tilde{O}(\deg_x f \deg_y f + n)$.*

Algorithm 2 can easily be extended to the case where $\nu_x(\mathcal{P}) > 1$ by partitioning $\mathcal{P}$ into $\nu_x(\mathcal{P})$ many subsets, each having x-valency one. This approach also has quasi-linear complexity in the input size as long as $\nu_x(\mathcal{P}) \ll n$, or more precisely if $n\nu_x(\mathcal{P}) \in \tilde{O}(n)$.

When $\nu_x(\mathcal{P})$ is large, this strategy is costly, and we proceed instead by shearing the point set, as proposed by Nüsken and Ziegler [17], so that the resulting point set has distinct x-coordinates: by taking $\theta \in \mathbb{L} \setminus \mathbb{K}$, where $\mathbb{L}$ is an extension field of $\mathbb{K}$ of degree 2, we apply the map $(\alpha, \beta) \mapsto (\alpha + \theta\beta, \beta)$ to each element of $\mathcal{P}$. The problem then reduces to evaluating $\bar{f} = f(x - \theta y, y)$ at the sheared points. To compute $\bar{f}$, [17] provides an algorithm with complexity $O(\mathsf{M}(d_x(d_x+d_y))\log(d_x))$ using a univariate Taylor shift of f seen as a polynomial in x over the ring $\mathbb{L}[y]$. Algorithm 3 describes an algorithm for this task which improves the cost on the logarithmic level, by using Taylor shifts of the homogeneous components of f.

Algorithm 3 ShearPoly(f, a, b)

Input: $f = \sum_{i=0}^{d_x} \sum_{j=0}^{d_y} f_{i,j} x^i y^j \in \mathbb{L}[x,y]$; $a \in \mathbb{L}$ and $b \in \mathbb{L}$.
Output: $f(ax + by, y)$.
1: **for** $t = 0, \dots, d_x + d_y$ **do**
2: $h_t \leftarrow \sum_{i=\max(0,\, t-d_y)}^{\min(t,\, d_x)} f_{i,t-i} z^i \in \mathbb{L}[z]$
3: $s_t \leftarrow h_t(az + b)$ ▷ *Taylor shift*
4: **return** $\sum_{t=0}^{d_x+d_y} y^t s_t(x/y)$

Theorem 4.2. *Algorithm 3 correctly computes $f(ax+by, y)$, which has x-degree at most d_x and y-degree at most $d_x + d_y$, at a cost of $O((d_x+d_y)\mathsf{M}(d_x)\log(d_x)) \subset \tilde{O}(d_x(d_x+d_y))$ operations in $\mathbb{L}$.*

Proof. Observe that $y^t h_t(x/y)$ is the homogeneous component of f of degree t, and in particular $f = \sum_{t=0}^{d_x+d_y} y^t h_t(x/y)$. Thus

$$f(ax+by, y) = \sum_{t=0}^{d_x+d_y} y^t h_t\left(\frac{ax+by}{y}\right) = \sum_{t=0}^{d_x+d_y} y^t s_t(x/y),$$

hence the correctness. The degree bounds on the output are straightforward. As for complexity, only Line 3 uses arithmetic operations. First, scaling $h_t(z) \mapsto h_t(az)$ costs $O(d_x)$ operations in $\mathbb{L}$, since $\deg h_t \leq d_x$; then the Taylor shift $h_t(az) \mapsto h_t(az+b)$ costs $O(\mathsf{M}(d_x)\log(d_x))$ operations in $\mathbb{L}$ according to [29, Fact 2.1(iv)]. Summing over the $d_x + d_y$ iterations yields the claimed bound. □

This leads to Algorithm 4, where $\mathcal{P}$ may have repeated α_i's.

Algorithm 4 MPE-Shear$_{d,\boldsymbol{\eta},\mathcal{P}}(f)$

Preinput: an integer $d \in \mathbb{Z}_{>0}$; a $(d,1)$-reshaping sequence $\boldsymbol{\eta}$; a point set $\mathcal{P} = \{(\alpha_i, \beta_i)\}_{i=1}^n \subset \mathbb{K}^2$.
Precomputation:
a: $(\mathbb{L}, \theta) \leftarrow$ degree 2 extension of $\mathbb{K}$, element $\theta \in \mathbb{L} \setminus \mathbb{K}$
b: $\bar{\mathcal{P}} \leftarrow \{(\alpha_i + \theta\beta_i, \beta_i)\}_{i=1}^n \subset \mathbb{L}^2$
c: Do the precomputation of MPE-DistinctX$_{d,\boldsymbol{\eta},\bar{\mathcal{P}}}$
Input: $f \in \mathbb{K}[x,y]$ with $\deg_x f + \deg_y f < d$.
Output: $\big(f(\alpha_1,\beta_1), \dots, f(\alpha_n,\beta_n)\big) \in \mathbb{K}^n$.
1: $\bar{f} \leftarrow$ ShearPoly$(f, 1, -\theta)$ ▷ $\bar{f} = f(x - \theta y, y)$
2: **return** MPE-DistinctX$_{d,\boldsymbol{\eta},\bar{\mathcal{P}}}(\bar{f})$

Theorem 4.3. *Algorithm 4 is correct. If $\bar{\mathcal{P}}$ is $\boldsymbol{\eta}$-balanced and $\boldsymbol{\eta}$ has length in $O(\log(n))$, its complexity is $\tilde{O}(\deg_x f(\deg_x f + \deg_y f) + n)$.*

5 INTERPOLATION

In this section we use reshaping for the interpolation problem in a similar setting: we input a point set $\mathcal{P}$ for precomputation, and input interpolation values at online time. When $\mathcal{P}$ is appropriately balanced, we solve the interpolation problem in quasi-linear time (see Algorithm 5). The strategy is to first shear the point set to have unique y-coordinates and compute $u \in \mathbb{L}[y]$ which interpolates the values on the sheared y-coordinates. Then we reshape this into $r \in \mathbb{L}[x,y]$ with x- and y-degrees roughly $\sqrt{n}$. Shearing back this polynomial to interpolate the original point set is now in quasi-linear time; a last reshaping allows us to meet the target y-degree.

Theorem 5.1. *Algorithm 5 is correct and has complexity*

$$\tilde{O}\Big(k_1 n + k_2\Big(\sqrt{n} + \sum_{j=1}^{k_1} \deg_x g_{1,j}\Big)^2 + \sum_{\ell=1}^{2} k_\ell \sum_{j=1}^{k_\ell} \eta_{\ell,k} \deg_x g_{\ell,j}\Big).$$

If $\bar{\mathcal{P}}$ is $\boldsymbol{\eta}_1$-balanced and $\mathcal{P}$ is $\boldsymbol{\eta}_2$-balanced, and both $\boldsymbol{\eta}_1$ and $\boldsymbol{\eta}_2$ have length in $O(\log n)$, then the complexity is $\tilde{O}(n)$.

Proof. First note that a reshaping sequence of length $O(\log n)$ and satisfying the preinput constraints exists, due to Corollary 3.6 and the assumption $d \geq \nu_x(\mathcal{P})$. For correctness, observe that all points in $\bar{\mathcal{P}}$ have pairwise distinct y-coordinates, so computing u makes sense. Viewing u as an element of $\mathbb{L}[x,y]$ with $\deg_x u = 0$, we

Algorithm 5 $\textsc{Interpolate}_{d,\boldsymbol{\eta},\mathcal{P}}(\boldsymbol{\gamma})$

Preinput: an integer $d \in \mathbb{Z}_{>0}$; an (n,d)-reshaping sequence $\boldsymbol{\eta} = (\eta_i)_{i=0}^{k}$ such that $\eta_{k_1} = \lfloor\sqrt{n}\rfloor$ for some k_1; a point set $\mathcal{P} = \{(\alpha_i,\beta_i)\}_{i=1}^{n} \subseteq \mathbb{K}^2$ such that $\nu_x(\mathcal{P}) \le d \le \lfloor\sqrt{n}\rfloor + 1$ and $\nu_x(\mathcal{P}) \le \min_{1\le i\le k}(2\eta_i - \eta_{i-1} + 1)$.
Precomputation:
a: $\boldsymbol{\eta}_1 \leftarrow (\eta_i)_{i=0}^{k_1}$ and $\boldsymbol{\eta}_2 \leftarrow (\eta_i)_{i=k_1}^{k}$
b: $(\mathbb{L},\theta) \leftarrow \begin{cases}(\mathbb{K},0) \text{ if } \nu_y(\mathcal{P}) = 1\\ \text{degree 2 extension of } \mathbb{K}, \theta \in \mathbb{L}\setminus\mathbb{K} \text{ otherwise}\end{cases}$
c: $\bar{\mathcal{P}} \leftarrow \{(\alpha_i,\bar{\beta}_i)\}_{i=1}^{n}$, where $\bar{\beta}_i = \theta\alpha_i + \beta_i$
d: $\boldsymbol{g}_1 \leftarrow \boldsymbol{\eta}_1$-reshaper for $\bar{\mathcal{P}}$
e: $\boldsymbol{g}_2 \leftarrow \boldsymbol{\eta}_2$-reshaper for $\mathcal{P}$
Input: Interpolation values $\boldsymbol{\gamma} = (\gamma_i)_{i=1}^{n} \in \mathbb{K}^n$.
Output: $f \in \mathbb{K}[x,y]$ satisfying $f(\alpha_i,\beta_i) = \gamma_i$ for $i = 1,\dots,n$, $\deg_y f < d$ and $\deg_x f \le \lfloor\sqrt{n}\rfloor + \sum_{g\in\boldsymbol{g}_1\cup\boldsymbol{g}_2} \deg_x g$.
1: $u \in \mathbb{L}[y]$ with $\deg u < n$ and $u(\bar{\beta}_i) = \gamma_i$ for $i = 1,\dots,n$
2: $r \leftarrow \textsc{Reshape}(u, \boldsymbol{\eta}_1, \boldsymbol{g}_1) \in \mathbb{L}[x,y]$
3: $s \leftarrow r(x, \theta x + y)$ ▷ *using* ShearPoly
4: Write $s = s_1 + \theta s_2$, where $s_1, s_2 \in \mathbb{K}[x,y]$
5: **return** $\textsc{Reshape}(s_1, \boldsymbol{\eta}_2, \boldsymbol{g}_2) \in \mathbb{K}[x,y]$

have $u(\alpha_i,\bar{\beta}_i) = \gamma_i$. By Theorem 3.2 then r has the same evaluations and $\deg_y r < \lfloor\sqrt{n}\rfloor$ and $\deg_x r \le \sum_{i=1}^{k_1} \deg_x g_{1,i}$.

Then, in both cases $\nu_y(\mathcal{P}) = 1$ and $\nu_y(\mathcal{P}) > 1$, we have

$$\gamma_i = r(\alpha_i,\bar{\beta}_i) = s(\alpha_i,\beta_i) = s_1(\alpha_i,\beta_i) + \theta s_2(\alpha_i,\beta_i)$$

for $i = 1,\dots,n$. Since $s_1, s_2 \in \mathbb{K}[x,y]$ and all γ_i's are in $\mathbb{K}$, we get $s_2(\alpha_i,\beta_i) = 0$ and $s_1(\alpha_i,\beta_i) = \gamma_i$ for $i = 1,\dots,n$. We also then have that $\deg_y s_1 \le \deg_y s < \lfloor\sqrt{n}\rfloor$ and

$$\deg_x s_1 \le \deg_x s \le \deg_y r + \deg_x r \le \lfloor\sqrt{n}\rfloor + \textstyle\sum_{j=1}^{k_1} \deg_x g_{1,j}\ .$$

Thus, by Theorem 3.2 again, the output f is such that $f(\alpha_i,\beta_i) = \gamma_i$ for $i = 1,\dots,n$, and $\deg_y f < d$, and

$$\deg_x f \le \lfloor\sqrt{n}\rfloor + \textstyle\sum_{j=1}^{k_1} \deg_x g_{1,j} + \sum_{j=1}^{k_2} \deg_x g_{2,j}\ .$$

The complexity bound gathers the calls to Algorithms 1 and 3, and the relaxed cost assuming balancedness is due to Lemma 3.4. □

6 MODULAR COMPOSITION

We now turn to the following modular composition problem: given $M, A \in \mathbb{K}[x]$ with $n := \deg_x M > \deg_x A$, and $f \in \mathbb{K}[x,y]$, compute

$$f(x, A(x)) \operatorname{rem} M(x) \in \mathbb{K}[x]\ . \tag{2}$$

We consider the variant of the problem where M and A are available for precomputation. Computing (2) is tantamount to computing the unique element of $(f + I) \cap \mathbb{K}[x]$ of degree less than n, for the ideal $I = \langle M, y - A\rangle \subseteq \mathbb{K}[x,y]$. One can thus see this as a reshaping task: given f of some y-degree, reshape it to a polynomial of y-degree 0 while keeping it fixed modulo I: this is formalised as Algorithm 6.

Like for point sets above, if $\boldsymbol{\eta} = (\eta_i)_{i=0}^{k}$ is a reshaping sequence, we say that $I = \langle M, y - A\rangle$ is $\boldsymbol{\eta}$-balanced if there exists an $\boldsymbol{\eta}$-reshaper $\boldsymbol{g} = (g_i)_{i=1}^{k}$ for I such that $\deg_x g_i \le \lfloor \frac{n}{2\eta_i - \eta_{i-1} + 1} \rfloor + 1$.

Theorem 6.1. *Algorithm 6 is correct. If $\langle M, y - A\rangle$ is $\boldsymbol{\eta}$-balanced and $\boldsymbol{\eta}$ has length in $O(\log(n))$, the complexity is $\tilde{O}(\deg_x f \deg_y f + n)$.*

Algorithm 6 $\textsc{ModComp}_{d,\boldsymbol{\eta},M,A}(f)$

Preinput: $d \in \mathbb{Z}_{>0}$; a $(d,1)$-reshaping sequence $\boldsymbol{\eta}$; polynomials $M, A \in \mathbb{K}[x]$ with $n := \deg_x M > \deg_x A$.
Precomputation:
a: $\boldsymbol{g} \leftarrow \boldsymbol{\eta}$-reshaper for $\langle M, y - A\rangle$
Input: $f \in \mathbb{K}[x,y]$ with $\deg_y f < d$.
Output: $f(x, A) \operatorname{rem} M \in \mathbb{K}[x]$.
1: $\hat{f} \leftarrow \textsc{Reshape}(f, \boldsymbol{\eta}, \boldsymbol{g}) \in \mathbb{K}[x]$
2: **return** $\hat{f} \operatorname{rem} M$ ▷ *univariate division with remainder*

7 PRECOMPUTING RESHAPERS

7.1 Reshapers for general ideals

Here we describe Algorithm 7 for precomputing reshapers for any zero-dimensional ideal $I \subseteq \mathbb{K}[x,y]$, given a $\prec_{\text{lex}}$-Gröbner basis of I. It operates through the $\mathbb{K}[x]$-module $I_\delta := \{f \in I \mid \deg_y f < \delta\}$, so we first expound the relation between this and I as a corollary of Lazard's structure theorem on bivariate $\prec_{\text{lex}}$-Gröbner bases [12].

Corollary 7.1. *Let $G = \{b_0,\dots,b_s\} \subset \mathbb{K}[x,y]$ be a minimal $\prec_{\text{lex}}$-Gröbner basis defining an ideal $I = \langle G\rangle$. For $\delta \in \mathbb{Z}_{>0}$, let $I_\delta = \{f \in I \mid \deg_y f < \delta\}$, let $\hat{s} = \max\{i \mid \deg_y b_i < \delta,\ 0 \le i \le s\}$, let $d_i = \deg_y b_i$ for $0 \le i \le \hat{s}$ and $d_{\hat{s}+1} = \delta$. Then I_δ is a $\mathbb{K}[x]$-submodule of $\mathbb{K}[x,y]_{\deg_y < \delta}$ which is free of rank $\delta - d_0$ and admits the basis $\{y^j b_i \mid 0 \le j < d_{i+1} - d_i, 0 \le i \le \hat{s}\}$.*

A proof is given in appendix. We will use the following $\mathbb{K}[x]$-module isomorphism which converts between bivariate polynomials of bounded y-degree and vectors over $\mathbb{K}[x]$: for any $\delta \in \mathbb{Z}_{>0}$,

$$\phi_\delta : f = \textstyle\sum_{j=0}^{\delta-1} f_j(x) y^j \in \mathbb{K}[x,y] \mapsto [f_0,\dots,f_{\delta-1}] \in \mathbb{K}[x]^{1\times\delta}\ .$$

If I is zero-dimensional then in Corollary 7.1 we have $d_0 = 0$ and I_δ has rank δ. Any basis B of I_δ can be represented as a nonsingular matrix $M_B \in \mathbb{K}[x]^{\delta\times\delta}$ whose rows are $\phi_\delta(B)$. Then, $\Delta(I_\delta) := \deg\det(M_B)$ does not depend on the choice of B since all bases of I_δ have the same determinant up to scalar multiplication.

In this section, we use the *Popov form* [20], which can be defined for any matrix and with "shifts"; here we only need the unshifted, nonsingular square case.

Definition 7.2. For any row vector $\boldsymbol{v} \in \mathbb{K}[x]^{1\times\delta}$ its *row degree* denoted $\deg \boldsymbol{v}$ is the maximal degree among its entries. The *pivot* of $\boldsymbol{v}$ is the rightmost entry of $\boldsymbol{v}$ with degree $\deg \boldsymbol{v}$. A nonsingular matrix $P = [p_{ij}] \in \mathbb{K}[x]^{\delta\times\delta}$ is in *Popov form* if p_{ii} is the pivot of the ith row, is monic, and $\deg p_{ii} > \deg p_{ji}$ for any $j \ne i$.

For a (free) $\mathbb{K}[x]$-submodule $\mathcal{M} \subset \mathbb{K}[x]^{1\times\delta}$ of rank δ, we identify a basis of $\mathcal{M}$ as the rows of a nonsingular matrix in $\mathbb{K}[x]^{\delta\times\delta}$. Any such $\mathcal{M}$ has a unique basis $P \in \mathbb{K}[x]^{\delta\times\delta}$ in Popov form, which we call *the Popov basis of* $\mathcal{M}$. It has minimal row degrees in the following sense: if $N \in \mathbb{K}[x]^{\delta\times\delta}$ is another basis of $\mathcal{M}$, there is a bijection ψ from the rows of P to the rows of N such that $\deg \boldsymbol{p} \le \deg \psi(\boldsymbol{p})$ for any row $\boldsymbol{p}$ of P. The Popov basis satisfies $\Delta(\mathcal{M}) = \Delta(P) = |\operatorname{cdeg}(P)|$, using the following notation: the sum of the entries of a tuple $\boldsymbol{t} \in \mathbb{Z}_{\ge 0}^{\delta}$ is denoted $|\boldsymbol{t}|$; the column degree of a matrix $B \in \mathbb{K}[x]^{\delta\times\delta}$ is $\operatorname{cdeg}(B) = (d_i)_{i=1}^{\delta} \in \mathbb{Z}_{\ge 0}^{\delta}$, with d_i the largest degree in the ith column of B (for a zero column, $d_i = 0$).

The next result allows us to compute Popov forms efficiently.

Proposition 7.3 ([16]). *There is an algorithm which inputs a nonsingular matrix $B \in \mathbb{K}[x]^{\delta\times\delta}$ and outputs the Popov basis of the $\mathbb{K}[x]$-row space of B using $\tilde{O}(\delta^{\omega-1}|\mathrm{cdeg}(B)|)$ operations in $\mathbb{K}$, assuming that $\delta \in O(|\mathrm{cdeg}(B)|)$.*

Since Popov forms are "column reduced", they are well suited for matrix division with remainder [10, Thm. 6.3-15]: if $P \in \mathbb{K}[x]^{\delta\times\delta}$ is the Popov basis of $\mathcal{M}$, then for any $\boldsymbol{v} \in \mathbb{K}[x]^{1\times\delta}$ there is a unique $\boldsymbol{u} \in \boldsymbol{v} + \mathcal{M}$ such that $\mathrm{cdeg}(\boldsymbol{u}) < \mathrm{cdeg}(P)$ entrywise; we denote $\boldsymbol{u} = \boldsymbol{v} \operatorname{rem} P$. Furthermore, $\boldsymbol{u}$ has minimal row degree among all vectors in $\boldsymbol{v} + \mathcal{M}$. Such remainders can be computed efficiently:

Proposition 7.4 ([16]). *There is an algorithm which inputs a Popov form $P \in \mathbb{K}[x]^{\delta\times\delta}$ and $\boldsymbol{v} \in \mathbb{K}[x]^{1\times\delta}$ such that $\mathrm{cdeg}(\boldsymbol{v}) < \mathrm{cdeg}(P) + (\Delta(P), \dots, \Delta(P))$ entrywise, and outputs $\boldsymbol{v} \operatorname{rem} P$ using $\tilde{O}(\delta^{\omega-1}\,\Delta(P))$ operations in $\mathbb{K}$, assuming that $\delta \in O(\Delta(P))$.*

Algorithm 7 ComputeReshaper(G, η, δ)

Input: A reduced $\prec_{\mathrm{lex}}$-Gröbner basis $G = \{b_0, \dots, b_s\} \subset \mathbb{K}[x, y]$, sorted by increasing y-degree, for a zero-dimensional ideal I (hence $b_0 \in \mathbb{K}[x]$); $\eta, \delta \in \mathbb{Z}_{>0}$ with $\delta < \eta$.
Output: If no polynomial in $y^\eta + I$ has y-degree $< \delta$, "Fail"; otherwise, $g = y^\eta - \hat{g} \in I$ with $\deg_y \hat{g} < \delta$ and $\deg_x \hat{g}$ minimal.

1: $R \leftarrow y^\eta \operatorname{rem} G$
2: **if** $\deg_y R \geq \delta$ **then return** "Fail"
3: $B_\delta \leftarrow$ basis of $I_\delta = \{f \in I \mid \deg_y f < \delta\}$ as in Corollary 7.1
4: $B \in \mathbb{K}[x]^{\delta\times\delta} \leftarrow$ row-wise applying ϕ_δ to elements of B_δ
5: $P \in \mathbb{K}[x]^{\delta\times\delta} \leftarrow$ Popov basis of I_δ from the basis B
6: $\hat{g} \leftarrow -\phi_\delta^{-1}(\phi_\delta(R) \operatorname{rem} P) \in \mathbb{K}[x, y]$
7: **return** $g = y^\eta - \hat{g} \in \mathbb{K}[x, y]$

Theorem 7.5. *Algorithm 7 is correct. Assuming $\eta \in O(\Delta(I_\delta))$, it costs $\tilde{O}(\delta^{\omega-1}\,\Delta(I_\delta) + \eta s \deg_x b_0)$ operations in $\mathbb{K}$.*

Proof. Since G is a $\prec_{\mathrm{lex}}$-Gröbner basis, if $y^\eta + I$ contains a polynomial of y-degree less than δ, then $\deg_y(y^\eta \operatorname{rem} G) \leq \delta$ and the algorithm does not fail at Line 2.

For correctness of the output, observe that $y^\eta - R \in I$ so satisfactory $g = y^\eta - \tilde{g}$ all have $\tilde{g} \in R + I_\delta$. Now, $\hat{g}$ of Line 6 is clearly in $R + I_\delta$ since P is the Popov basis of I_δ, but also $\hat{g}$ has minimal x-degree in the coset $R + I_\delta$. Hence among all g of the correct form, the algorithm returns that of minimal x-degree.

For complexity, work is done in Lines 1, 5 and 6. Since G is reduced, $\deg_x b_0 > \dots > \deg_x b_s$. Therefore the diagonal entries in B are dominant in their columns and $|\mathrm{cdeg}\, B| = \Delta(B) = \Delta(P) = \Delta(I_\delta)$. For Line 1, we use the algorithm of [23] with cost $\tilde{O}(\eta s \deg_x b_0)$, see Lemma A.2. Line 5 costs $\tilde{O}(\delta^{\omega-1}|\mathrm{cdeg}\, B|)$ by Proposition 7.3 and Line 6 costs $\tilde{O}(\delta^{\omega-1}\,\Delta(P))$ since $\deg_x R < \deg_x b_0 < \Delta(P)$. □

7.2 Reshapers for the considered problems

We turn to obtaining the reduced $\prec_{\mathrm{lex}}$-Gröbner basis of $\Gamma(\mathcal{P})$. We will consider the $\mathbb{K}[x]$-submodule $\Gamma_m(\mathcal{P}) = \Gamma(\mathcal{P}) \cap \mathbb{K}[x, y]_{\deg_y < m}$ which by Lemma 2.2 and Corollary 7.1 is free and of rank m. To obtain a $\prec_{\mathrm{lex}}$-Gröbner basis, our approach is to first compute the Hermite basis of $\Gamma_m(\mathcal{P})$. This is the unique basis whose corresponding matrix $H \subset \mathbb{K}[x]^{m\times m}$ is lower triangular, with each diagonal entry monic and strictly dominating the degrees in its column.

Lemma 7.6. *For any point set $\mathcal{P} \subseteq \mathbb{K}^2$ and any $m > \nu_x(\mathcal{P})$, we have $\Gamma(\mathcal{P}) = \langle \Gamma_m(\mathcal{P}) \rangle$ and $\Delta(\Gamma_m(\mathcal{P})) = |\mathcal{P}|$.*

Proof. By Lemma 2.2 the elements of the reduced $\prec_{\mathrm{lex}}$-Gröbner basis of $\Gamma(\mathcal{P})$ have y-degree at most $\nu_x(\mathcal{P})$, implying the first claim. Further, this means the quotient $\mathbb{K}[x, y]/\Gamma(\mathcal{P})$ is isomorphic to the quotient of modules $\mathbb{K}[x, y]_{\deg_y < m}/\Gamma_m(\mathcal{P})$. It is a basic property of zero-dimensional varieties that the $\mathbb{K}$-dimension of the former is the number of points in $\mathcal{P}$, which is hence also the $\mathbb{K}$-dimension of the latter. This dimension is $\Delta(\Gamma_m(\mathcal{P}))$ by [16, Lem. 2.3]. □

Proposition 7.7. *There is an algorithm which inputs $\mathcal{P} \subset \mathbb{K}^2$ and outputs the reduced $\prec_{\mathrm{lex}}$-Gröbner basis of $\Gamma(\mathcal{P})$ and has complexity $\tilde{O}(\nu_x(\mathcal{P})^{\omega-1}|\mathcal{P}|)$.*

Proof. Let $\Gamma = \Gamma(\mathcal{P})$, $\Gamma_m = \Gamma_m(\mathcal{P})$, and $m = \nu_x(\mathcal{P}) + 1$. We first compute the Hermite basis H of $\Gamma_m(\mathcal{P})$ in time $\tilde{O}(m^{\omega-1}|\mathcal{P}|)$ using (a special case of) [9, Thm. 1.5], in which taking $\boldsymbol{s} = (0, n, \dots, (m-1)n)$ ensures that the $\boldsymbol{s}$-Popov basis P of Γ_m is the Hermite basis.

Let $G = \{g_0, \dots, g_{m-1}\} \subset \mathbb{K}[x, y]$ be given as the ϕ_m^{-1}-image of the rows of H. By Lemma 7.6 and since H is lower triangular, G is a $\prec_{\mathrm{lex}}$-Gröbner basis of Γ but not necessarily minimal. Construct $G' \subseteq G$ from G by excluding the elements $g \in G$ such that there is $g' \in G$ with $\deg_y g' < \deg_y g$ and $\deg_x(\mathrm{LC}_y(g')) \leq \deg_x(\mathrm{LC}_y(g))$, i.e. $\mathrm{LT}_{\mathrm{lex}}(g')$ divides $\mathrm{LT}_{\mathrm{lex}}(g)$. This makes G' a minimal $\prec_{\mathrm{lex}}$-Gröbner basis of Γ [4, Lem. 3 of Chap. 2 §7], and we claim it is the reduced one. Indeed, since H is in Hermite form, the selection criteria for G' ensures that for any $g \neq g'$ in G' and any term $x^i y^j$ in g', we have $i < \deg_x(\mathrm{LT}_{\mathrm{lex}}(g))$ or $j < \deg_y g$, and hence G' is reduced. Obtaining G' from H costs no arithmetic operations. □

Corollary 7.8. *Given a point set $\mathcal{P} \subseteq \mathbb{K}^2$ of cardinality n and a reshaping sequence $\boldsymbol{\eta} = (\eta_i)_{i=0}^k$ with $n \geq \eta_k$ and satisfying the condition of Lemma 3.5, then we can determine if $\mathcal{P}$ is $\boldsymbol{\eta}$-balanced and compute an $\boldsymbol{\eta}$-reshaper $\boldsymbol{g} = (g_i)_{i=1}^k$ for $\mathcal{P}$ where each element has minimal possible x-degree in complexity $\tilde{O}(k\eta_0^{\omega-1}n + \eta_0 \nu_x nk)$.*

Proof. By Proposition 7.7, computing a reduced $\prec_{\mathrm{lex}}$-Gröbner basis $G = (b_i)_{i=0}^{\nu_x}$ of $\Gamma(\mathcal{P})$ costs $\tilde{O}(\nu_x^{\omega-1}n) \subset \tilde{O}(\eta_0^{\omega-1}n)$. We then run Algorithm 7 on input $\eta = \eta_i$ and $\delta_i = 2\eta_i - \eta_{i-1} + 1 > \nu_x$ for $i = 1, \dots, k$. Lemma 7.6 ensures $\Delta(\Gamma_\delta(\mathcal{P})) = n$ for any $\delta > \nu_x$, thus the cost of each call to Algorithm 7 becomes $\tilde{O}(\eta_0^{\omega-1}n + \eta_0 \nu_x n)$. □

Corollary 7.9. *Given $M, A \in \mathbb{K}[x]$ with $n := \deg M > \deg A$ and a reshaping sequence $\boldsymbol{\eta} = (\eta_i)_{i=0}^k$ with $n \geq \eta_k$, then we can determine if $I := \langle M, y - A \rangle$ is $\boldsymbol{\eta}$-balanced and compute an $\boldsymbol{\eta}$-reshaper $\boldsymbol{g} = (g_i)_{i=1}^k$ for $\mathcal{P}$ where each element has minimal possible x-degree in complexity $\tilde{O}(k\eta_0^{\omega-1}n)$.*

Proof. For any δ, and using the notation of Algorithm 7, the basis B of I_δ is lower triangular with diagonal entries $(M, 1, \dots, 1)$. Hence $\Delta(B) = \Delta(I_\delta) = n$. Using $s = 1$ and $\deg_x b_0 = \deg_x M = n$, the cost follows from Theorem 7.5. □

8 GENERICITY

Now we show that on random input our algorithms usually have quasi-linear complexity, i.e. that random point sets are balanced and that $\langle M, y - A \rangle$ is balanced for random univariate A, M.

Lemma 8.1. *Let $\alpha_1, \ldots, \alpha_n \in \mathbb{K}$ be distinct, let $y_1, \ldots, y_n$ be new indeterminates, and consider for $s \in \mathbb{Z}_{>0}$ the matrix*

$$A_s = \left[V_s \mid DV_s \mid \ldots \mid D^{m-1}V_s\right] \in \mathbb{K}[y_1, \ldots, y_n]^{n \times ms} \quad (3)$$

where D is the diagonal matrix with entries $(y_1, \ldots, y_n)$, and $V_s = [\alpha_i^{j-1}]_{1 \le i \le n, 1 \le j \le s} \in \mathbb{K}^{n \times s}$. Then A_s has rank $\min(n, ms)$.

Proof. Note that by rank of a matrix over $\mathbb{K}[y_1, \ldots, y_n]$, we mean the rank of that matrix seen as over the field of fractions $\mathbb{K}(y_1, \ldots, y_n)$. If we specialise y_i to α_i^s for $i = 1, \ldots, n$, we obtain the Vandermonde matrix $\hat{A}_s = [\alpha_i^{j-1}]_{1 \le i \le n, 1 \le j \le ms} \in \mathbb{K}^{n \times ms}$ of the points $\alpha_1, \ldots, \alpha_n$. Since these points are distinct, $\hat{A}_s$ has full rank $\min(n, ms)$. Hence A_s must also have full rank. □

The columns of A_s can be identified to monomials $x^i y^j$ with $i < s$ and $j < m$. In particular, if $p \in \Gamma(\mathcal{P})$ is a bivariate polynomial with x-degree less than s and y-degree less than m which vanishes on a point set $\mathcal{P} = \{(\alpha_i, \beta_i)\}_{i=1}^n \subset \mathbb{K}^2$ with distinct α_i's, then the coefficients of p form a vector in the right kernel of the matrix $\hat{A}_s = (A_s)_{|y_i \to \beta_i} \in \mathbb{K}^{n \times ms}$ specializing y_i to β_i.

The next lemma determines the exact row degrees of the Popov basis $P \in \mathbb{K}[x]^{m \times m}$ of $\phi_m(\Gamma_m(\mathcal{P}))$ for a "random" point set $\mathcal{P}$, where $\Gamma_m(\mathcal{P}) = \Gamma(\mathcal{P}) \cap \mathbb{K}[x, y]_{\deg_y < m}$ as in Section 7.2.

Lemma 8.2. *Let $\alpha_1, \ldots, \alpha_n \in \mathbb{K}$ be distinct, let $\mathcal{T} \subseteq \mathbb{K}$ be a finite subset, and let $\lambda : \mathbb{K}^n \to \mathbb{K}^n$ be an affine map. For $\gamma_1, \ldots, \gamma_n \in \mathcal{T}$ chosen independently and uniformly at random, set $\mathcal{P} = \{(\alpha_i, \beta_i)\}_{i=1}^n$ where $(\beta_1, \ldots, \beta_n) = \lambda(\gamma_1, \ldots, \gamma_n)$. Let $m \in \mathbb{Z}$ with $\nu_x(\mathcal{P}) < m \le n$ and let $(d, t) = \text{QUO\_REM}(n, m)$. With probability at least $1 - 2nm/|\mathcal{T}|$, the Popov basis $P \in \mathbb{K}[x]^{m \times m}$ of $\phi_m(\Gamma_m(\mathcal{P}))$ has exactly $m - t$ rows of degree d and t rows of degree $d+1$ and in particular $\deg_x P \le d+1$.*

Proof. Let $p_1, \ldots, p_m \in \mathbb{K}[x, y]$ be the polynomials defined by the rows of P. Lemma 2.2 shows $\Delta(P) = n = \sum_{i=1}^m \deg_x p_i$.

For any $s \in \mathbb{Z}_{>0}$, let $A_s \in \mathbb{K}[y_1, \ldots, y_n]^{n \times ms}$ be as in Lemma 8.1, hence $\text{rank}(A_s) = \min(n, ms)$. Let $\hat{A}_s = (A_s)_{|y_i \to \beta_i} \in \mathbb{K}^{n \times ms}$. Taking $s = d$, as mentioned above, if $\deg_x p_i < d$ for some i, then the coefficient vector of p_i is in the right kernel of $\hat{A}_d$, and so $\text{rank}(\hat{A}_d) < \text{rank}(A_d) = md \le n$. Thus, letting $M \in \mathbb{K}[y_1, \ldots, y_n]$ be a non-zero $md \times md$ minor of A_d then $M(\beta_1, \ldots, \beta_n) = M(\lambda(\gamma_1, \ldots, \gamma_n)) = 0$; M has degree at most $m-1$ in each variable, so the total degree of M is less than nm, and since λ is affine the composition $M(\lambda(z_1, \ldots, z_n))$ also has total degree less than nm. Then, by Lemma 2.1 the probability that $M(\lambda(\gamma_1, \ldots, \gamma_n)) = 0$ is at most $nm/|\mathcal{T}|$.

Assume now that all rows of P have degree at least d. For each i such that $\deg_x p_i = d$, the coefficients of p_i form a vector in the right kernel of $\hat{A}_{d+1} \in \mathbb{K}^{n \times m(d+1)}$. By Lemma 8.1, A_{d+1} has a right kernel (over the fractions) of dimension $m(d+1) - n = m - t$. Since the rows of P are linearly independent over $\mathbb{K}[x]$, and therefore also over $\mathbb{K}$, whenever $\text{rank}(\hat{A}_{d+1}) = \text{rank}(A_{d+1})$ at most $m - t$ rows of P have x-degree d. We thus consider $N \in \mathbb{K}[y_1, \ldots, y_n]$ a non-zero $n \times n$ minor of A_{d+1}. Again N has total degree less than nm and the probability that $N(\beta_1, \ldots, \beta_n) = N(\lambda(\gamma_1, \ldots, \gamma_n)) = 0$ is at most $nm/|\mathcal{T}|$, bounding the probability that $\text{rank}(\hat{A}_{d+1}) < \text{rank}(A_{d+1})$.

Hence, with probability at least $1 - 2nd/|\mathcal{T}|$, P has all rows of degree at least d and j rows of degree exactly d with $j \le m - t$. Each of the remaining $m - j$ rows has degree at least $d + 1$, while their degrees must sum to $n - jd = md + t - jd = (m-j)d + t \le (m-j)(d+1)$. Hence each of them has degree exactly $d + 1$. □

Algorithm 7 for computing reshapers outputs a $g = y^\eta - \hat{g}$ with $\deg_y \hat{g} < \delta$ satisfying $\deg_x \hat{g} \le \deg_x P$, where P is the Popov basis of $\Gamma_\delta(\mathcal{P})$. Lemma 8.2 states that generically we can expect $\deg_x P \le \lfloor \frac{n}{\delta} \rfloor + 1$, and so when $\delta = 2\eta_i - \eta_{i-1} + 1$ in a reshaping sequence, this matches the definition of $\boldsymbol{\eta}$-balanced.

Corollary 8.3. *Let $\alpha_1, \ldots, \alpha_n \in \mathbb{K}$ be distinct, let $\mathcal{T} \subseteq \mathbb{K}$ a finite subset, and let $\lambda : \mathbb{K}^n \to \mathbb{K}^n$ be an affine map. For $\gamma_1, \ldots, \gamma_n \in \mathcal{T}$ chosen independently and uniformly at random, set $\mathcal{P} = \{(\alpha_i, \beta_i)\}_{i=1}^n$ where $(\beta_1, \ldots, \beta_n) = \lambda(\gamma_1, \ldots, \gamma_n)$. Let $\boldsymbol{\eta} = (\eta_i)_{i=0}^k$ be a reshaping sequence with $\eta_0 \le n$ and satisfying the constraint of Lemma 3.5. Then $\mathcal{P}$ is $\boldsymbol{\eta}$-balanced with probability at least $1 - n^2k/|\mathcal{T}|$.*

The above proposition directly applies to both our MPE and interpolation algorithms on random point sets with unique x-coordinates. Note that in the case of interpolation, where the point set is sheared if its y-valency is greater than one, the property of being $\boldsymbol{\eta}$-balanced is not inherited a priori by the sheared point set. The probability of being $\boldsymbol{\eta}$-balanced, however, is preserved, since the shearing acts as an affine transformation on the y-coordinates. There are many formulations depending on the type of randomness one needs over the point sets; the following is a simple example over finite fields:

Corollary 8.4. *Let $d, n \in \mathbb{Z}_{>0}$ with $d \le n$ and $\mathbb{F}_q$ be a finite field with q elements, and let $\mathcal{P} = \{(\alpha_i, \beta_i)\}_{i=1}^n \subseteq \mathbb{F}_q^2$ be chosen uniformly at random among point sets with cardinality n. Then with probability of at least $\left(1 - \frac{n^2}{q}\right)\left(1 - \frac{3n^2(\log_{3/2}(n)+1)}{q}\right)$ over the choice of $\mathcal{P}$ the following two problems can be solved with cost $\tilde{O}(n)$:*

1. *Input polynomial $f \in \mathbb{F}_q[x, y]$ such that $\deg_x f < n/d$ and $\deg_y f < d$, and output $(f(\alpha_i, \beta_i))_{i=1}^n \in \mathbb{F}_q^n$.*
2. *Input interpolation values $\boldsymbol{\gamma} = (\gamma_i)_{i=1}^n \in \mathbb{F}_q^n$, and output $f \in \mathbb{F}_q[x, y]$ satisfying $f(\alpha_i, \beta_i) = \gamma_i$ for $i = 1, \ldots, n$, as well as $\deg_y f < d$ and $\deg_x f \le cn$ for some constant c which depends only on n and d.*

Proof sketch. The probability simply bounds the probability that $\mathcal{P}$ has unique x-coordinates *and* that it is balanced in all the necessary ways. By Corollary 3.6 there is an appropriate reshaping sequence of length at most $\log_{3/2}(n) + 2$. □

We do not make a claim about the genericity in Algorithm 4: due to the shearing in that algorithm, the arguments of this section do not immediately apply. Lastly, we turn to modular composition.

Theorem 8.5. *Let $M \in \mathbb{K}[x]$ be square-free of degree n and let $\boldsymbol{\eta}$ be a $(d, 1)$-reshaping sequence of length k with $0 < d \le n$. Let $\mathcal{T} \subseteq \mathbb{K}$ be a finite subset, and let $A = \sum_{i=0}^{n-1} a_i x^{i-1} \in \mathbb{K}[x]$ where $a_0, \ldots, a_{n-1}$ are chosen independently and uniformly at random from $\mathcal{T}$. Then $\langle M, y - A \rangle$ is $\boldsymbol{\eta}$-balanced with probability at least $1 - n^2k/|\mathcal{T}|$.*

Proof. Let $\mathbb{L}$ be the splitting field of M, so $M = \prod_{i=1}^n (x - \alpha_i)$ for some pairwise distinct $\alpha_1, \ldots, \alpha_n \in \mathbb{L}$. Define the stochastic variables $\beta_i = A(\alpha_i)$ for $i = 1, \ldots, n$; the map $\lambda(a_0, \ldots, a_{n-1}) = (\beta_1, \ldots, \beta_n)$ is $\mathbb{L}$-linear. Consider $\mathcal{P} = \{(\alpha_i, \beta_i)\}_{i=1}^n \subseteq \mathbb{L}^2$. Then Corollary 8.3 implies that $\mathcal{P}$ is $\boldsymbol{\eta}$-balanced with probability at least $1 - \frac{n^2k}{|\mathcal{T}|}$. In this case, for each i there exists $g_i = y^{\eta_i} + \hat{g}_i \in I_{\mathbb{L}}$ where

$\deg_y \hat{g}_i < 2\eta_i - \eta_{i-1}$ and $\deg_x \hat{g}_i \leq \lfloor \frac{n}{2\eta_i - \eta_{i-1}+1} \rfloor + 1$, and where $I_{\mathbb{L}} = \langle M, y - A\rangle \otimes_{\mathbb{K}} \mathbb{L}$. Let $\{1, \zeta, \ldots, \zeta^{s-1}\} \subset \mathbb{L}$ be a basis of $\mathbb{L} : \mathbb{K}$ and write $g_i = g_{i,0} + \zeta g_{i,1} + \ldots + \zeta^{s-1} g_{i,s-1}$ with $g_{i,j} \in \mathbb{K}[x, y]$. Then $g_i \in I_{\mathbb{L}}$ implies that $g_{i,0} \in I$, and by the shape of g_i then $g_{i,0} = y^{\eta_i} + \hat{g}_{i,0}$ where the x- and y-degree of $\hat{g}_{i,0}$ satisfy the same bounds as $\hat{g}_i$. Then the tuple $\boldsymbol{g}_0 = (g_{1,0}, \ldots, g_{k,0}) \in \mathbb{K}[x, y]^k$ forms a balanced $\boldsymbol{\eta}$-reshaper for I. □

REFERENCES

[1] E. F. Assmus and J. D. Key. 1992. *Designs and Their Codes.* Cambridge University Press. https://doi.org/10.1017/CBO9781316529836

[2] R. P. Brent and H. T. Kung. 1978. Fast Algorithms for Manipulating Formal Power Series. *J. ACM* 25, 4 (1978), 581–595. https://doi.org/10.1145/322092.322099

[3] D. G. Cantor and E. Kaltofen. 1991. On fast multiplication of polynomials over arbitrary algebras. *Acta Informatica* 28, 7 (1991), 693–701. https://doi.org/10.1007/BF01178683

[4] D. A. Cox, J. Little, and D. O'Shea. 2015. *Ideals, Varieties, and Algorithms* (4th ed.). Springer. https://doi.org/10.1007/978-3-319-16721-3

[5] N. Coxon. 2018. Fast systematic encoding of multiplicity codes. *J. Symb. Comput.* (2018). https://doi.org/10.1016/j.jsc.2018.08.005

[6] X. Dahan. 2009. Size of Coefficients of Lexicographical Gröbner Bases: The Zero-Dimensional, Radical and Bivariate Case. In *Proceedings ISSAC 2009.* 119–126. https://doi.org/10.1145/1576702.1576721

[7] R. A. DeMillo and R. J. Lipton. 1978. A Probabilistic Remark on Algebraic Program Testing. *Inf. Process. Lett.* 7, 4 (1978), 193–195. https://doi.org/10.1016/0020-0190(78)90067-4

[8] W. Hart, F. Johansson, and S. Pancratz. 2015. FLINT: Fast Library for Number Theory. Version 2.5.2, http://flintlib.org.

[9] C.-P. Jeannerod, V. Neiger, É. Schost, and G. Villard. 2016. Fast Computation of Minimal Interpolation Bases in Popov Form for Arbitrary Shifts. In *Proceedings ISSAC 2016.* 295–302. https://doi.org/10.1145/2930889.2930928

[10] T Kailath. 1980. *Linear Systems.* Prentice-Hall.

[11] K. Kedlaya and C. Umans. 2011. Fast Polynomial Factorization and Modular Composition. *SIAM J. Comput.* 40, 6 (Jan. 2011), 1767–1802. https://doi.org/10.1137/08073408X

[12] D. Lazard. 1985. Ideal bases and primary decomposition: case of two variables. *J. Symb. Comput.* 1, 3 (1985). https://doi.org/10.1016/S0747-7171(85)80035-3

[13] F. Le Gall. 2014. Powers of tensors and fast matrix multiplication. In *Proceedings ISSAC 2014.* ACM, 296–303. https://doi.org/10.1145/2608628.2608664

[14] S. Miura. 1993. Algebraic geometric codes on certain plane curves. *Electronics and Communications in Japan (Part III: Fundamental Electronic Science)* 76, 12 (1993), 1–13. https://doi.org/10.1002/ecjc.4430761201

[15] V. Neiger, B. Salvy, É. Schost, and G. Villard. 2020. Faster modular composition (work in progress).

[16] V. Neiger and T. X. Vu. 2017. Computing Canonical Bases of Modules of Univariate Relations. In *Proceedings ISSAC 2017.* https://doi.org/10.1145/3087604.3087656

[17] M. Nüsken and M. Ziegler. 2004. Fast Multipoint Evaluation of Bivariate Polynomials. In *Proceedings ESA 2004.* https://doi.org/10.1007/978-3-540-30140-0_49

[18] V. Y. Pan. 1994. Simple Multivariate Polynomial Multiplication. *J. Symb. Comput.* 18, 3 (1994), 183–186. https://doi.org/10.1006/jsco.1994.1042

[19] M. S. Paterson and L. J. Stockmeyer. 1973. On the number of nonscalar multiplications necessary to evaluate polynomials. *SIAM J. Comput.* 2, 1 (1973), 60–66. https://doi.org/10.1137/0202007

[20] V Popov. 1970. Some properties of the control systems with irreducible matrix-transfer functions. In *Seminar on Diff. Eq. and Dyn. Sys., II.* 169–180. https://doi.org/10.1007/BFb0059934

[21] J. T. Schwartz. 1980. Fast Probabilistic Algorithms for Verification of Polynomial Identities. *J. ACM* 27, 4 (1980), 701–717. https://doi.org/10.1145/322217.322225

[22] V. Shoup. 2020. NTL: A Library for doing Number Theory, version 11.4.3. `http://www.shoup.net`.

[23] J. van der Hoeven. 2015. On the complexity of multivariate polynomial division. In *Proceedings ACA 2015.* 447–458. https://doi.org/10.1007/978-3-319-56932-1_28

[24] J. van der Hoeven and G. Lecerf. 2018. Modular composition via factorization. *J. Complexity* 48 (2018), 36–68. https://doi.org/10.1016/j.jco.2018.05.002

[25] J. van der Hoeven and G. Lecerf. 2019. Fast multivariate multi-point evaluation revisited. *J. Complexity* (2019). https://doi.org/10.1016/j.jco.2019.04.001

[26] J. van der Hoeven and É. Schost. 2013. Multi-point evaluation in higher dimensions. *Appl. Algebra Eng. Commun. Comput.* 24, 1 (2013), 37–52. https://doi.org/10.1007/s00200-012-0179-3

[27] G. Villard. 2018. On computing the resultant of generic bivariate polynomials. In *Proceedings ISSAC 2018.* 391–398. https://doi.org/10.1145/3208976.3209020

[28] G. Villard. 2018. On computing the resultant of generic bivariate polynomials. Presentation at ISSAC 2018. http://www.issac-conference.org/2018/slides/villard-computingresultant.pdf

[29] J. von zur Gathen. 1990. Functional decomposition of polynomials: the tame case. *J. Symb. Comput.* 9, 3 (1990). https://doi.org/10.1016/S0747-7171(08)80014-4

[30] J. von zur Gathen and J. Gerhard. 2013. *Modern Computer Algebra* (3rd ed.). Cambridge University Press. https://doi.org/10.1017/CBO9781139856065

[31] R. Zippel. 1979. Probabilistic algorithms for sparse polynomials. In *Proceedings EUROSAM'79.* 216–226. https://doi.org/10.1007/3-540-09519-5_73

APPENDIX

Corollary A.1 (of [12]). *Let $G = \{b_0, \ldots, b_s\} \subset \mathbb{K}[x, y]$ be a minimal $\prec_{\text{lex}}$-Gröbner basis, sorted according to $\prec_{\text{lex}}$. Then*

(1) $\deg_y b_0 < \ldots < \deg_y b_s$*; and*
(2) $\mathsf{LC}_y(b_s) \mid \mathsf{LC}_y(b_{s-1}) \mid \cdots \mid \mathsf{LC}_y(b_0)$.

Proof of Corollary 7.1. Since I is an ideal of $\mathbb{K}[x, y]$ and $I_\delta = I \cap \mathbb{K}[x, y]_{\deg_y < \delta}$, then I_δ is a $\mathbb{K}[x]$-submodule of $\mathbb{K}[x, y]_{\deg_y < \delta}$. Let $\mathcal{B}$ denote the (claimed) basis in the corollary. Clearly $\mathcal{B} \subseteq I_\delta$, and the elements of $\mathcal{B}$ all have different y-degree and so are $\mathbb{K}[x]$-linearly independent. Also $|\mathcal{B}| = \delta - d_0$, so if $\mathcal{B}$ generates I_δ then it is a basis of it and the rank of I_δ is $\delta - d_0$. It remains to show that $\mathcal{B}$ generates I_δ, so take some $f \in I_\delta$. Since $f \in I$ the multivariate division algorithm using G and the order $\prec_{\text{lex}}$ results in $q_0, \ldots, q_s \in \mathbb{K}[x, y]$ such that $f = q_0 b_0 + \ldots + q_s b_s$ with $\deg_y q_i \leq \deg_y f - \deg_y b_i$. Since $\deg_y f < \delta$ this means $q_{\hat{s}+1} = \ldots = q_s = 0$. Say that in each iteration of the division algorithm, we use the greatest index i for which $\mathsf{LT}_{\text{lex}}(b_i)$ divides the leading term of the current remainder. Thus no term of $q_i b_i$ is divisible by $\mathsf{LT}_{\text{lex}}(b_{i+1})$ for any $i < s$. But by Corollary A.1 then $\mathsf{LC}_y(b_{i+1})$ divides $\mathsf{LC}_y(b_i)$, and so if $\deg_y(q_i b_i) \geq \deg_y b_{i+1}$ then $\mathsf{LT}_{\text{lex}}(b_{i+1}) \mid \mathsf{LT}_{\text{lex}}(q_i b_i)$. Consequently $\deg_y q_i < \deg_y b_{i+1} - \deg_y b_i$, and therefore f is a $\mathbb{K}[x]$-linear combination of the elements of $\mathcal{B}$. □

Lemma A.2. *There is an algorithm which inputs a $\prec_{\text{lex}}$-Gröbner basis $G = [b_0, \ldots, b_s] \subseteq \mathbb{K}[x, y]$ with $\deg_y b_0 = 0$, and a polynomial $f \in \mathbb{K}[x, y]$, and outputs $f \operatorname{rem} G$ in time $\tilde{O}(|G| d_x(\deg_y f))$, where $d_x = \max(\deg_x f, \deg_x b_0)$.*

Proof. This is a special case of [23]: the multivariate division algorithm computes $q_0, \ldots, q_s, R \in \mathbb{K}[x, y]$ such that $f = q_0 b_0 + \ldots + q_s b_s + R$ with $R = f \operatorname{rem} G$, and the cost of the algorithm can be bounded as

$$\textstyle\sum_{i=0}^{s} \deg_x^\circ(q_i b_i) \deg_y^\circ(q_i b_i) + \deg_x^\circ(R) \deg_y^\circ(R)\,,$$

where $\deg_x^\circ(\cdot)$ denotes an a priori upper bound on the x-degree, and similarly for $\deg_y^\circ(\cdot)$. Since G is a $\prec_{\text{lex}}$-Gröbner basis, then $\deg_y^\circ(q_i b_i) \leq \deg_y f$ and $\deg_y^\circ(R) \leq \deg_y f$. For the x-degrees, note that in an iteration of the division algorithm where $b_i, i > 0$ is used, then $\deg_x \tilde{R} < \deg_x b_0$, where $\tilde{R}$ is the current remainder, since otherwise the algorithm would have reduced by b_0 as $\deg_y b_0 = 0$. Hence $\deg_x(q_i) \leq \deg_x(q_i \mathsf{LT}_{\text{lex}}(b_i)) < \deg_x b_0$ and so $\deg_x^\circ(q_i b_i) \leq 2 \deg_x b_0$. Similarly, $\deg_x^\circ(R) < \deg_x b_0$. Left is only $\deg_x^\circ(q_0 b_0)$: since $q_0 b_0 = f - q_1 b_1 - \ldots - q_s b_s - R$, then $\deg_x(q_0 b_0) \leq \max(\deg_x f,\ 2 \deg_x b_0)$. □

Conditional Lower Bounds on the Spectrahedral Representation of Explicit Hyperbolicity Cones

Rafael Oliveira
rafael@uwaterloo.ca
University of Waterloo
Waterloo, Ontario

ABSTRACT

Over the past decade there has been growing interest on characterizing which convex cones over $\mathbb{R}^n$ are spectrahedral, that is, are a linear section of the cone of positive semidefinite matrices. This interest is largely motivated by applications in control theory, optimization and combinatorics. One particular class of convex cones of interest is the class of hyperbolicity cones, where the (still open) Generalized Lax Conjecture states that every hyperbolicity cone is spectrahedral. Recent works [1, 2] have established that the hyperbolicity cones of the elementary symmetric polynomials and the homogeneous multivariate matching polynomial are spectrahedral, but the question of whether there exists an efficient spectrahedral representation for such cones remains open. Previous work [11] has provided exponential lower bounds on the spectrahedral representation of *non-explicit* hyperbolicity cones which are known to be spectrahedral. The current best lower unconditional bounds for *explicit cones* are the *linear* lower bounds proved by [7].

In this paper we establish the first *superpolynomial* hardness of the minimal spectrahedral representation for an *explicit family of hyperbolicity cones*, assuming Valiant's *VP vs VNP* conjecture is true, that is, that the permanent polynomial cannot be computed by algebraic circuits of polynomial size. More precisely, we prove that the hyperbolicity cone of Amini's *homogeneous matching polynomial* must require superpolynomial spectrahedral representations, assuming that Valiant's conjecture is true. This is the first work providing a (conditional) superpolynomial lower bound on the spectrahedral representation of an explicit hyperbolicity cone.

CCS CONCEPTS

• **Theory of computation** → **Algebraic complexity theory**; **Convex optimization**; *Semidefinite programming*; • **Mathematics of computing** → *Convex optimization.*

KEYWORDS

Algebraic Complexity, Hyperbolic Polynomials, Hyperbolicity Cones, Convex Optimization, Semidefinite Programming

ISSAC '20, July 20–23, 2020, Kalamata, Greece

ACM ISBN 978-1-4503-7100-1/20/07…$15.00
https://doi.org/10.1145/3373207.3404010

ACM Reference Format:
Rafael Oliveira. 2020. Conditional Lower Bounds on the Spectrahedral Representation of Explicit Hyperbolicity Cones. In *International Symposium on Symbolic and Algebraic Computation (ISSAC '20), July 20–23, 2020, Kalamata, Greece.* ACM, New York, NY, USA, 6 pages. https://doi.org/10.1145/3373207.3404010

1 INTRODUCTION

Let $\mathbf{x} = (x_1, \ldots, x_n)$ be a vector of variables $x_1, \ldots, x_n$ and $\mathbf{a} = (a_1, \ldots, a_n) \in \mathbb{R}^n$ be a vector of elements $a_1, \ldots, a_n$ from $\mathbb{R}$. A homogeneous polynomial $h(\mathbf{x}) \in \mathbb{R}[x_1, \ldots, x_n]$ is hyperbolic with respect to a direction $\mathbf{e} := (e_1, \ldots, e_n) \in \mathbb{R}^n$ if $h(\mathbf{e}) \neq 0$ and for all vectors $\mathbf{a} \in \mathbb{R}^n$, the univariate polynomial $f(t) := h(t\mathbf{e} - \mathbf{a})$ only has real zeros. By a result due to Gårding [3], each hyperbolic polynomial $h(\mathbf{x})$ defines a *hyperbolicity cone*, a closed convex cone denoted by $\Lambda_+(h, \mathbf{e})$ and defined as

$$\Lambda_+(h, \mathbf{e}) := \{\mathbf{a} \in \mathbb{R}^n \mid \text{ all roots of } h(t\mathbf{e} - \mathbf{a}) \text{ are non-negative}\}.$$

Gårding also showed [3] that $\Lambda_+(h, \mathbf{e})$ can be equivalently defined as the closure of the connected component of $\{\mathbf{a} \in \mathbb{R}^n \mid h(\mathbf{a}) \neq 0\}$ that contains $\mathbf{e}$.

Hyperbolic polynomials and hyperbolicity cones originated in the theory of PDE in the works of Petrovsky and Gårding, and are of importance in combinatorics and optimization. Hyperbolicity cones are important objects in optimization, as they generalize semidefinite cones and Güler [4] showed that one could generalize interior point methods of optimization to hyperbolicity cones. Since then the theory of hyperbolic programming has been vastly expanded, see [12] and references therein.

Despite much progress on the optimization side of hyperbolic programming, the geometric and complexity theoretic aspects of hyperbolicity cones are much less understood.

On the geometric side, an important open question is concerned with how general the class of hyperbolicity cones is. *Spectrahedral cones*, that is, linear sections of the cone of positive semidefinite matrices, form the most well-known examples of hyperbolicity cones. The generalized Lax conjecture states that every hyperbolicity cone is also a spectrahedral cone, whereas the projected Lax conjecture states that every hyperbolicity cone is a linear projection of a spectrahedral cone. Despite much recent work and some impressive progress on these conjectures [8, 10], they remain open.

The origins of these conjectures came from partial differential equations. When the number of variables of a hyperbolic polynomial is 3, say $h(x, y, z)$ is hyperbolic in direction (a, b, c), Lax conjectured [9] that any such hyperbolic polynomial could be written as a determinant of a linear combination of symmetric matrices of the form $xA + yB + zC$ such that $a \cdot A + b \cdot B + c \cdot C$ is positive

definite. This conjecture certainly implies that for 3 variables, every hyperbolicity cone is a spectrahedral cone. A positive answer to this conjecture was given by Helton and Vinnikov [5].

On the complexity theoretic side, very little is known about the complexity of representing hyperbolicity cones which are known to be spectrahedral. In the recent work [11], the authors prove exponential lower bounds even for approximate spectrahedral representations of *non-explicit* hyperbolicity cones which are spectrahedral. However, prior to the present work, no superpolynomial lower bound on the spectrahedral representation for an *explicit* hyperbolicity cone which is also spectrahedral was known. In the next section we present our main result and the overview of its proof, which is given formally in the next sections.

1.1 Main result and proof overview

In this paper, we prove a conditional lower bound on the minimal spectrahedral representation of the hyperbolicity cone of an *explicit family* of spectrahedral polynomials. More precisely, we prove the following theorem:

Theorem 1.1. *There exists an explicit family of hyperbolic polynomials* $\{h_n(\mathbf{x})\}_{n\geq 1}$ *and directions* $\{\mathbf{e}_n\}_{n\geq 1}$*, where* $h_n(\mathbf{x})$ *has* $\mathrm{poly}(n)$ *variables and* $\mathrm{poly}(n)$ *degree, whose hyperbolicity cone* $\Lambda_+(h_n, \mathbf{e}_n)$ *is spectrahedral and such that any spectrahedral representation of* $\Lambda_+(h_n, \mathbf{e}_n)$ *must have superpolynomial size in n, assuming that VP* $\neq$ *VNP.*

High-level ideas of the proof: The high level idea guiding the proof of Theorem 1.1 comes from the combination of the four facts below:

(1) Every spectrahedral cone has a corresponding definite determinantal representation. This follows by the definition of the spectrahedral cone.
(2) Irreducible hyperbolic polynomials are the minimal defining polynomials of their hyperbolicity cones. This fact follows from standard results in real algebraic geometry, and a proof is given in [5, Lemma 2.1].
(3) A necessary condition for the hyperbolicity cone of an irreducible hyperbolic polynomial $h(\mathbf{x})$ to be spectrahedral is the existence of a definite determinantal polynomial which is a multiple of $h(\mathbf{x})$. In Proposition 2.4 a necessary and sufficient condition is given.
(4) Factors of polynomials of small degree computed by small algebraic circuits also have small algebraic circuits, as was proved in the seminal work [6].

The facts above yield a useful necessary condition for a hyperbolicity cone to have a polynomial sized spectrahedral representation, and this necessary condition comes from algebraic complexity: the hyperbolic polynomial must be computed by polynomial sized circuits! This can be seen as follows: given a hyperbolicity cone, take its minimal defining polynomial $h(\mathbf{x})$. By [5, Lemma 2.1], any other polynomial $q(\mathbf{x})$ defining the same hyperbolicity cone must be a multiple of $h(\mathbf{x})$. If the hyperbolicity cone of $h(\mathbf{x})$ is spectrahedral, then there is a definite determinantal polynomial $D(\mathbf{x})$ defining the hyperbolicity cone of $h(\mathbf{x})$. If $D(\mathbf{x})$ can be defined by polynomial sized matrices, then the polynomial $D(\mathbf{x})$ can be computed by polynomial sized circuits. Thus, Kaltofen's seminal result (item 4) tells us that $h(\mathbf{x})$ can also be computed by polynomial sized circuits!

With the necessary condition above, the proof strategy is straightforward: simply construct an explicit irreducible hyperbolic polynomial $h(\mathbf{x})$ that requires superpolynomial sized algebraic circuits to compute it, and whose hyperbolicity cone is spectrahedral. Irreducibility of $h(\mathbf{x})$ implies that it is the minimal defining polynomial of its hyperbolicity cone, by item 2 above. Hardness of $h(\mathbf{x})$ and the necessary condition given by the previous paragraph, implies that any definite determinantal representation of the hyperbolicity cone of $h(\mathbf{x})$ must have superpolynomial size.

The only task left is to construct an irreducible hyperbolic polynomial which has a spectrahedral hyperbolicity cone and that is hard to compute by algebraic circuits. And it just so happens that Amini's homogeneous matching polynomial over the complete bipartite graph has all the properties above. Amini [1] shows that the homogeneous matching polynomial has a spectrahedral hyperbolicity cone. In Section 4 we show that this polynomial is irreducible for the complete bipartite graph.

Since we do not currently know superpolynomial lower bounds on the circuit complexity of any explicit polynomial, we will prove a conditional lower bound, which is based on Valiant's conjecture that VP $\neq$ VNP. Valiant's conjecture can be stated as: the Permanent polynomial cannot be computed by polynomial sized circuits. Thus, to prove that Amini's homogeneous matching polynomial is hard, we prove a reduction result: we show that if the matching polynomial of the complete bipartite graph can be computed by polynomial sized circuits, then there is a polynomial sized circuit computing the Permanent.

1.2 Related Work

Much work in the past decade has focused on proving generalizations of the Lax conjecture, whose aim is to relate hyperbolicity cones to spectrahedral cones. The *generalized Lax conjecture* states that every hyperbolicity cone is spectrahedral, while the *projected Lax conjecture* states that every hyperbolicity cone is the projection of a spectrahedral cone.

In [8], the author makes progress towards the generalized Lax conjecture, proving that every smooth hyperbolic polynomial is a factor of a definite determinantal polynomial, thus establishing one part of the equivalence from Proposition 2.4. In [10], the authors prove that smooth hyperbolicity cones are projections of spectrahedra, thus showing that the projected Lax conjecture is holds for almost all hyperbolicity cones. However, in these papers the computational complexity of their constructions is still left unexplored, and the current work is a step forward in understanding the computational complexity of these hyperbolicity cones.

On the lower bounds/impossibility side, [13] proves that many compact convex semialgebraic sets in euclidean space are not projections of spectrahedra. In [11], the authors prove exponential lower bounds on the spectrahedral representations of non-explicit spectrahedral hyperbolicity cones. Their lower bounds are unconditional, albeit being non-explicit.

1.3 Organization

In Section 2 we formally define hyperbolic polynomials and their hyperbolicity cones, spectrahedral cones and definite determinantal representations, establishing the basic facts about them, as well as the interconnections between these concepts. In Section 3 we establish the basic definitions and facts that we will need from Algebraic Complexity Theory, including the irreducibility and hardness of a variant of the Permanent polynomial. In Section 4 we prove the main result of the paper, which is the conditional lower bound on the spectrahedral representation of the hyperbolicity cone of the matching polynomial. In Section 5 we conclude and present some open problems.

2 HYPERBOLIC POLYNOMIALS AND SPECTRAHEDRALITY

In this section we formally define hyperbolic polynomials, definite determinantal representations, spectrahedral representations, and establish the known relationships between these three concepts.

2.1 Hyperbolic Polynomials and Definite Determinantal Representation

In this section we formally give the main definitions and background needed from hyperbolic polynomials and definite determinantal representations which will be used in the later sections.

Definition 2.1 (Hyperbolic Polynomial)**.** A homogeneous polynomial $h(\mathbf{x}) \in \mathbb{R}[\mathbf{x}]$ of degree d is hyperbolic with respect to direction $\mathbf{e} \in \mathbb{R}^n$ if $h(\mathbf{e}) \neq 0$ and for every $\mathbf{a} \in \mathbb{R}^n$, the univariate polynomial $h(t \cdot \mathbf{e} - \mathbf{a})$ is real rooted (counting their multiplicities). That is, $h(t \cdot \mathbf{e} - \mathbf{a})$ has exactly d real roots.

Definition 2.2 (Hyperbolicity Cone)**.** If $h(\mathbf{x}) \in \mathbb{R}[\mathbf{x}]$ is a hyperbolic polynomial with respect to direction $\mathbf{e}$, its *hyperbolicity cone* is the set defined by

$$\Lambda_+(h, \mathbf{e}) := \{\mathbf{a} \in \mathbb{R}^n \mid \text{ all roots of } h(t\mathbf{e} - \mathbf{a}) \text{ are non-negative } \}.$$

Definition 2.3 (Definite Determinantal Representation)**.** We say that a homogeneous polynomial $h(x) \in \mathbb{R}[\mathbf{x}]$ has a definite determinantal representation at $\mathbf{b} \in \mathbb{R}^n$ if there are $A_1, \ldots, A_n \in \mathsf{Sym}_d(\mathbb{R})$ and $\lambda \in \mathbb{R}^*$ such that:

(1) $\sum_{i=1}^{n} b_i \cdot A_i \succ 0$

(2) $h(\mathbf{x}) = \lambda \cdot \det\left(\sum_{i=1}^{n} x_i \cdot A_i\right)$.

Proposition 2.4 (Spectrahedral Representation Equivalent Formulation [17]). *Let $h \in \mathbb{R}[\mathbf{x}]$ be hyperbolic with respect to $\mathbf{e} \in \mathbb{R}^n$. The hyperbolicity cone $\Lambda_+(h, \mathbf{e})$ is spectrahedral if, and only if, there is a hyperbolic polynomial $q \in \mathbb{R}[\mathbf{x}]$ with respect to $\mathbf{e}$ such that the following two conditions are satisfied:*

(1) $q \cdot h$ has a definite determinantal representation at $\mathbf{e}$

(2) $\Lambda_+(h, \mathbf{e}) \subseteq \Lambda_+(q, \mathbf{e})$.

The following follows from [5, Lemma 2.1]. It essentially states that the hyperbolicity cone $\Lambda_+(h, \mathbf{e})$ of an irreducible hyperbolic polynomial h has the polynomial h as its *minimal defining polynomial*. That is, any other polynomial g also defining $\Lambda_+(h, \mathbf{e})$ must be a multiple of h.

Proposition 2.5 (Hyperbolic Cones of Irreducible Polynomials). *If $h \in \mathbb{R}[\mathbf{x}]$ is an irreducible and hyperbolic polynomial with respect to $\mathbf{e} \in \mathbb{R}^n$, and $q \in \mathbb{R}[\mathbf{x}]$ is a hyperbolic polynomial such that $\Lambda_+(h, \mathbf{e}) = \Lambda_+(q, \mathbf{e})$, then h divides q.*

If a hyperbolicity cone $\Lambda_+(h, \mathbf{e})$ is spectrahedral, i.e. a linear section of the positive semidefinite cone, let

$$\Lambda_+(h, \mathbf{e}) = \left\{\mathbf{a} \in \mathbb{R}^n \mid \sum_{i=1} a_i \cdot A_i \succeq 0\right\}$$

be any spectrahedral representation of the hyperbolicity cone, where $A_i \in Sym_D(\mathbb{R})$ are real symmetric matrices of dimension D.

In this case, we have that $P(\mathbf{x}) = \mathrm{Det}(\sum_{i=1}^{n} A_i \cdot x_i)$ is a hyperbolic poylnomial at $\mathbf{e}$ such that $\Lambda_+(P, \mathbf{e}) = \Lambda_+(h, \mathbf{e})$. Thus, if $h(\mathbf{x})$ is an irreducible polynomial, by Proposition 2.5, we must have that $h(\mathbf{x})$ divides $P(\mathbf{x})$. We will need this fact in the proof of our main result in Section 4.

2.2 Homogeneous Multivariate Matching Polynomial

In this section we describe our candidate hard polynomial, which was first defined in [1, Definition 2.1] as a multivariate generalization of the univariate matching polynomial from algebraic combinatorics, and as a variant on the multivariate matching polynomial of Heilmann and Lieb.

Definition 2.6 (Homogeneous Multivariate Matching Polynomial [1])**.** Let $G(V, E)$ be an undirected graph, $\mathbf{x} = (x_v)_{v \in V}$ and $\mathbf{w} = (w_e)_{e \in E}$ be indeterminates. The *homogeneous multivariate matching polynomial* is defined by

$$\mu_G(\mathbf{x}, \mathbf{w}) = \sum_{M \in \mathcal{M}(G)} (-1)^{|M|} \cdot \prod_{v \notin V(M)} x_v \cdot \prod_{e \in M} w_e^2, \qquad (1)$$

where in the equation above $\mathcal{M}(G)$ is the set of all matchings of G (including the empty set), M is a matching of G (the collection of edges forming the matching), $V(M)$ is the set of vertices participating in the matching M and $|M|$ is the number of edges in the matching.

Remark 2.7. *Note that if a graph G has perfect matchings, they are captured by $\mu_G(\mathbf{x}, \mathbf{w})$ by setting $\mathbf{x} = \mathbf{0}$. That is,*

$$\mu_G(\mathbf{0}, \mathbf{w}) = \sum_{M \text{ is perfect matching}} (-1)^{|M|} \cdot \prod_{e \in M} w_e^2$$

Throughout this section, we let $\mathbf{e} := (\mathbf{1}_V, \mathbf{0}_E)$ be the direction given by the all one's vector in the variables $(x_v)_{v \in V}$ and the zero vector in the variables $(w_e)_{e \in E}$. In [1, Theorem 2.12], Amini shows that the hyperbolicity cone $\Lambda_+(\mu_G, \mathbf{e})$ is spectrahedral.

Proposition 2.8 (Spectrahedrality of Matching Polynomial [1]). *The hyperbolicity cone $\Lambda_+(\mu_G(\mathbf{x}, \mathbf{w}), \mathbf{e})$ is spectrahedral.*

From the fact above, together with Proposition 2.4, we obtain the following corollary.

Corollary 2.9. *There exists a hyperbolic polynomial $q \in \mathbb{R}[\mathbf{x}, \mathbf{w}]$ w.r.t. direction $\mathbf{e}$ such that the polynomial $q \cdot \mu_G(\mathbf{x}, \mathbf{w})$ has a definite determinantal representation and $\Lambda_+(\mu_G, \mathbf{e}) \subseteq \Lambda_+(q, \mathbf{e})$.*

3 ALGEBRAIC COMPLEXITY

In this section, we define the basic notions of algebraic complexity and establish the basic facts which we will need for the proof of our main theorem in the next section. We start with the definition of an algebraic circuit, which can be found in [14].

Definition 3.1 (Algebraic Circuits)**.** An algebraic circuit Φ over a field $\mathbb{F}$ and a set of variables $\mathbf{x} = (x_1, \ldots, x_n)$ is a directed acyclic graph defined as follows. The vertices of Φ are the gates of the circuit, and each gate of indegree 0 is labeled by either a variable from $\mathbf{x}$ or by a field element from $\mathbb{F}$. Every other gate in Φ is labeled by either $+, \times$ and has indegree 2.

From the definition above, one can see that an algebraic circuit computes polynomials in a natural way. Each input gate is either a variable or a field element, and a + gate computes the polynomial given by the sum of its input gates, and a $\times$ gate computes the product of its input gates. We say that a circuit Φ computes a polynomial p if there is a gate of Φ which computes the polynomial p.

The size of an algebraic circuit is defined as the number of gates in the circuit. The formal degree of a circuit Φ is defined inductively as follows: an input gate of Φ has degree 1 if it is a variable, and 0 otherwise. For any + gate $u = v + w$ of the circuit, we make $\deg(u) = \max\{\deg(v), \deg(w)\}$ and for a $\times$ gate $u = v \times w$ we make $\deg(u) = \deg(v) + \deg(w)$. We define the degree of Φ as the maximum degree among the degrees of the gates of Φ.

We say that a circuit Φ is a homogeneous circuit if each gate of Φ computes a homogeneous polynomial. Note that in a homogeneous circuit Φ computing a (homogeneous) polynomial p of degree d only the gates of degree $\leq d$ are needed from Φ. Hence, if we are interested in the computation of p alone, we can assume that Φ has degree d as well.

Given a polynomial $p(\mathbf{x})$, denote its homogeneous component of degree r by $H_r[p(\mathbf{x})]$. The following proposition due to [15] tells us that given an algebraic circuit of polynomial size, we can efficiently compute its low degree components with algebraic circuits. A proof can be found in [14, Theorem 2.2].

Proposition 3.2 (Complexity of Computing Homogeneous Components [15]). *If $p(\mathbf{x}) \in \mathbb{R}[\mathbf{x}]$ can be computed by an algebraic circuit $\Phi(\mathbf{x})$ of size s, then for every $r \in \mathbb{N}$, there is a homogeneous circuit $\Psi(\mathbf{x})$ of size at most $O(r^2 s)$ computing $H_0[p(\mathbf{x})], \ldots, H_r[p(\mathbf{x})]$.*

Remark 3.3. *Note that in the proposition above, there is no requirement on the degree of the circuit Φ, while the homogeneous circuit Ψ will have degree bounded by r.*

One of the main goals of algebraic complexity theory is to classify which families of polynomials $\{p_n\}_{n\geq 1}$ where $p_n \in \mathbb{F}[x_1, \ldots, x_n]$ can be computed by a family of algebraic circuits $\{\Phi_n\}_{n\geq 1}$ of polynomial size. The theory has mostly been concerned with families of polynomials $\{p_n\}_{n\geq 1}$ with $\deg(p_n)$ being a polynomial function of n.

For such families of polynomials having polynomial degree in the number of variables, the class of families of polynomials which can be computed by a family of algebraic circuits of polynomial size is denoted by VP. This is the class of "efficiently computable" polynomials.

One of the most important family of polynomials which is in VP is the family defined by the determinant polynomial: given an $n \times n$ symbolic matrix X,

$$\mathrm{Det}_n(X) = \sum_{\sigma \in S_n} (-1)^{\sigma} \prod_{i=1}^{n} X_{i\sigma(i)}.$$

Another important class of families of polynomials is the class denoted by VNP, which is the algebraic analogue of the class NP, and informally speaking is the class of families of polynomials which can be "defined efficiently." For a more precise definition see [14, Definition 1.3].

There is a beautiful theory of completeness and reductions for these algebraic classes, analogue to the theory developed in the boolean setting for P and NP, whose origins trace back to the seminal work of Valiant [16]. One of the major open problems in algebraic complexity theory, posed by Valiant, is whether the classes VP and VNP are different or not.

One complete family of polynomials in VNP is defined by the permanent polynomial: given an $n \times n$ symbolic matrix X,

$$\mathrm{Per}_n(X) = \sum_{\sigma \in S_n} \prod_{i=1}^{n} X_{i\sigma(i)},$$

and therefore the VP versus VNP question can be stated as:

Conjecture 3.4 (Valiant's VP $\neq$ VNP Conjecture). *The family defined by the permanent polynomials $\{\mathrm{Per}_n(X)\}n \geq 1$ cannot be computed by circuits in VP.*

For the sake of conciseness, we shall from now on refer to a family of polynomials simply by one of its elements. For instance, when talking about the family defined by the permanent polynomials of degree n, we shall simply talk about the polynomial $\mathrm{Per}_n(X)$. The parameter defining the family of polynomials is n. Thus, we will refer to the polynomial $\mathrm{Per}_n(X)$ and the family $\{\mathrm{Per}_n(X)\}_{n\geq 1}$ interchangeably.

The class VP enjoys many closure properties under fundamental algebraic operations. One of its most remarkable was proved in the seminal work of Kaltofen [6] and states that the class VP is closed under factorization.

Proposition 3.5 (Factors are Closed in VP [6]). *If a polynomial $p(\mathbf{x}) \in \mathbb{R}[\mathbf{x}]$ of degree d can be computed by an algebraic circuit of size s, then any factor $g(\mathbf{x})$ of the polynomial $p(\mathbf{x})$ can be computed by an algebraic circuit of size $\mathrm{poly}(n, s, d)$.*

We now proceed to establishing two lemmas that shall be important for us in the subsequent sections. From now on, we will be working over the base field $\mathbb{R}$. The first lemma establishes the VNP-hardness of a particular polynomial: the squared permanent polynomial, which is defined below.

Lemma 3.6 (Complexity of the squared Permanent). *Let $W = (w_{i,j}^2)_{i,j=1}^n$ be a symbolic matrix over the variables $\mathbf{w} = (w_{i,j})_{i,j=1}^n$. If VP $\neq$ VNP then any algebraic circuit computing $\mathrm{Per}_n(W)$ must have superpolynomial size.*

Proof. Assume, for the sake of contradiction, that there is a circuit $\Phi(\mathbf{w})$ of size $O(n^c)$ computing $\mathrm{Per}_n(W)$, where $c \in \mathbb{Z}$ is a positive constant.

Let $u_{i,j} = (1 - x_{i,j})^{1/2}$. Then, $\Phi(\mathbf{u}) = \mathrm{Per}_n(J - X)$, where J is the all-ones matrix and $X = (x_{i,j})$ is a pure symbolic matrix.

Each $u_{i,j}$ is a univariate real analytic function on the variable $x_{i,j}$ over the ball of radius $1/2$ around the origin. Take the Taylor expansion of $u_{i,j}$ around $x_{i,j} = 0$. Call this Taylor series $v_{i,j}$. The truncated Taylor series $v_{i,j}$, truncated at degree n, can be computed by an algebraic circuit of size $O(n)$, as it is an univariate polynomial of degree n. Let $T_{i,j}$ be the truncation of $v_{i,j}$ at degree n.

Letting $T = (T_{i,j})_{i,j=1}^n$, we have that

$$(-1)^n \cdot \mathrm{Per}_n(X) = H_n[\mathrm{Per}_n(J - X)] = H_n[\Phi(T)].$$

Note that $\Phi(T)$ is a circuit of size[1] $O(n^{c+3})$, as we replaced each variable $w_{i,j}$ in the cicuit $\Phi(\mathbf{w})$ by the truncated Taylor expansion $T_{i,j}$ of $u_{i,j}$, and we saw that each $T_{i,j}$ can be computed by a circuit of size $O(n)$. As there are n^2 such Taylor expansions, the size is $O(n^{c+3})$.

By applying Proposition 3.2, the homogeneous part of degree n of $\Phi(T)$ can be computed by a homogeneous circuit of size $O(n^2 \cdot n^{c+3}) = O(n^{c+5})$ and degree n. This implies that $\mathrm{Per}(X) \in \mathsf{VP}$, which would imply that $\mathsf{VP} = \mathsf{VNP}$. □

We will also need to establish that the squared permanent is an irreducible polynomial. This will be important in our proof that the homogeneous matching polynomial of the complete bipartite graph is irreducible.

Lemma 3.7 (Irreducibility of the squared Permanent). *Let $n \geq 2$ and $W = (w_{i,j}^2)_{i,j=1}^n$ be a symbolic matrix over the variables $\mathbf{w} = (w_{i,j})_{i,j=1}^n$. Then the polynomial $\mathrm{Per}(W)$ is irreducible over $\mathbb{R}[\mathbf{w}]$.*

Proof. Suppose $\mathrm{Per}(W) = p(\mathbf{w}) \cdot q(\mathbf{w})$. Assume there is some entry $(i, j) \in [n]^2$ such that $p(\mathbf{w})$ is linear w.r.t. $w_{i,j}$. In this case, $q(\mathbf{w})$ is also linear w.r.t. $w_{i,j}$ and we would have $p(\mathbf{w}) = a_p \cdot w_{i,j} + b_p$ and $q(\mathbf{w}) = a_q \cdot w_{i,j} + b_q$, where $a_p, b_p, a_q, b_q \in \mathbb{R}[\mathbf{w}]$ are nonzero polynomials which do not depend on $w_{i,j}$. In this case, we have that $a_p \cdot a_q$ computes the permanent of the (i, j)-minor of W (and thus is a sum of squares polynomial) and we have that $b_p \cdot b_q$ computes another sum of squares polynomial (due to the cofactor expansion of the Permanent). This implies that $a_p \cdot a_q > 0$ for all its non-zero values, and so is $b_p \cdot b_q > 0$.

However, as $\mathrm{Per}(W) = p(\mathbf{w}) \cdot q(\mathbf{w})$, the linear term in $w_{i,j}$ in the multiplication $p(\mathbf{w}) \cdot q(\mathbf{w})$ must vanish, thus implying $a_p \cdot b_q + a_q \cdot b_p = 0$, which implies that $a_p \cdot b_q \cdot a_q \cdot b_p < 0$ for any non-zero evaluation of these polynomials, contradicting the previous paragraph.

Thus, we are left with the case where for each $w_{i,j}$, we have that either $p(\mathbf{w}) = w_{i,j}^2 \cdot a_p + b_p$ and $q(\mathbf{w}) = b_q$, where $a_p, b_p, b_q \in \mathbb{R}[\mathbf{w}]$ do not depend on $w_{i,j}$, or the other way around (q is the purely quadratic polynomial in $w_{i,j}$ whereas p is constant in $w_{i,j}$). In this case, since no linear terms on any $w_{i,j}$ appear in the factorization $\mathrm{Per}(W) = p(\mathbf{w}) \cdot q(\mathbf{w})$, this factorization after doing a change of variables $x_{i,j} = w_{i,j}^2$ yields a polynomial factorization of the usual permanent, which is known to be irreducible for $n \geq 2$. □

[1] The more precise bound is $O(n^{\max(c,3)})$, since the size of a composition of circuits is simply the sum of the sizes of the circuits being used.

4 COMPLEXITY OF DEFINITE DETERMINANTAL REPRESENTATIONS

In this section we prove the main result of this paper: the conditional complexity lower bound on the spectrahedral representation of the matching polynomial for the complete bipartite graph $K_{n,n}$.

For this section, we will let $\mu(\mathbf{x}, \mathbf{w}) \triangleq \mu_{K_{n,n}}(\mathbf{x}, \mathbf{w})$ and $\mathbf{e} = (\mathbf{1}_n, \mathbf{1}_n, \mathbf{0}_{E(K_{n,n})})$ be the hyperbolicity direction for $\mu(\mathbf{x}, \mathbf{w})$ from Amini's theorem.

Lemma 4.1 (Complexity of Complete Bipartite Matching Polynomial). *Assuming VP ≠ VNP, that is, that the permanent polynomial has super-polynomial circuit size, then the polynomial $\mu(\mathbf{x}, \mathbf{w})$ requires super polynomial size circuits.*

Proof. Let $W = (w_{ij}^2)_{i,j=1}^n$ be a symbolic matrix. Note that $\mu(\mathbf{0}, \mathbf{w}) = \mathrm{Per}_n(W)$. By Lemma 3.6 and our assumption that VP ≠ VNP, we have that $\mu(\mathbf{0}, \mathbf{w})$ requires superpolynomial-sized circuits to compute it.

If $\Phi(\mathbf{x}, \mathbf{w})$ is any algebraic circuit computing $\mu(\mathbf{x}, \mathbf{w})$ with size s (i.e., having s gates, one of them computing the polynomial $\mu(\mathbf{x}, \mathbf{w})$), the circuit $\Phi(\mathbf{0}, \mathbf{w})$, obtained by setting the input variables $\mathbf{x}$ to $\mathbf{0}$, also has size $\leq s$ and computes the polynomial $\mu(\mathbf{0}, \mathbf{w})$. As $\Phi(\mathbf{0}, \mathbf{w})$ requires superpolynomial size, by the previous paragraph, we also have that $\Phi(\mathbf{x}, \mathbf{w})$ requires superpolynomial size. □

Lemma 4.2 (Irreducibility of Complete Bipartite Matching Polynomial). *The polynomial $\mu(\mathbf{x}, \mathbf{w})$ is irreducible over $\mathbb{R}[\mathbf{x}, \mathbf{w}]$.*

Proof. Suppose, for the sake of contradiction, that $\mu(\mathbf{x}, \mathbf{w})$ factors. Then, there exist polynomials $p(\mathbf{x}, \mathbf{w})$ and $q(\mathbf{x}, \mathbf{w})$ such that $\mu(\mathbf{x}, \mathbf{w}) = p(\mathbf{x}, \mathbf{w}) \cdot q(\mathbf{x}, \mathbf{w})$. Consider the polynomials above in the ring $(\mathbb{R}[\mathbf{w}])[\mathbf{x}]$. As the constant coefficient of $\mu(\mathbf{x}, \mathbf{w})$ is $\mu(\mathbf{0}, \mathbf{w}) = (-1)^n \cdot \mathrm{Per}_n(W)$, which is nonzero, we must have that $p(\mathbf{0}, \mathbf{w})$ and $q(\mathbf{0}, \mathbf{w})$ are nonzero. However, by Lemma 3.7, we have that $\mathrm{Per}_n(W)$ is irreducible, which implies w.l.o.g. that $p(\mathbf{0}, \mathbf{w}) = (-1)^n \cdot \mathrm{Per}_n(W)$ and $q(\mathbf{0}, \mathbf{w}) = 1$.

Since $\mu(\mathbf{x}, \mathbf{0}) = \prod_{1 \leq i \leq 2n} x_i$ is nonzero, we must have $p(\mathbf{x}, \mathbf{0})$ and $q(\mathbf{x}, \mathbf{0})$ are nonzero. If we look at $\mu(\mathbf{x}, \mathbf{0}) = p(\mathbf{x}, \mathbf{0}) \cdot q(\mathbf{x}, \mathbf{0})$, we have that $q(\mathbf{x}, \mathbf{0})$ must either be constant or a monomial over $\mathbf{x}$. As the previous paragraph implies $q(\mathbf{0}, \mathbf{0}) = 1$, $q(\mathbf{x}, \mathbf{0})$ cannot be a non-constant monomial over $\mathbf{x}$, as that would imply $q(\mathbf{0}, \mathbf{0}) = 0$. Hence, we have that $p(\mathbf{x}, \mathbf{0}) = \prod_{1 \leq i \leq 2n} x_i$.

If $q(\mathbf{x}, \mathbf{w})$ is a non-constant polynomial, any of its non-constant monomials must depend on both $\mathbf{x}$ and $\mathbf{w}$ variables, as $q(\mathbf{0}, \mathbf{w}) = q(\mathbf{x}, \mathbf{0}) = 1$. If $q(\mathbf{x}, \mathbf{w})$ depends on some $\mathbf{x}$ variable, say x_1 w.l.o.g., write $q(\mathbf{x}, \mathbf{w}) = q_1(\mathbf{x}, \mathbf{w})x_1 + q_0(\mathbf{x}, \mathbf{y})$, where q_0 does not depend on x_1. As $\mu(\mathbf{x}, \mathbf{w})$ is linear in x_1, we must have that q is linear in x_1 and p does not depend on x_1. However, this contradicts the fact that $p(\mathbf{x}, \mathbf{0}) = \prod_{1 \leq i \leq 2n} x_i$. Hence, we conclude that $q(\mathbf{x}, \mathbf{w})$ does not depend on any $\mathbf{x}$ variable, which implies $q(\mathbf{x}, \mathbf{w}) = q(\mathbf{0}, \mathbf{w}) = 1$, which proves that $\mu(\mathbf{x}, \mathbf{w})$ is irreducible. □

Putting the pieces together, we can now prove our main result: assuming that VP ≠ VNP, any spectrahedral representation of the hyperbolicity cone of the complete bipartite matching polynomial has superpolynomial size.

Theorem 4.3 (Hardness of Spectrahedral Representation). *Assuming that VP ≠ VNP, the following is true: any spectrahedral representation of the spectrahedral cone $\Lambda_+(\mu, \mathbf{e})$ of the matching polynomial $\mu_{K_{n,n}}(\mathbf{x}, \mathbf{w})$ has superpolynomial dimension.*

Proof. Let $(A_i)_{i\in[n]} \cup (B_j)_{j\in[n]} \cup (C_{(i,j)})_{(i,j)\in[n]^2}$ be a spectrahedral representation of the hyperbolicity cone $\Lambda_+(\mu, \mathbf{e})$ of the polynomial $\mu(\mathbf{x}, \mathbf{w})$, where $A_i, B_j, C_{(i,j)} \in Sym_d(\mathbb{R})$ are real symmetric matrices of dimension d such that $\sum_{i\in[n]} A_i + \sum_{j\in[n]} B_j \succ 0$. Let

$$g(\mathbf{x}, \mathbf{w}) = \mathrm{Det}\left(\sum_{i,j=1}^{n} A_i x_i + B_j x_{n+j} + C_{(i,j)} w_{(i,j)}\right)$$

The irreducibility of $\mu(\mathbf{x}, \mathbf{w})$ proved in Lemma 4.2, together with Proposition 2.5 tell us that $\mu(\mathbf{x}, \mathbf{w})$ divides $g(\mathbf{x}, \mathbf{w})$. If $d = \mathrm{poly}(n)$, the equality above gives an arithmetic circuit of size $\mathrm{poly}(d)$ computing $g(\mathbf{x}, \mathbf{w})$. In this case Proposition 3.5 and $\mu(\mathbf{x}, \mathbf{w}) \mid g(\mathbf{x}, \mathbf{w})$ imply that $\mu(\mathbf{x}, \mathbf{w})$ is computed by algebraic circuits of polynomial size, which contradicts Lemma 4.1. □

5 CONCLUSION AND OPEN PROBLEMS

In this paper we gave the first (conditional) lower bound on the spectrahedral representation of an *explicit* hyperbolicity cone which is known to be spectrahedral. An important component of our proof was to observe that the algebraic circuit complexity of the minimal defining polynomial of this hyperbolicity cone plays an important role in lower bounding the spectrahedral representation. Removing the standard complexity assumption on the proof above is the first open problem left by this work. It would be interesting to see whether the hyperbolicity assumption, and the special nature of the spectrahedral (or definite determinantal) representation could be further used to improve the lower bound above.

Another interesting question, in the viewpoint of optimization, is whether the complexity of representing a hyperbolicity cone (the ones known to be spectrahedral) via its hyperbolic polynomial can in general be much more efficient than representing it via its spectrahedral representation. This could show that using hyperbolic polynomials could provide faster ways of testing membership in in the hyperbolicity cone, than via checking the corresponding inequality given by the spectrahedral representation.

To achieve such a separation between representation by giving a circuit for the hyperbolic polynomial, one would have to find a hyperbolic polynomial (with a spectrahedral hyperbolicity cone) which can be computed by small algebraic circuits, but any definite determinantal representation of it is large. The elementary symmetric polynomials are great candidates for such separation, as they can be computed by algebraic circuits of $O(n^3)$ size. On the other hand, the best upper bound on the spectrahedral representation of the hyperbolicity cones of the elementary symmetric polynomials is exponential [1, 2]. Thus, another open question is to obtain a lower bound on the spectrahedral representation of these hyperbolicity cones.

For optimization, the best possible separation which could show the advantages of hyperbolic programming is with respect to spectrahedral shadows. In this case, one would have to exhibit a hyperbolicity cone which can be efficiently described through a small algebraic circuit computing its minimal defining polynomial, but for which any spectrahedral shadow of this cone is of superpolynomial size.

ACKNOWLEDGMENTS

The author is grateful to Nikhil Srivastava for posing the question at the Simons program and for useful conversations throughout this work. The author is also grateful to Mario Kummer and Rainer Sinn for useful conversations throughout the course of this work. This work started while the author was a research fellow at the Simons Institute, Berkeley and part of this work was done while the author was a postdoctoral fellow at the Department of Computer Science, University of Toronto.

REFERENCES

[1] Nima Amini. 2019. Spectrahedrality of hyperbolicity cones of multivariate matching polynomials. *Journal of Algebraic Combinatorics* 50, 2 (2019), 165–190.
[2] Petter Brändén. 2014. Hyperbolicity cones of elementary symmetric polynomials are spectrahedral. *Optimization Letters* 8, 5 (2014), 1773–1782.
[3] Lars Gårding. 1959. An inequality for hyperbolic polynomials. *Journal of Mathematics and Mechanics* (1959), 957–965.
[4] Osman Güler. 1997. Hyperbolic polynomials and interior point methods for convex programming. *Mathematics of Operations Research* 22, 2 (1997), 350–377.
[5] J William Helton and Victor Vinnikov. 2007. Linear matrix inequality representation of sets. *Communications on Pure and Applied Mathematics: A Journal Issued by the Courant Institute of Mathematical Sciences* 60, 5 (2007), 654–674.
[6] Erich Kaltofen. 1989. Factorization of polynomials given by straight-line programs. *Randomness and Computation* 5, 375-412 (1989), 2–3.
[7] Mario Kummer. 2016. Two results on the size of spectrahedral descriptions. *SIAM Journal on Optimization* 26, 1 (2016), 589–601.
[8] Mario Kummer. 2017. Determinantal representations and Bézoutians. *Mathematische Zeitschrift* 285, 1-2 (2017), 445–459.
[9] Peter D Lax. 1958. Differential equations, difference equations and matrix theory. *Communications on Pure and Applied Mathematics* 11, 2 (1958), 175–194.
[10] Tim Netzer and Raman Sanyal. 2015. Smooth hyperbolicity cones are spectrahedral shadows. *Mathematical Programming* 153, 1 (2015), 213–221.
[11] Prasad Raghavendra, Nick Ryder, Nikhil Srivastava, and Benjamin Weitz. 2019. Exponential lower bounds on spectrahedral representations of hyperbolicity cones. In *Proceedings of the Thirtieth Annual ACM-SIAM Symposium on Discrete Algorithms*. SIAM, 2322–2332.
[12] James Renegar. 2004. *Hyperbolic programs, and their derivative relaxations.* Technical Report. Cornell University Operations Research and Industrial Engineering.
[13] Claus Scheiderer. 2018. Spectrahedral shadows. *SIAM Journal on Applied Algebra and Geometry* 2, 1 (2018), 26–44.
[14] Amir Shpilka and Amir Yehudayoff. 2010. Arithmetic circuits: a survey of recent results and open questions. *Foundations and Trends® in Theoretical Computer Science* 5, 3–4 (2010), 207–388.
[15] Volker Strassen. 1973. Vermeidung von Divisionen. *Journal für die reine und angewandte Mathematik* 264 (1973), 184–202.
[16] Leslie G Valiant. 1979. Completeness classes in algebra. In *Proceedings of the eleventh annual ACM symposium on Theory of computing*. 249–261.
[17] Victor Vinnikov. 2012. LMI representations of convex semialgebraic sets and determinantal representations of algebraic hypersurfaces: past, present, and future. In *Mathematical methods in systems, optimization, and control*. Springer, 325–349.

Ideal Interpolation, H-Bases and Symmetry

Erick Rodriguez Bazan
Université Côte d'Azur, France
Inria Méditerranée, France
erick-david.rodriguez-bazan@inria.fr

Evelyne Hubert
Université Côte d'Azur, France
Inria Méditerranée, France
evelyne.hubert@inria.fr

ABSTRACT

Multivariate Lagrange and Hermite interpolation are examples of ideal interpolation. More generally an ideal interpolation problem is defined by a set of linear forms, on the polynomial ring, whose kernels intersect into an ideal.

For an ideal interpolation problem with symmetry, we address the simultaneous computation of a symmetry adapted basis of the least interpolation space and the symmetry adapted H-basis of the ideal. Beside its manifest presence in the output, symmetry is exploited computationally at all stages of the algorithm.

CCS CONCEPTS

• **Computing methodologies → Symbolic and algebraic algorithms**;

KEYWORDS

Interpolation; Symmetry; Representation Theory; Group Action; H-basis; Macaulay matrix; Vandermonde matrix

ACM Reference Format:
Erick Rodriguez Bazan and Evelyne Hubert. 2020. Ideal Interpolation, H-Bases and Symmetry. In *International Symposium on Symbolic and Algebraic Computation (ISSAC '20), July 20–23, 2020, Kalamata, Greece.* ACM, New York, NY, USA, 8 pages. https://doi.org/10.1145/3373207.3404057

1 INTRODUCTION

Preserving and exploiting symmetry in algebraic computations is a challenge that has been addressed within a few topics and, mostly, for specific groups of symmetry; For instance interpolation and symmetric group [23], cubature [4, 14], global optimisation [17, 32], equivariant dynamical systems [15, 20] and solving systems of polynomial equations [12, 13, 16, 19, 21, 31, 38]. In [33] we addressed multivariate interpolation and in this article we go further with ideal interpolation. We provide an algorithm to compute simultaneously a symmetry adapted basis of the least interpolation space and a symmetry adapted H-basis of the associated ideal. In addition to being manifest in the output, symmetry is exploited all along the algorithm to reduce the size of the matrices involved, and avoid sizable redundancies. Based on QR-decomposition (as opposed to LU-decomposition previously) the algorithm also lends itself to numerical computations.

Multivariate Lagrange, and Hermite, interpolation are examples of the encompassing notion of ideal interpolation, introduced in [2]. They are defined by linear forms consisting of evaluation at some nodes, and possibly composed with differential operators, without *gaps*. More generally a space of linear forms Λ on the polynomial ring $\mathbb{K}[\mathrm{x}] = \mathbb{K}[x_1, \ldots, x_n]$ is an ideal interpolation scheme if

$$\mathcal{I} = \bigcap_{\lambda \in \Lambda} \ker \lambda = \{p \in \mathbb{K}[\mathrm{x}] : \lambda(p) = 0, \text{ for all } \lambda \text{ in } \Lambda\} \qquad (1)$$

is an ideal in $\mathbb{K}[\mathrm{x}]$. In the case of Lagrange interpolation, $\mathcal{I}$ is the ideal of the nodes and is thus a radical ideal.

If Λ is invariant under the action of a group G, then so is $\mathcal{I}$. In [33] we addressed the computation of an interpolation space for Λ *i.e.,* a subspace of the polynomial ring that has a unique interpolant for each instantiated interpolation problem, that is both invariant and of minimal degree. An interpolation space for Λ identifies with the quotient space $\mathbb{K}[\mathrm{x}]/\mathcal{I}$. Hence a number of operations related to $\mathcal{I}$ can already be performed with a basis of an interpolation space for Λ: decide of membership to $\mathcal{I}$, determine normal forms of polynomials modulo $\mathcal{I}$ and compute matrices of multiplication maps in $\mathbb{K}[\mathrm{x}]/\mathcal{I}$. Yet it has also proved relevant to compute Gröbner bases or H-bases of $\mathcal{I}$.

Initiated in [26], for a set Λ of point evaluations, computing a Gröbner basis of $\mathcal{I}$ found applications in the design of experiments [29, 30]. As pointed out in [25], one can furthermore interpret the FGLM algorithm [10] as an instance of this problem. The linear forms are the coefficients, in the normal forms, of the reduced monomials. The alternative approach in [11] can be understood similarly.

The resulting algorithm then pertains to the Berlekamp-Massey-Sakata algorithm and is related the multivariate version of Prony's problem to compute Gröbner bases, border bases, or H-bases [1, 28, 35, 36]

All ,the above mentioned algorithms and complexity analyses heavily depend on a term order and basis of monomials. These are notoriously not suited for preserving symmetry. Our ambition in this paper is to showcase how symmetry can be embedded in the representation of both the interpolation space and the representation of the ideal. This is a marker for the more canonical representations.

The *least interpolation space*, defined in [6], and revisited in [33] is a canonically defined interpolation space. It serves here as the canonical representation of the quotient of the polynomial algebra by the ideal. It has great properties, even beyond symmetry, that cannot be achieved by a space spanned by monomials. In [33] we freed the computation of the least interpolation space from its reliance on the monomial basis by introducing *dual bases*. We pursue this approach here for the representation of the ideal by H-bases [24, 27]. Where Gröbner bases single out leading terms with a term order, H-bases work with leading forms and the orthogonality with respect to the apolar product. The least interpolation space

ISSAC '20, July 20–23, 2020, Kalamata, Greece

ACM ISBN 978-1-4503-7100-1/20/07…$15.00
https://doi.org/10.1145/3373207.3404057

then reveals itself as the orthogonal complement of the ideal of leading forms.

As a result, computing a H-basis of the interpolation ideal is achieved with linear algebra in subspaces of homogeneous polynomials of growing degrees. Yet we shall first redefine the concepts at play in an intrinsic manner, contrary to the computation centered approach in [27, 34]. The precise algorithm we shall offer to compute H-bases somehow fits in the loose sketch proposed in [5]. Yet we are now in a position to incorporate symmetry in a natural way, refining the algorithm to exploit it; A totally original contribution.

Symmetry is preserved and exploited thanks to the block diagonal structure of the matrices at play in the algorithms. This block diagonalisation, with predicted repetitions in the blocks, happens when the underlying maps are discovered to be equivariant and expressed in the related *symmetry adapted bases*. The case of the Vandermonde matrix was settled in [33]. In this paper, we also need the matrix of the prolongation map, known in the monomial basis as the Macaulay matrix. Figuring out the equivariance of this map is one of the original key results of this paper.

The paper is organized as follows. In Section 2 we define ideal interpolation and explain the identification of an interpolation space with the quotient algebra. In Section 3 we review H-bases and discuss how they can be computed in the ideal interpolation setting. In Section 4 we provide an algorithm to compute simultaneously a basis of the least interpolation space and an orthogonal H-basis of the ideal. In Section 5 we show how the Macaulay matrix can be block diagonalized in the presence of symmetry. This is then applied in Section 6 to obtain an algorithm to compute simultaneously a symmetry adapted basis of the least interpolation space and a symmetry adapted H-basis of the ideal. All along the paper, the definitions and notations comply with those in [33].

2 IDEAL INTERPOLATION

In this section, we consider the ideal interpolation problem and explain the identification of an interpolation space with the quotient algebra. We recall that the least interpolation space is the orthogonal complement of the ideal of the leading forms, $\mathcal{I}^0$.

$\mathbb{K}$ denotes either $\mathbb{C}$ or $\mathbb{R}$. $\mathbb{K}[\mathrm{x}] = \mathbb{K}[x_1, \ldots, x_n]$ denotes the ring of polynomials in the variables $x_1, \ldots, x_n$ with coefficients in $\mathbb{K}$; $\mathbb{K}[\mathrm{x}]_{\leq d}$ and $\mathbb{K}[\mathrm{x}]_d$ the $\mathbb{K}$−vector spaces of polynomials of degree at most d and the space of homogeneous polynomials of degree d respectively. The *dual* of $\mathbb{K}[\mathrm{x}]$, the set of $\mathbb{K}$−linear forms on $\mathbb{K}[\mathrm{x}]$, is denoted by $\mathbb{K}[\mathrm{x}]^*$. A typical example of a linear form on $\mathbb{K}[\mathrm{x}]$ is the evaluation e_ξ at a point ξ of $\mathbb{K}^n$: $e_\xi(p) = p(\xi)$.

$\mathbb{K}[\mathrm{x}]^*$ can be identified with the ring of formal power series $\mathbb{K}[[\partial]] = \mathbb{K}[[\partial_1, \ldots, \partial_r]]$, with the understanding that $\partial^\beta(x^\alpha) = \alpha!$ or 0 according to whether $\alpha = \beta$ or not. Concomitantly $\mathbb{K}[\mathrm{x}]$ is equipped with the apolar product that is defined, for $p = \sum_\alpha p_\alpha x^\alpha$ and $q = \sum_\alpha q_\alpha x^\alpha$, by $\langle p, q\rangle := \overline{p}(\partial)q = \sum_\alpha \alpha! \overline{p}_\alpha q_\alpha \in \mathbb{K}$.

If $\mathcal{P}$ is a (homogeneous) basis of $\mathbb{K}[\mathrm{x}]$ we denote $\mathcal{P}^\dagger$ its dual with respect to this scalar product. For $\lambda \in \mathbb{K}[\mathrm{x}]^*$ we can write $\lambda = \sum_{p\in\mathcal{P}} \lambda(p)\, p^\dagger(\partial)$.

An *interpolation problem* is a pair (Λ, ϕ) where Λ is a finite dimensional linear subspace of $\mathbb{K}[\mathrm{x}]^*$ and $\phi : \Lambda \longrightarrow \mathbb{K}$ is a $\mathbb{K}$-linear map. An interpolant, *i.e.,* a solution to the interpolation problem, is a polynomial p such that $\lambda(p) = \phi(\lambda)$ for any $\lambda \in \Lambda$. An *interpolation space for* Λ is a polynomial subspace P of $\mathbb{K}[\mathrm{x}]$ such that there is a unique interpolant for any map ϕ.

The *least interpolation space* $\Lambda_\downarrow$ was introduced in [7], and revisited in [33]. The least term $\lambda_\downarrow \in \mathbb{K}[\mathrm{x}]$ of a power series $\lambda \in \mathbb{K}[[\partial]]$ is the unique homogeneous polynomial for which $\lambda - \lambda_\downarrow(\partial)$ vanishes to highest possible order at the origin. Given a linear space of linear forms Λ, we define $\Lambda_\downarrow$ as the linear span of all $\lambda_\downarrow$ with $\lambda \in \Lambda$.

If $\mathcal{L} = \{\lambda_1, \lambda_2, \ldots, \lambda_r\}$ is a basis of Λ and $\mathcal{P} = \{p_1, p_2, \ldots, p_r\} \subset \mathbb{K}[\mathrm{x}]$, then $\mathcal{P}$ is a basis for an interpolation space of Λ if and only if the *Vandermonde matrix*

$$\mathrm{W}_{\mathcal{L}}^{\mathcal{P}} := \left[\lambda_i\,(p_j)\right]_{\substack{1\leq i\leq r\\ 1\leq j\leq r}} \tag{2}$$

is invertible. This latter is to be interpreted as the matrix in the bases $\mathcal{P}$ and the dual of $\mathcal{L}$ of the restriction of the Vandermonde operator $w : \mathbb{K}[\mathrm{x}] \to \Lambda^*$ such that $w(p)(\lambda) = \lambda(p)$. This is the adjoint of embedding $\Lambda \hookrightarrow \mathbb{K}[\mathrm{x}]^*$ and hence is surjective.

All along this paper we shall assume that

$$\mathcal{I} = \ker w = \cap_{\lambda\in\Lambda} \ker \lambda$$

is an ideal. When for instance $\Lambda = \langle e_{\xi_1}, \ldots, e_{\xi_r}\rangle_K$ then $\mathcal{I}$ is the ideal of the points $\{\xi_1, \ldots, \xi_r\} \subset \mathbb{K}[\mathrm{x}]$. One sees in general that $\dim \mathbb{K}[\mathrm{x}]/\mathcal{I} = \dim \Lambda^* = \dim \Lambda =: r$.

With $\mathcal{Q} = \{q_1, \ldots, q_r\} \subset \mathbb{K}[\mathrm{x}]$, we can identify $\mathbb{K}[\mathrm{x}]/\mathcal{I}$ with $\langle \mathcal{Q}\rangle_{\mathbb{K}}$ if $\langle \mathcal{Q}\rangle_{\mathbb{K}} \oplus \mathcal{I} = \mathbb{K}[\mathrm{x}]$. With a slight shortcut, we say that $\mathcal{Q}$ is a basis for $\mathbb{K}[\mathrm{x}]/\mathcal{I}$.

Proposition 2.1. *$\mathcal{Q} = \{q_1, \ldots, q_r\} \subset \mathbb{K}[\mathrm{x}]$ spans an interpolation space for Λ iff it is a basis for the quotient $\mathbb{K}[\mathrm{x}]/\mathcal{I}$.*

Proof. If $\mathcal{Q} = \{q_1, \ldots, q_r\}$ is a basis of $\mathbb{K}[\mathrm{x}]/\mathcal{I}$ then for any $p \in \mathbb{K}[\mathrm{x}]$ there is a $q \in \langle q_1, \ldots, q_r\rangle_{\mathbb{K}}$ such that $p \equiv q \bmod \mathcal{I}$. Hence $\lambda(p) = \lambda(q)$ for any $\lambda \in \Lambda$ and thus $\langle \mathcal{Q}\rangle_{\mathbb{K}}$ is an interpolation space for Λ. Conversely if $\langle q_1, \ldots, q_r\rangle_{\mathbb{K}}$ is an interpolation space for Λ then $\{q_1, \ldots, q_r\}$ are linearly independent modulo $\mathcal{I}$ and therefore a basis for $\mathbb{K}[\mathrm{x}]/\mathcal{I}$. Indeed if $q = a_1q_1 + \ldots + a_rq_r \in \mathcal{I}$ then any interpolation problem has multiple solutions in $\langle \mathcal{Q}\rangle_{\mathbb{K}}$, i.e, if p is the solution of (Λ, ϕ) so is $p + q$, contradicting the interpolation uniqueness on $\langle \mathcal{Q}\rangle_{\mathbb{K}}$. □

For $p \in \mathbb{K}[\mathrm{x}]$ we can find its natural projection on $\mathbb{K}[\mathrm{x}]/\mathcal{I}$ by taking the unique $q \in \langle \mathcal{Q}\rangle_{\mathbb{K}}$ that satisfies $\lambda(q) = \lambda(p)$ for all $\lambda \in \Lambda$. From a computational point of view, q is obtained by solving the Vandermonde system, i.e.,

$$q = (q_1, \ldots, q_r)\left(\mathrm{W}_{\mathcal{L}}^{\mathcal{Q}}\right)^{-1}\begin{pmatrix}\lambda_1(p)\\ \vdots\\ \lambda_r(p)\end{pmatrix} \text{ with } \mathcal{L} = \{\lambda_1, \ldots, \lambda_r\} \text{ a basis of } \Lambda.$$

Similarly, the matrix of the multiplication map, in the basis $\mathcal{Q}$, is

$$\begin{array}{rccc} \mathfrak{m}_p : & \mathbb{K}[\mathrm{x}]/\mathcal{I} & \to & \mathbb{K}[\mathrm{x}]/\mathcal{I},\\ & [q] & \mapsto & [pq]\end{array}$$

is obtained as $[\mathfrak{m}_p]_{\mathcal{Q}} = \left(\mathrm{W}_{\mathcal{L}}^{\mathcal{Q}}\right)^{-1} \mathrm{W}_{\mathcal{L}\circ m_p}^{\mathcal{Q}}$ where $\mathcal{L} \circ m_p = \{\lambda_1 \circ m_p, \ldots, \lambda_r \circ m_p\}$.

When working with Gröbner bases, one fixes a term order and focuses on leading terms of polynomials and the initial ideal of $\mathcal{I}$. The basis of choice for $\mathbb{K}[\mathrm{x}]/\mathcal{I}$ consists of the monomials that do not belong to the initial ideal. An H-basis of $\mathcal{I}$ is somehow the complement of the least interpolation space $\Lambda_\downarrow$ and hence can

be made to reflect the possible invariance of Λ and $\mathcal{I}$. Instead of leading terms, the focus is then on the leading homogeneous forms.

Hereafter we denote by p^0 the leading homogeneous form of p, i.e., the unique homogeneous polynomial such that $\deg\left(p - p^0\right) < \deg(p)$. Given a set of polynomials P we denote $P^0 = \left\{p^0 \mid p \in P\right\}$.

Proposition 2.2. *Let Q be an interpolation space of minimal degree for Λ. Then $Q \oplus \mathcal{I}^0 = \mathbb{K}[\mathrm{x}]$.*

Proof. We proceed by induction on the degree, i.e, we assume that any polynomial p in $\mathbb{K}[\mathrm{x}]_{\leq d}$ can be written as $p = q + l$ where $q \in Q$ and $l \in \mathcal{I}^0$. Note that the hypothesis holds trivially when d is equal to zero.

Now let $p \in \mathbb{K}[\mathrm{x}]_{\leq d+1}$. Since $\mathbb{K}[\mathrm{x}] = \langle Q \rangle_{\mathbb{K}} \oplus \mathcal{I}$ there exists $q \in Q$ and $l \in \mathcal{I}$ such that $p = q + l$. Since Q is of minimal degree, q and l are in $\mathbb{K}[\mathrm{x}]_{\leq d+1}$. Writing $l = l^0 + l_1$ he have $p = q + l^0 + l_1$ with $l_1 \in \mathbb{K}[\mathrm{x}]_{\leq d}$ then by induction $l_1 = q_1 + l_2$ with $q_1 \in Q$ and $l_2 \in \mathcal{I}^0$ and therefore $p = q + q_1 + l^0 + l_2 \in Q \oplus I^0$. □

As a consequence we retrieve the result of [7, Theorem 4.8].

Corollary 2.3. *Considering orthogonality with respect to the apolar product it holds that $\Lambda_{\downarrow} \overset{\perp}{\oplus} \mathcal{I}^0 = \mathbb{K}[\mathrm{x}]$.*

Proof. Follows from the fact that $\lambda(p) = 0 \Rightarrow \langle \lambda_{\downarrow}, p^0 \rangle = 0$. □

3 H-BASES

H-bases were introduced by [24]. The use of H-basis in interpolation has been further studied in [27, 34]. In this section we review the definitions and present the sketch of an algorithm to compute the H-basis of $\mathcal{I} = \bigcap_{\lambda \in \Lambda} \ker \lambda$.

Definition 3.1. A finite set $\mathcal{H} := \{h_1, \ldots, h_m\} \subset \mathbb{K}[\mathrm{x}]$ is an H-basis of the ideal $\mathcal{I} := \langle h_1, \ldots, h_m \rangle$ if, for all $p \in \mathcal{I}$ there are $g_1, \ldots g_m$ such that,

$$p = \sum_{i=1}^{m} h_i g_i \text{ and } \deg(h_i) + \deg(g_i) \leq \deg(p), i = 1, \ldots, m.$$

Theorem 3.2. *[27] Let $\mathcal{H} := \{h_1, \ldots, h_m\}$ and $\mathcal{I} := \langle \mathcal{H} \rangle$. Then the following conditions are equivalent:*

(1) *$\mathcal{H}$ is an H-basis of $\mathcal{I}$.*
(2) *$\mathcal{I}^0 := \left\langle \left\{h^0 \mid h \in \mathcal{I}\right\} \right\rangle = \left\langle h_1^0, \ldots, h_m^0 \right\rangle$.*

Hilbert Basis Theorem says that $\mathcal{I}^0$ has a finite basis, hence any ideal in $\mathbb{K}[\mathrm{x}]$ has a finite H-basis. We shall now introduce the concepts of minimal, orthogonal and reduced H-basis. The notion of orthogonality is considered w.r.t the apolar product. Our definitions somewhat differ from [27] as we dissociate them from the computational aspect. We need to introduce first the following vector space of homogeneous polynomials.

Definition 3.3. Given a set $\mathcal{H} = \{h_1, \ldots, h_m\}$ of homogeneous polynomials in $\mathbb{K}[\mathrm{x}]$ and a degree d, we define the subspace $\mathrm{V}_d(\mathcal{H})$ as

$$\mathrm{V}_d(\mathcal{H}) = \left\{ \sum_{i=1}^{s} g_i h_i \;\middle|\; g_i \in \mathbb{K}[\mathrm{x}]_{d-\deg(h_i)} \right\} \subset \mathbb{K}[\mathrm{x}]_d.$$

$\mathrm{V}_d(\mathcal{H})$ is the image of the linear map ψ_d:

$$\begin{array}{rccc} \psi_{d,h}: & \mathbb{K}[\mathrm{x}]_{d-d_1} \times \ldots \times \mathbb{K}[\mathrm{x}]_{d-d_m} & \to & \mathbb{K}[\mathrm{x}]_d \\ & (g_1, \ldots, g_m) & \to & \sum_{i=1}^{m} g_i h_i \end{array}.$$

We denote by $\mathrm{M}_{\mathcal{M}_d, \mathcal{P}_d}(\mathcal{H})$ the matrix of ψ_d in the bases $\mathcal{M}_d$ and $\mathcal{P}_d$ of $\mathbb{K}[\mathrm{x}]_{d-d_1} \times \ldots \times \mathbb{K}[\mathrm{x}]_{d-d_m}$ and $\mathbb{K}[\mathrm{x}]_d$ respectively. It is referred to as the Macaulay matrix for $\mathcal{H}$. We can write $\mathrm{V}_d(\mathcal{H})$ as

$$\mathrm{V}_d(\mathcal{H}) = \left\{ \sum_{i=0}^{|\mathcal{P}_d|} a_i p_i \;\middle|\; (a_1, \ldots, a_{|\mathcal{P}_d|}) \in \mathcal{R}\left(\mathrm{M}_{\mathcal{M}_d, \mathcal{P}_d}(\mathcal{H})\right) \right\},$$

where $\mathcal{R}\left(\mathrm{M}_{\mathcal{M}_d, \mathcal{P}_d}(\mathcal{H})\right)$ denotes the column space of $\mathrm{M}_{\mathcal{M}_d, \mathcal{P}_d}(\mathcal{H})$.

We shall use the notation P_d^0 for the set of the degree d elements of P^0. In other words $P_d^0 = P^0 \cap \mathbb{K}[\mathrm{x}]_d$.

Definition 3.4. We say that an H-basis $\mathcal{H}$ is minimal if, for any $d \in \mathbb{N}$, $\mathcal{H}_d^0$ is linearly independent and

$$V_d\left(\mathcal{I}_{d-1}^0\right) \oplus \left\langle \mathcal{H}_d^0 \right\rangle_{\mathbb{K}} = \mathcal{I}_d^0. \tag{3}$$

Furthermore $\mathcal{H}$ is said to be orthogonal if $\left\langle \mathcal{H}_d^0 \right\rangle_{\mathbb{K}}$ is the orthogonal complement of $V_d\left(\mathcal{I}_{d-1}^0\right)$ in $\mathcal{I}_d^0$.

Note that if h_i and h_j are two elements with $\deg h_i > \deg h_j$ of an orthogonal H-basis we have

$$\left\langle h_i^0, p h_j^0 \right\rangle = 0 \text{ for all } p \in \mathbb{K}[\mathrm{x}]_{\deg h_i - \deg h_j}.$$

Definition 3.5. Let $\mathcal{H} = \{h_1, \ldots, h_m\}$ be an orthogonal H-basis of an ideal $\mathcal{I}$. The *reduced* H-basis of $\mathcal{H}$ is defined by

$$\widetilde{\mathcal{H}} = \left\{ h_1^0 - \widetilde{h_1^0}, \ldots, h_m^0 - \widetilde{h_m^0} \right\} \tag{4}$$

where, for $p \in \mathbb{K}[x]$, $\tilde{p}$ is the projection of p on the orthogonal complement of $\mathcal{I}^0$ parallel to $\mathcal{I}$.

[27, Lemma 6.2] show how $\tilde{p}$ can be computed given $\mathcal{H}$.

Schematic computation of H-bases. In the next section we elaborate on an algorithm to compute concomitantly the least interpolation space and an H-basis for the ideal associated to a set of linear forms Λ. As a way of introduction we reproduce the sketch of an algorithm as proposed by [5] to compute an H-basis until degree D. It is based on the asumption that we have access to a basis of $\mathcal{I}_d := \mathcal{I} \cap \mathbb{K}[\mathrm{x}]_{\leq d}$ for any d.

Algorithm 1 [5] H-basis construction

Input: - a degree D.
- basis for $\mathcal{I}_d$ for $1 \leq d \leq D$.
Output : - an H-basis until degree D

$\mathcal{H} \leftarrow \{\}$;
for $d = 0$ **to** D **do**
  $C_d \leftarrow$ a basis of $\mathrm{V}_d(\mathcal{H}^0)$;
  $\mathcal{B}_d \leftarrow$ a basis for the complement of $\mathrm{V}_d(\mathcal{H})$ in $\mathcal{I}_d^0$;
  $\widehat{\mathcal{B}}_d \leftarrow$ projection of $\mathcal{B}_d$ in $\mathcal{I}_d$
  $\mathcal{H} \leftarrow \mathcal{H} \bigcup \widehat{\mathcal{B}}_d$;
return $\mathcal{H}$;

The correctness of Algorithm 1 is shown by induction. Assume that $\mathcal{H}_{d-1}$ consists of the polynomials in an H-basis of $\mathcal{I}$ up to degree $d-1$. Consider $p \in \mathcal{I}$ with $\deg(p) = d$. By Step 4 in Algorithm 1 we have

$$p^0 = \sum_{h_i \in \mathcal{H}} h_i^0 g_i + \sum_{b_i \in \mathcal{B}_d} a_i b_i \tag{5}$$

with $g_i \in \mathbb{K}[x]_{d-\deg(h_i)}$ and $a_i \in \mathbb{K}$. From (5) we have that $p \in \mathcal{I}$ and $\sum_{h_i \in \mathcal{H}} h_i g_i + \sum_{b_i \in \mathcal{B}_{d+1}} a_i \hat{b}_i \in \mathcal{I}$ have the same leading form. Thus

$$p - \sum_{h_i \in \mathcal{H}_{d-1}} h_i g_i - \sum_{b_i \in \mathcal{B}_d} a_i \hat{b}_i \in \mathcal{I}_{d-1}$$

therefore using the induction hypothesis we get that

$$p = \sum_{h_i \in \mathcal{H}_{d-1}} h_i g_i + \sum_{b_i \in \mathcal{B}_{d+1}} a_i \hat{b}_i + \sum_{h_i \in \mathcal{H}_{d-1}} h_i q_i$$

with $q_i \in \mathbb{K}[x]_{\leq d-1-\deg(h_i)}$ and therefore $\mathcal{H}$ is an H-basis.

Algorithm 1 can be applied in the ideal interpolation scheme. In this setting a basis of $\mathcal{I}_d$ can be computed for any d using Linear Algebra techniques due to the following relation.

$$\mathcal{I}_d = \left\{ \sum_{i=1}^{|\mathcal{P}_{\leq d}|} a_i p_i \mid \left(a_1, \ldots, a_{|\mathcal{P}_{\leq d}|}\right)^t \in \ker\left(\mathrm{W}_{\mathcal{L}}^{\mathcal{P}_{\leq d}}\right) \text{ and } p_i \in \mathcal{P}_{\leq d} \right\},$$

for any basis $\mathcal{P}_{\leq d}$ of $\mathbb{K}[x]_{\leq d}$.

In the next section we will give an efficient and detailed version of Algorithm 1 in the ideal interpolation case. We will integrate the computations of an H-basis for $\mathcal{I} = \cap_{\lambda \in \Lambda} \ker \lambda$ and a basis for $\Lambda_\downarrow$.

When the ideal is given by a set of generators it is also possible to compute an H-basis with linear algebra if you know a bound on the degree of the syzygies of the generators. A numerical approach, using singular value decomposition, was introduced in [22]. Alternatively an extension of Buchberger's algorithm is presented in [27]. It relies, at each step, on the computation of a basis for the module of syzygies of a set of homogeneous polynomials.

4 SIMULTANEOUS COMPUTATION OF THE H-BASIS AND LEAST INTERPOLATION SPACE

In this section we present an algorithm to compute both a (orthogonal) basis of $\Lambda_\downarrow$ and an orthogonal H-basis $\mathcal{H}$ of the ideal $\mathcal{I} = \cap_{\lambda \in \Lambda} \ker \lambda$. We proceed degree by degree. At each iteration of the algorithm we compute a basis of $\Lambda_\downarrow \cap \mathbb{K}[x]_d$ and the set $\mathcal{H}_d^0 = \mathcal{H}^0 \cap \mathbb{K}[x]_d$. Recall from Corollary 2.3, Theorem 3.2, and Definition 3.4 that

$$\mathbb{K}[x] = \Lambda_\downarrow \overset{\perp}{\oplus} \mathcal{I}^0, \quad \mathcal{I}^0 = \langle \mathcal{H}^0 \rangle, \quad \text{and} \quad \mathcal{I}_d^0 = V_d\left(\mathcal{I}_{d-1}^0\right) \overset{\perp}{\oplus} \left\langle \mathcal{H}_d^0 \right\rangle_{\mathbb{K}}.$$

$\mathcal{I}$ is the kernel of the Vandermonde operator while $\Lambda_\downarrow$ can be inferred from a rank revealing form of the Vandermonde matrix. With orthogonality prevailing in the objects we compute it is natural that the QR-decomposition plays a central role in our algorithm.

For a $m \times n$ matrix M, the QR-decomposition is $\mathrm{M} = \mathrm{QR}$ where Q is a $m \times m$ orthogonal matrix and R is a $m \times n$ upper triangular matrix. If r is the rank of M the first r columns of Q form an orthogonal basis of the column space of M and the remaining $m - r$ columns of Q form an orthogonal basis of the kernel of M^T [18, Theorem 5.2.1]. We thus often denote the QR-decomposition of a matrix M as

$$[\mathrm{Q}_1 \mid \mathrm{Q}_2] \cdot \begin{bmatrix} \mathrm{R} \\ 0 \end{bmatrix} = \mathrm{M}$$

where $\mathrm{Q}_1 \in \mathbb{K}^{m \times r}, \mathrm{Q}_2 \in \mathbb{K}^{m \times (m-r)}$ and $\mathrm{R} \in \mathbb{K}^{r \times n}$. Algorithms to compute the QR-decomposition can be found for instance in [18].

In the Lagrange interpolation case, Fassino and Möller [8] already used the QR-decomposition to propose a variant of the BM-algorithm [26] so as to compute a monomial basis of an interpolation space, the complement of the initial ideal for a chosen term order. They furthermore study the gain in numerical stability for perturbed data. We shall use QR-decomposition to further obtain a homogeneous basis of $\Lambda_\downarrow$ and an orthogonal H-basis of the ideal.

Due to Corollary 2.3 the reduction $\tilde{p}$ of p that appeared in Definition 3.5 is the unique interpolant of p in $\Lambda_\downarrow$.

Definition 4.1. Given a space of linear forms Λ, we denote by $\Lambda_{\geq d}$ the subspace of Λ given by

$$\Lambda_{\geq d} = \left\{ \lambda \in \Lambda \mid \lambda_\downarrow \in \mathbb{K}[x]_{\geqslant d} \right\} \cup \{0\}.$$

Hereafter we organize the elements of the bases of $\mathbb{K}[x]$, Λ, or their subspaces, as row vectors. In particular $\mathcal{P}$ and $\mathcal{P}^\dagger$ are dual homogeneous bases for $\mathbb{K}[x]$ according to the apolar product. Their degree part $\mathcal{P}_d$ and $\mathcal{P}_d^\dagger$ are dual bases of $\mathbb{K}[x]_d$.

A basis $\mathcal{L}_{\geq d}$ of $\Lambda_{\geq d}$ can be computed inductively thanks to the following observation.

Proposition 4.2. *Assume $\mathcal{L}_{\geq d}$ is a basis of $\Lambda_{\geq d}$. Consider the QR-decomposition*

$$\mathrm{W}_{\mathcal{L}_{\geq d}}^{\mathcal{P}_d} = [\mathrm{Q}_1 \mid \mathrm{Q}_2] \cdot \begin{bmatrix} \mathrm{R}_d \\ 0 \end{bmatrix}$$

and the related change of basis $[\mathcal{L}_d \mid \mathcal{L}_{\geq d+1}] = \mathcal{L}_{\geq d} \cdot [\mathrm{Q}_1 \mid \mathrm{Q}_2]$. Then

- *$\mathcal{L}_{\geq d+1}$ is a basis of $\Lambda_{\geq d+1}$;*
- *$\mathrm{R}_d = \mathrm{W}_{\mathcal{L}_d}^{\mathcal{P}_d}$ has full row rank;*
- *The components of $\mathcal{L}_{d\downarrow} = \mathcal{P}_d^\dagger \cdot R_d^T$ form a basis of $\Lambda_\downarrow \cap \mathbb{K}[x]_d$.*

We shall furthermore denote by $\mathcal{L}_{\leq d} = \bigcup_{i=0}^d \mathcal{L}_i$ the thus constructed basis of a complement of $\Lambda_{\geq d+1}$ in Λ.

Proof. It all follows from the fact that a change of basis $\mathcal{L}' = \mathcal{L}\mathrm{Q}$ of Λ implies that $\mathrm{W}_{\mathcal{L}'}^{\mathcal{P}} = \mathrm{Q}^T \mathrm{W}_{\mathcal{L}}^{\mathcal{P}}$. In the present case $Q = [\mathrm{Q}_1 \mid \mathrm{Q}_2]$ is orthogonal and hence $Q^T = Q^{-1}$.

The last point simply follows from the fact that, for $\lambda \in \Lambda$, $\lambda = \sum_{p \in \mathcal{P}} \lambda(p) p^\dagger(\partial)$. Hence if $T = \mathrm{W}_{\mathcal{L}}^{\mathcal{P}}$ then the j-th component of $\mathcal{L}$ is $\sum_i t_{ji} p^\dagger(\partial)$. □

This construction gives us a basis of $\Lambda_\downarrow \cap \mathbb{K}[x]_d$ in addition to a basis of $\Lambda_{\geq d+1}$ to pursue the computation at the next degree. Before going there, we need to compute a basis $\mathcal{H}_d^0$ for the complement of $V_d(\mathcal{H}_{<d}^0)$ in $\mathcal{I}_d^0$. For that we shall use an additional QR-decomposition as explained in Proposition 4.5, after two preparatory lemmas.

Lemma 4.3. *Let $d \geq 0$ and let $\mathcal{P}_d$ be a basis of $\mathbb{K}[x]_d$ then:*

$$\mathcal{I}_d^0 = \left\{ \sum_{i=1}^{|\mathcal{P}_d|} a_i p_i \mid \left(a_1, \ldots, a_{|\mathcal{P}_d|}\right)^t \in \ker\left(\mathrm{W}_{\mathcal{L}_d}^{\mathcal{P}_d}\right) \text{ and } p_i \in \mathcal{P}_d \right\}.$$

Proof. Recall that $\mathcal{I}$ is the kernel of the Vandermonde operator, and $\mathrm{W}_{\mathcal{L}}^{\mathcal{P}}$ is the matrix of this latter. The Vandermonde submatrix $\mathrm{W}_{\mathcal{L}_{\leq d}}^{\mathcal{P}_{\leq d}}$ can be written as follows

$$\mathrm{W}_{\mathcal{L}_{\leq d}}^{\mathcal{P}_{\leq d}} = \mathrm{W}_{[\mathcal{L}_{\leq d-1} \mid \mathcal{L}_d]}^{\mathcal{P}_{\leq d}} = \begin{pmatrix} \mathrm{W}_{\mathcal{L}_{\leq d-1}}^{\mathcal{P}_{\leq d-1}} & \mathrm{W}_{\mathcal{L}_{\leq d-1}}^{\mathcal{P}_d} \\ 0 & \mathrm{W}_{\mathcal{L}_d}^{\mathcal{P}_d} \end{pmatrix} \tag{6}$$

where $\mathrm{W}_{\mathcal{L}_{\leq d-1}}^{\mathcal{P}_{\leq d-1}}$ has full row rank.

Assume first that p is a polynomial in $\mathcal{I}_d^0$. Then there is $q \in \mathcal{I}$ of degree d such that $q^0 = p$. Let $\mathrm{q} = \begin{pmatrix} \mathrm{q}_{\leq d-1} \\ \mathrm{q}_d \end{pmatrix}$ and $\mathrm{p} = \mathrm{q}_d$ be the coefficients of q and p respectively in the basis $\mathcal{P}$. As $q \in \mathcal{I}_d$ we have that

$$\mathrm{W}_{\mathcal{L}_{\leq d}}^{\mathcal{P}_{\leq d}} \cdot \mathrm{q} = \begin{pmatrix} \mathrm{W}_{\mathcal{L}_{\leq d-1}}^{\mathcal{P}_{\leq (d-1)}} \cdot \mathrm{q}_{\leq d-1} + \mathrm{W}_{\mathcal{L}_{\leq d-1}}^{\mathcal{P}_d} \cdot \mathrm{q}_d \\ \mathrm{W}_{\mathcal{L}_d}^{\mathcal{P}_d} \cdot \mathrm{q}_d \end{pmatrix} = 0$$

and therefore $\mathrm{p} = \mathrm{q}_d$ is in kernel of $\mathrm{W}_{\mathcal{L}_d}^{\mathcal{P}_d}$. Now let v a vector in the kernel of $\mathrm{W}_{\mathcal{L}_d}^{\mathcal{P}_d}$. A vector u such that $\begin{pmatrix} \mathrm{u} \\ \mathrm{v} \end{pmatrix} \in \mathbb{K}^{\binom{n+d}{d}}$ and $\mathrm{W}_{\mathcal{L}_{\leq d}}^{\mathcal{P}_{\leq d}} \cdot \begin{pmatrix} \mathrm{u} \\ \mathrm{v} \end{pmatrix} = 0$ can be found as the solution of the following equation.

$$\mathrm{W}_{\mathcal{L}_{\leq d-1}}^{\mathcal{P}_{\leq (d-1)}} \mathrm{u} = \mathrm{W}_{\mathcal{L}_d}^{\mathcal{P}_d} \mathrm{v} - \mathrm{W}_{\mathcal{L}_{\leq d-1}}^{\mathcal{P}_d} \mathrm{v}. \tag{7}$$

As $\mathrm{W}_{\mathcal{L}_{\leq d-1}}^{\mathcal{P}_{\leq (d-1)}}$ has full row rank, Equation 7 always has a solution. Then $\mathcal{P}_{\leq d} \cdot \begin{pmatrix} \mathrm{u} \\ \mathrm{v} \end{pmatrix} \in \mathcal{I}$ and therefore $\mathcal{P}_d \cdot \mathrm{v} \in \mathcal{I}_d^0$. □

Lemma 4.4. *Consider the row vector* q *of coefficients of a polynomial* q *of* $\mathbb{K}[\mathrm{x}]_d$ *in the basis* $\mathcal{P}_d$*. The polynomial* q *is in the orthogonal complement of* $\mathrm{V}_d(\mathcal{H})$ *in* $\mathbb{K}[\mathrm{x}]_d$ *if and only if the row vector* q *is in the left kernel of* $\mathrm{M}_{\mathcal{M}_d, \mathcal{P}_d^\dagger}(\mathcal{H})$.

Proof. The columns of $\mathrm{M}_{\mathcal{M}_d, \mathcal{P}_d^\dagger}$ are the vectors of coefficients, in the basis $\mathcal{P}_d^\dagger$, of polynomials that span $\mathrm{V}_d(\mathcal{H})$. The membership of q in the left kernel of $\mathrm{M}_{\mathcal{M}_d, \mathcal{P}_d^\dagger}(\mathcal{H})$ translates as the apolar product of q with these vectors to be zero. And conversely. □

Proposition 4.5. *Consider the QR-decomposition*

$$\begin{bmatrix} \left(\mathrm{W}_{\mathcal{L}_d}^{\mathcal{P}_d}\right)^T & \mathrm{M}_{\mathcal{M}_d, \mathcal{P}_d^\dagger}(\mathcal{H}) \end{bmatrix} = [\mathrm{Q}_1 \mid \mathrm{Q}_2] \cdot \begin{bmatrix} \mathrm{R} \\ 0 \end{bmatrix}$$

The components of the row vector $\mathcal{P}_d \cdot Q_2$ *span the orthogonal complement of* $\mathrm{V}_d(\mathcal{H})$ *in* $\mathcal{I}_d^0$.

Proof. The columns in Q_2 span $\ker \mathrm{W}_{\mathcal{L}_d}^{\mathcal{P}_d} \cap \ker \left(\mathrm{M}_{\mathcal{M}_d, \mathcal{P}_d^\dagger}\right)^t$. The result thus follows from Lemmas 4.3 and 4.4. □

We are now able to show the correctness and termination of Algorithm 2.

Algorithm 2

Input: - $\mathcal{L}$ a basis of Λ ($r = |\mathcal{L}| = \dim(\Lambda)$)
- $\mathcal{P}$ a basis of $\mathbb{K}[\mathrm{x}]_{\leq r}$
- $\mathcal{P}^\dagger$ the dual basis of $\mathcal{P}$ w.r.t the apolar product.

Output: - $\mathcal{H}$ a reduced H-basis for $\mathcal{I} := \ker \Lambda$
- $\mathcal{P}_\Lambda$ a basis of the least interpolation space of Λ.

$\mathcal{H}^0 \leftarrow \{\}$, $\mathcal{P}_\Lambda \leftarrow \{\}$
$d \leftarrow 0$
$\mathcal{L}_{\leq 0} \leftarrow \{\}$, $\mathcal{L}_{\geq 0} \leftarrow \mathcal{L}$
while $\mathcal{L}_{\geq d} \neq \{\}$ **do**
  $\mathrm{Q} \cdot \begin{bmatrix} \mathrm{R}_d \\ 0 \end{bmatrix} = \mathrm{W}_{\mathcal{L}_{\geq d}}^{\mathcal{P}_d}$ ▷ QR-decomposition of $\mathrm{W}_{\mathcal{L}_{\geq d}}^{\mathcal{P}_d}$
  $\mathcal{P}_\Lambda \leftarrow \mathcal{P}_\Lambda \bigcup \mathcal{P}_d^\dagger \cdot \mathrm{R}_d^T$
  $[\mathcal{L}_d \mid \mathcal{L}_{\geq d+1}] \leftarrow \mathcal{L}_{\geq d} \cdot \mathrm{Q}^T$ ▷ Note that $\mathrm{R}_d = \mathrm{W}_{\mathcal{L}_d}^{\mathcal{P}_d}$
  $\mathcal{L}_{\leq d+1} \leftarrow \mathcal{L}_{\leq d} \cup \mathcal{L}_d$
  $[\mathrm{Q}_1 \mid \mathrm{Q}_2] \cdot \mathrm{R} = \begin{bmatrix} \mathrm{R}_d^T & \mathrm{M}_{\mathcal{M}_d, \mathcal{P}_d^\dagger}(\mathcal{H}) \end{bmatrix}$
  $\mathcal{H}^0 \leftarrow \mathcal{H}^0 \bigcup \mathcal{P}_d \cdot \mathrm{Q}_2$
  $d \leftarrow d + 1$
for all $p \in \mathcal{H}^0$ **do**
  $\mathcal{H} \leftarrow \mathcal{H} \bigcup \left\{ p - \mathcal{P}_\Lambda \left(\mathrm{W}_{\mathcal{L}_{\leq d}}^{\mathcal{P}_\Lambda}\right)^{-1} (\mathcal{L}_{\leq d})^T \right\}$
return $(\mathcal{H}, \mathcal{P}_\Lambda)$

Correctness. In the spirit of Algorithm 1, Algorithm 2 proceeds degree by degree. At the iteration for degree d we first compute a basis for $\Lambda_{\geq d+1}$ by splitting $\mathcal{L}_{\geq d}$ into $\mathcal{L}_{\geq d+1}$ and $\mathcal{L}_d$. As explained in Proposition 4.2, this is obtained through the QR-decomposition of $\mathrm{W}_{\mathcal{L}_{\geq d}}^{\mathcal{P}_d}$. From this decomposition we also obtain a basis for $\Lambda_\downarrow \cap \mathbb{K}[\mathrm{x}]_d$ as well as $\mathrm{W}_{\mathcal{L}_d}^{\mathcal{P}_d}$. We then go after $\mathcal{H}_d^0$, which spans the orthogonal complement of $\mathrm{V}_d(\mathcal{H}_{\leq d-1}^0)$ in $\mathcal{I}_d^0$. The elements of $\mathcal{H}_d^0$ are computed via intersection of $\ker \mathrm{W}_{\mathcal{L}_d}^{\mathcal{P}_d}$ and $\ker \left(\mathrm{M}_{\mathcal{M}_d, \mathcal{P}_d^\dagger}\right)^t$ as showed in Proposition 4.5. Algorithm 2 stops when we reach a degree δ such that $\mathcal{L}_{\geq \delta}$ is empty. Notice that for $d \geq \delta$ the matrix $\mathrm{W}_{\mathcal{L}_d}^{\mathcal{P}_d}$ is an empty matrix and therefore its kernel is the full space $\mathbb{K}[\mathrm{x}]_d$. Then as a consequence of Lemma 4.3, for all $d > \delta$ we have that $V_d(\mathcal{I}_{d-1}^0) = \mathcal{I}_d^0$ hence $\langle \mathcal{H}_d^0 \rangle$ is an empty set. The latter implies that when the algorithm stops we have computed the full H-basis $\mathcal{H}^0$ for $\mathcal{I}^0$.

We then obtain an H-basis of $\mathcal{I}$ by finding the projections, onto $\Lambda_\downarrow$ and parallel to $\mathcal{I}$, of the elements of $\mathcal{H}^0$. These are the polynomials of $\Lambda_\downarrow$ interpolating the elements of $\mathcal{H}^0$ according to Λ.

Termination. Considering $r := \dim(\Lambda)$ we have that $\mathcal{L}_{\geq r}$ is an empty set, this implies that in the worst case our algorithm stops after r iterations.

Complexity. The most expensive computational step in Algorithms 2 is the computation of the kernel of $\left[\left(\mathrm{W}_{\mathcal{L}_d}^{\mathcal{P}_d}\right)^T \mathrm{M}_{\mathcal{M}_d, \mathcal{P}_d^\dagger}(\mathcal{H})\right]$, with number of columns and rows given by

$$\begin{aligned} \mathrm{row}(d) &= \tbinom{d+n-1}{n-1} = \tfrac{d^{n-1}}{(n-1)!} + \mathcal{O}\left(d^{n-1}\right) \\ \mathrm{col}(d) &= \textstyle\sum_{i=1}^{|\mathcal{H}|} \tbinom{d-d_i+n-1}{n-1} + |\mathcal{L}_d| = \tfrac{|\mathcal{H}| d^{n-1}}{(n-1)!} + \mathcal{O}\left(d^{n-1}\right) \end{aligned} \tag{8}$$

where $d_1, \ldots, d_{|\mathcal{H}|}$ are the degrees of the elements of the computed H-basis until degree d. Then the computational complexity of Algorithm 2 relies on the method used for the kernel computation of $VM(d)$, which in our case is the QR-decomposition.

We are giving a frame for the simultaneous computation of an H-basis and the Least interpolation space, but there is still room for improving the performance of Algorithm 2. The structure of the Macaulay matrix might be taken into account to alleviate the linear algebra operations as for instance in [1]. We can also consider different variants of Algorithm 2. In Proposition 4.6 we show that orthogonal bases for $\mathbb{K}[\mathrm{x}]_d \cap \Lambda_\downarrow$ and $\mathcal{I}_d^0$ can be simultaneously computed by applying QR-decomposition in the Vandermonde matrix $(\mathrm{W}_{\mathcal{L}_{\geq d}}^{\mathcal{P}_d})^T$. Therefore we can split Step 9 in two steps. First we do a QR-decomposition $(\mathrm{W}_{\mathcal{L}_{\geq d}}^{\mathcal{P}_d})^T$ to obtain orthogonal bases of $\mathbb{K}[\mathrm{x}]_d \cap \Lambda_\downarrow$ and $\mathcal{I}_d^0$. Once that we have in hand a basis of $\mathcal{I}_d^0$ we obtain the elements of $\mathcal{H}_d$ as its complement in the column space of $\mathrm{M}_{\mathcal{M}_d, \mathcal{P}_d^\dagger}(\mathcal{H})$.

Proposition 4.6. *Let* $[\mathrm{Q}_1 \mid \mathrm{Q}_2] \cdot \begin{bmatrix} \mathrm{R}_d \\ 0 \end{bmatrix} = \left(\mathrm{W}_{\mathcal{L}_{\geq d}}^{\mathcal{P}_d}\right)^T$ *be a QR-decomposition of* $\left(\mathrm{W}_{\mathcal{L}_{\geq d}}^{\mathcal{P}_d}\right)^T$. *Let* r *be the rank of* $\left(\mathrm{W}_{\mathcal{L}_{\geq d}}^{\mathcal{P}_d}\right)^T$. *Let* $\{q_1 \ldots q_r\}$ *and* $\{q_{r+1} \ldots q_m\}$ *be the columns of* Q_1 *and* Q_2 *respectively. Then the following holds:*

1. $\mathcal{P}_{\Lambda,d} = \left\{\mathcal{P}_d^\dagger \cdot q_1, \ldots, \mathcal{P}_d^\dagger \cdot q_r\right\}$ *is a basis of* $\mathbb{K}[\mathrm{x}]_d \cap \Lambda_\downarrow$.
2. $\mathcal{N} = \{\mathcal{P}_d \cdot q_{r+1}, \ldots, \mathcal{P}_d \cdot q_m\}$ *is a basis of* $\mathcal{I}_d^0$.
3. *If* $p \in \mathcal{P}_{\Lambda,d}$ *and* $q \in \mathcal{N}$ *then* $\langle p, q\rangle = 0$, *i.e.,* $\mathbb{K}[\mathrm{x}] = (\Lambda_\downarrow \cap \mathbb{K}[\mathrm{x}]_d) \overset{\perp}{\oplus} \mathcal{I}_d^0$.

In the case where $\mathcal{P}$ *is orthonormal with respect to the apolar product, i.e.* $\mathcal{P} = \mathcal{P}^\dagger$, *then* $\mathcal{P}_{\Lambda,d}$ *and* $\mathcal{N}$ *are also orthonormal bases.*

Proof. Let D such that $\mathcal{L}_{\geq D} = \{\}$ and let $\mathcal{L}_{\leq D} = \bigcup_{d \leq D} \mathcal{L}_d$ be a basis of Λ. Then the matrix $\mathrm{W}_{\mathcal{L}_{\leq D}}^{\mathcal{P}_{\leq D}}$ is block upper triangular with non singular diagonal blocks. Consider $\{a_1, \ldots a_\ell\} \in \mathbb{K}^{|\mathcal{P}_{\leq D}|}$ the rows of $\mathrm{W}_{\mathcal{L}_{\leq D}}^{\mathcal{P}_{\leq D}}$. By Proposition [33, Proposition 2.3] we have that $\mathcal{P}_\Lambda \left\{\left(\mathcal{P}_{\leq D}^\dagger \cdot a_1^t\right)_\downarrow, \ldots, \left(\mathcal{P}_{\leq D}^\dagger \cdot a_\ell^t\right)_\downarrow\right\}$ is a basis of $\Lambda_\downarrow$, we can rewrite $\mathcal{P}_\Lambda$ as $\bigcup_{d=1}^{D} \left\{\mathcal{P}_d^\dagger \cdot b_1^t, \ldots, \mathcal{P}_d^\dagger \cdot b_{\ell_d}^t\right\}$ where $\left\{b_1, \ldots, b_{\ell_d}\right\}$ is a basis of the row space of $\left(\mathrm{W}_{\mathcal{L}_d}^{\mathcal{P}_d}\right)$. Since $\mathcal{P}_\Lambda$ is a graded basis then $\left\{\mathcal{P}_d^\dagger \cdot b_1^t, \ldots, \mathcal{P}_d^\dagger \cdot b_{\ell_d}^t\right\}$ is a basis $\mathbb{K}[\mathrm{x}]_d \cap \Lambda_\downarrow$.

Part (2) in the proposition is a direct consequence of Lemma 4.3 and the fact that the columns of Q_2 form a basis of the kernel of $\mathrm{W}_{\mathcal{L}_{\geq d}}^{\mathcal{P}_d}$. Let now $q \in \mathcal{P}_{\Lambda,d}$ and $p \in \mathcal{N}$. Then,

$$\langle p, q\rangle = \left\langle \sum_{p_i \in \mathcal{P}_d} a_i p_i, \sum_{q_i \in \mathcal{P}_d^\dagger} b_i q_i \right\rangle = \sum_{i=1} a_i b_i = 0.$$

Last equality stems from a and b being different rows in Q. □

5 SYMMETRY REDUCTION

The symmetries we deal with are given by the linear action of a finite group G on $\mathbb{K}^n$. It is thus given by a representation ϑ of G on $\mathbb{K}^n$. It induces a representation ρ of G on $\mathbb{K}[\mathrm{x}]$ given by

$$\rho(g)p(x) = p(\vartheta(g^{-1})x). \tag{9}$$

It also induces a linear representation on the space of linear forms, the dual representation of ρ :

$$\rho^*(g)\lambda(p) = \lambda(\rho(g^{-1})p),\ p \in \mathbb{K}[\mathrm{x}] \text{ and } \lambda \in \mathbb{K}[\mathrm{x}]^*. \tag{10}$$

We shall deal with an invariant subspace Λ of $\mathbb{K}[\mathrm{x}]^*$. Hence the restriction of ρ^* to Λ is a linear representation of G in Λ.

In the above Algorithm 2, to compute an H-basis of $\mathcal{I} = \ker w$, we use the Vandermonde and Macaulay matrices. We showed in [33, Section 4.2] how the Vandermonde matrix can be block diagonalized using appropriate symmetry adapted bases of $\mathbb{K}[\mathrm{x}]$ and Λ. We show here how to obtain such a block diagonalization on the Macaulay matrix when the space spanned by $\mathcal{H}$ is invariant under the induced action of a group G on $\mathbb{K}[\mathrm{x}]$. The key relies on exhibiting the equivariance of the prolongation map $\Psi_{d,\mathrm{h}}$ defined in Section 3.

With notations compliant with [33], for any representation θ of a group G on a $\mathbb{K}$-vector space V, a *symmetry adapted basis* $\mathcal{P}$ of V is characterized by the fact that the matrix of the representation θ in $\mathcal{P}$ is

$$[\theta(g)]_{\mathcal{P}} = \mathrm{diag}\left(\mathrm{R}_1(g) \otimes \mathrm{I}_{c_1},\ \ldots,\ \mathrm{R}_N(g) \otimes \mathrm{I}_{c_N}\right).$$

where $\mathrm{R}_j = \left(r_{kl}^j\right)_{1 \leq k,l \leq n_j}$ is the matrix representation of the irreducible representation ρ_j of G and c_j is the multiplicity of ρ_j in θ. Hence $\mathcal{P} = \cup_{j=1}^N \mathcal{P}^j$ where $\mathcal{P}^j$ spans the isotypic component V_j associated to ρ_j. Introducing the map $\pi_{j,kl} = \frac{n_j}{|G|} \sum_{g \in G} r_{kl}^j(g^{-1})\theta(g)$ we can say that $\mathcal{P}^j$ is *determined* by $p_1^j, \ldots, p_{c_j}^j$ to mean that $p_1^j, \ldots, p_{c_j}^j$ is a basis of $\pi_{j,11}(V)$ and

$$\mathcal{P}^j = \{p_1^j, \ldots, p_{c_j}^j, \ldots, \pi_{j,n_j1}(p_1^j), \ldots, \pi_{j,n_j1}(p_{c_j}^j)\}. \tag{11}$$

When dealing with $\mathbb{K} = \mathbb{R}$, the statements we write are for the case where all the irreducible representations of G are absolutely irreducible, and thus the matrices $R_j(g)$ all have real entries. This is the case of all reflection groups. Yet these statements can be modified to also work with irreducible representations of complex type, which occur, for instance, for the cyclic group C_m with $m > 2$.

Consider now a set $\mathcal{H} = \{h_1, \ldots, h_l\}$ of homogeneous polynomials of $\mathbb{K}[\mathrm{x}]$. We denote $d_1, \ldots, d_\ell$ their respective degrees and $\mathrm{h} = [h_1, \ldots, h_\ell]$ the row vector of $\mathbb{K}[x]^\ell$. Associated to h, and a degree d, is the map introduced in Section 3

$$\begin{array}{rccl} \psi_{d,\mathrm{h}}: & \mathbb{K}[\mathrm{x}]_{d-d_1} \times \ldots \times \mathbb{K}[\mathrm{x}]_{d-d_\ell} & \to & \mathbb{K}[\mathrm{x}]_d \\ & \mathrm{f} = [f_1, \ldots, f_\ell]^t & \to & \mathrm{h} \cdot \mathrm{f}. \end{array} \tag{12}$$

We assume that $\mathcal{H}$ forms a basis of an invariant subspace of $\mathbb{K}[\mathrm{x}]$ and we call θ the restriction of the representation ρ to this subspace, while Θ is the matrix representation in the basis $\mathcal{H}$: $\Theta(g) = [\theta(g)]_{\mathcal{H}}$. Then $[\rho(g)(h_1), \ldots, \rho(g)(h_\ell)] = \mathrm{h} \circ \vartheta(g^{-1}) = \mathrm{h} \cdot \Theta(g)$. Note that, since the representation ρ on $\mathbb{K}[\mathrm{x}]$ preserves degree, $\deg h_i \neq \deg h_j \quad \Rightarrow \quad \Theta_{ij}(g) = 0,\ \forall g \in G$.

Proposition 5.1. *Consider* $\mathrm{h} = [h_1, \ldots, h_\ell] \in \mathbb{K}[x]_{d_1} \times \ldots \times \mathbb{K}[x]_{d_l}$ *and assume that* $\mathrm{h} \circ \vartheta(g^{-1}) = \mathrm{h} \cdot \Theta(g)$, *for all* $g \in G$. *For any* $d \in \mathbb{N}$, *the map* $\psi_{d,\mathrm{h}}$ *is* $\tau - \rho$ *equivariant for the representation* τ *on* $\mathbb{K}[\mathrm{x}]_{d-d_1} \times \ldots \times \mathbb{K}[\mathrm{x}]_{d-d_\ell}$ *defined by* $\tau(g)(\mathrm{f}) = \Theta(g) \cdot \mathrm{f} \circ \vartheta(g^{-1})$.

Proof. $(\rho(g)\circ\psi_{d,\mathrm{h}})(\mathrm{f}) = \rho(g)(\mathrm{h}\cdot\mathrm{f}) = \mathrm{h}\circ\vartheta(g^{-1})\cdot\mathrm{f}\circ\vartheta(g^{-1}) = \mathrm{h}\cdot\Theta(g)\cdot\mathrm{f}\circ\vartheta(g^{-1}) = (\psi_{\mathrm{h}}\circ\tau(g))(\mathrm{f})$. □

By application of [9, Theorem 2.5], the matrix of $\psi_{d,\mathrm{h}}$ is block diagonal in symmetry adapted bases of $\mathbb{K}[\mathrm{x}]_{d-d_1}\times\ldots\times\mathbb{K}[\mathrm{x}]_{d-d_\ell}$ and $\mathbb{K}[\mathrm{x}]_d$. Yet, in the algorithm to compute symmetry adapted H-basis, the set $\mathcal{H}$ increases with d at each iteration and τ changes accordingly. We proceed to discuss how to hasten the computation of a symmetry adapted basis of the evolving space $\mathbb{K}[\mathrm{x}]_{d-d_1}\times\ldots\times\mathbb{K}[\mathrm{x}]_{d-d_\ell}$.

The set $\mathcal{H}=\mathcal{H}^1\cup\ldots\mathcal{H}^N$ that we shall build, degree by degree, is actually a symmetry adapted basis. In particular, for $1\le i\le N$, $\mathcal{H}^i$ spans the isotypic component associated to the irreducible representation ρ_i. If the multiplicity of the latter, in the span of $\mathcal{H}$, is ℓ_i then the cardinality of $\mathcal{H}^i$ is $\ell_i\, n_i$. The matrices of the representation θ in this basis are $\Theta(g)=\mathrm{diag}(R_i(g)\otimes I_{\ell_i}|i=1\ldots N)$.

Assume $\mathcal{H}^i$ is determined by $h_{i,1},\ldots,h_{i,\ell_i}$, of respective degrees $d_{i,1},\ldots,d_{i,\ell_i}$. In other words, for $1\le l\le\ell_i$,

$$\mathrm{h}_{i,l}=\left[h_{i,l},\pi_{i,21}(h_{i,l}),\ldots,\pi_{i,n_i1}(h_{i,l})\right]$$

is such that $\mathrm{h}_{i,l}\circ\vartheta(g^{-1})=\mathrm{h}_{i,l}\cdot R_i(g)$. Hence the related product subspace $\mathbb{K}[\mathrm{x}]^{n_i}_{d-d_{i,l}}$ is invariant under τ. The symmetry adapted bases for all these subspaces can be combined into a symmetry adapted basis for the whole product space $\left(\mathbb{K}[\mathrm{x}]_{d_{1,1}}\times\mathbb{K}[\mathrm{x}]_{d_{1,\ell_1}}\right)^{n_1}\times\ldots\times\left(\mathbb{K}[\mathrm{x}]_{d_{N,1}}\times\mathbb{K}[\mathrm{x}]_{d_{1,\ell_N}}\right)^{n_N}$. Note that the components $\mathbb{K}[\mathrm{x}]^{n_i}_e$ with representation $\tau_{i,e}$ defined by $\tau_{i,e}(g)(\mathrm{f})=R_i(g)\cdot f\circ\vartheta(g^{-1})$ are bound to reappear several times in the overall algorithm of next section. Hence the symmetry adapted bases for the evolving τ can be computed dynamically.

6 CONSTRUCTING SYMMETRY ADAPTED H-BASIS

In this section we show, when the space Λ is invariant, an orthogonal equivariant H-basis $\mathcal{H}$ can be computed. In this setting, we exploit the symmetries of Λ to build $\mathcal{H}$. A robust and symmetry adapted version of Algorithm 2 is presented. The block diagonal structure of the Vandermonde and Macaulay matrices allow to reduce the size of the matrices to deal with. The H-basis obtained as the output of Algorithm 3 inherits the symmetries of Λ.

Proposition 6.1. *Let $\mathcal{I}=\cap_{\lambda\in\Lambda}\ker\lambda$ and $d\in\mathbb{N}$. If Λ is invariant, then so are $\mathcal{I}$, $\mathcal{I}^0$, $\mathcal{I}^0_d$, $V_d\left(\mathcal{I}^0_{<d}\right)$. Also, if $\mathcal{H}$ is an orthogonal H-basis of $\mathcal{I}$, then $\langle\mathcal{H}^0_d\rangle_{\mathbb{K}}$ is invariant.*

Proof. Let $p\in\mathcal{I}$ and $g\in G$, since Λ is closed under the action of G, $\lambda(\rho(g)(p))=\rho^*(g)\circ\lambda(p)=0$ for all $\lambda\in\Lambda$ therefore $\rho(g)(p)\in\mathcal{I}$ implying the invariance of $\mathcal{I}$. Considering d the degree of p we can write p as $p=p^0+p_1$, with $p_1\in\mathbb{K}[\mathrm{x}]_{<d}$. Then we have that $\rho(g)p=\rho(g)p^0+\rho(g)p_1\in\mathcal{I}$, as ρ is degree preserving then $\rho(g)p^0\in I^0_d$ and the invariance of $\mathcal{I}^0$ follows. Now for every $q=\sum_{h_i\in\mathcal{I}^0_{d-1}}q_ih_i\in V_d\left(\mathcal{I}^0_{\le d}\right)$, it holds that $\rho(g)q=\sum_{h_i\in\mathcal{I}^0_{d-1}}\rho(g)q_i\rho(g)h_i\subset V_d\left(\mathcal{I}^0_{\le d}\right)$, thus $V_d\left(\mathcal{I}^0_{\le d}\right)$ is an invariant subspace. Finally recalling (3) we conclude that $\langle\mathcal{H}^0_d\rangle_{\mathbb{K}}$ is also G-invariant for being the orthogonal complement of a G−invariant subspace. □

Algorithm 3 is a symmetry adapted version of Algorithm 2. In any iteration we compute $\mathcal{H}^0_d$ as a symmetry adapted basis of the orthogonal complement of $V_d(\mathcal{H}^0_{<d})$ in $\mathcal{I}^0$.

This structure is obtained degree by degree. Assuming that the elements of $\mathcal{H}^0_{<d}$ form a symmetry adapted basis it follows from [33, Section 4.2] and Proposition 5.1 that the matrices $\mathrm{W}^{\mathcal{P}_d}_{\mathcal{L}}$ and $\mathrm{M}_{\mathcal{M}_d,\mathcal{P}_d}(\mathcal{H}^0_{<d})$ are block diagonal. Computations over the symmetry blocks leads to the symmetry adapted structure of $\mathcal{H}^0_d$. For any degree d we only need to consider the matrices $\mathrm{W}^{\mathcal{P}^{i,1}_d}_{\mathcal{L}^{i,1}_{\ge d}}$ and $\mathrm{M}^i_d(\mathcal{H}^0_{<d})$, i.e., only one block per irreducible representation.

Once we have in hand $\mathcal{H}^0=\left[h^1_{11},\ldots,h^1_{1n_1},\ldots,h^N_{c_Nn_N}\right]^T$ and a symmetry adapted basis for $\Lambda_\downarrow$, we compute $\mathcal{H}$ by interpolation. Since $\mathcal{H}^0\in\mathbb{K}[\mathrm{x}]^\theta_\vartheta$, by [33, Proposition 3.5], its interpolant in $\Lambda_\downarrow$ is also $\vartheta-\theta$ equivariant. Therefore

$$\mathcal{H}=\left[h^1_{11}-\widetilde{h^1_{11}},\ldots,h^1_{1n_1}-\widetilde{h^1_{1n_1}},\ldots,h^N_{c_Nn_N}-\widetilde{h^N_{c_Nn_N}}\right]^T\in\mathbb{K}[\mathrm{x}]^\theta_\vartheta.$$

The set $\mathcal{H}$ of its component is thus a symmetry adapted basis. The correctness and termination of Algorithm 3 follow from the same arguments exposed for Algorithm 2. Note that both Macaulay and Vandermonde matrices split in $\sum_{i=1}^N n_i$ blocks. Assuming that the blocks are equally distributed and thanks to [37, Proposition 5] we can approximate the dimensions of the blocks by $\frac{\mathrm{M}^i(\mathcal{H}^0)}{\mathrm{M}(\mathcal{H}^0)}\approx\frac{\mathrm{W}^{\mathcal{P}^i}_{\mathcal{L}^i}}{\mathrm{W}^{\mathcal{P}}_{\mathcal{L}}}\approx\frac{1}{|G|}$. Therefore depending on the size of G the dimensions of the matrices to deal with in Algorithm 3 can be considerably reduced.

Algorithm 3

Input: - $\mathcal{L}$ a s.a.b of Λ ($r=|\mathcal{L}|=\dim(\Lambda)$, $r_i=\left|\mathcal{L}^{i,1}\right|$)
- $\mathcal{P}$ an orthonormal graded s.a.b of $\mathbb{K}[\mathrm{x}]_{\le r}$
- $\mathcal{M}_i$ a graded s.a.b of $\mathbb{K}[\mathrm{x}]^{n_i}_{\le r}$, $1\le i\le N$

Output: - $\mathcal{H}$ an orthogonal equivariant H-basis for $\mathcal{I}:=\ker\Lambda$
- $\mathcal{P}_\Lambda$ a s.a.b of the least interpolation space for Λ.

$\mathcal{H}^0\leftarrow\{\}$, $\mathcal{P}_\Lambda\leftarrow\{\}$
$d\leftarrow 0$
$\mathcal{L}_{\le 0}\leftarrow\{\}$, $\mathcal{L}_{\ge 0}\leftarrow\mathcal{L}$
while $\mathcal{L}_{\ge d}\neq\{\}$ **do**
  for $i=1$ **to** N such that $\mathcal{L}^{i,1}_{\ge d}\neq\emptyset$ **do**
    $\mathrm{Q}\cdot\begin{bmatrix}\mathrm{R}_{d,i}\\0\end{bmatrix}=\mathrm{W}^{\mathcal{P}^{i,1}_d}_{\mathcal{L}^{i,1}_{\ge d}}$ ▷ QR-decomposition of $\mathrm{W}^{\mathcal{P}^{i,1}_d}_{\mathcal{L}^{i,1}_{\ge d}}$
    $\left[\mathcal{L}^{i,1}_d\mid\mathcal{L}^{i,1}_{\ge d+1}\right]\leftarrow\mathcal{L}^{i,1}_{\ge d}\cdot\mathrm{Q}^T$
    $\mathcal{L}^{i,1}_{\le d+1}\leftarrow\mathcal{L}^{i,1}_{\le d}\cup\mathcal{L}^{i,1}_d$
    $[\mathrm{Q}_1\mid\mathrm{Q}_2]\cdot\mathrm{R}=\begin{bmatrix}\mathrm{R}^T_{d,i} & \mathrm{M}^i_d(\mathcal{H}^0)\end{bmatrix}$
    for $\alpha=1$ **to** n_i **do**
      $\mathcal{P}^i_\Lambda\leftarrow\mathcal{P}^i_\Lambda\cup\mathcal{P}^{i,\alpha}_d\cdot\mathrm{R}^T_{d,i}$
      $\mathcal{H}^0_i\leftarrow\mathcal{H}^0_i\bigcup\mathcal{P}^{i,\alpha}_d\cdot\mathrm{Q}_2$
  $d\leftarrow d+1$
for $i=1$ **to** N **do**
  for all $p\in\mathcal{H}^0_i$ **do**
    $\mathcal{H}\leftarrow\mathcal{H}\bigcup\left\{p-\mathcal{P}^{i,1}_\Lambda\left(\mathrm{W}^{\mathcal{P}^{i,1}_\Lambda}_{\mathcal{L}^{i,1}_{\le d}}\right)^{-1}\left(\mathcal{L}^{i,1}_{\le d}\right)^T\right\}$
return $(\mathcal{H},\mathcal{P}_\Lambda)$

Example 6.2. The subgroup of the orthogonal group $\mathbb{R}^3$ that leaves the regular the cube invariant is commonly called O_h. It has order 48 and 10 inequivalent irreducible representations whose dimensions are $(1, 1, 1, 1, 2, 2, 3, 3, 3, 3)$. Consider $\Xi \subset \mathbb{R}^3$ the invariant set of 26 points illustrated on Figure 1a. They are grouped in three orbits O_1, O_2 and O_3 of O_h. The points in O_1 are the vertices of a cube with the center at the origin and with edge length $\sqrt{3}$. The points in O_2 and in O_3 are the centers of the faces and middle of the edges of a cube with the center at the origin and edge length 1. Consider $\Lambda = \text{span}\left(\left\{e_\xi \mid \xi \in \Xi\right\} \bigcup \left\{e_\xi \circ D_{\vec{\xi}} \mid \xi \in O_2\right\}\right)$. Λ is an invariant subspace and $\mathcal{I} = \bigcap_{\lambda\in\Lambda} \ker \lambda$ is an ideal. An orthogonal equivariant H-basis $\mathcal{H}$ of $\mathcal{I}$ is given by

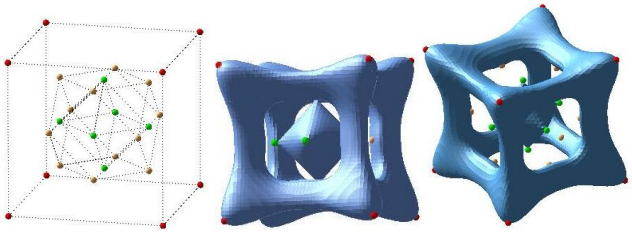
$$h_1^1 = \left[-\tfrac{36}{37} + \tfrac{109}{37}\left(x^2+y^2+z^2\right) - \tfrac{110}{37}\left(x^4+y^4+z^4\right) - \tfrac{36}{37}\left(x^2y^2+x^2z^2+y^2z^2\right) + x^6+y^6+z^6\right]$$

$$h_1^7 = \left[yz^3 - y^3z,\, xz^3 - x^3z,\, xy^3 - x^3y\right]$$

$$h_2^7 = \left[x\left(y^4 - y^2 + z^4 - z^2 - 3(x^4 - 2x^2 + 1)\right),\, y\left(z^4 - z^2 + x^4 - x^2 - 3(y^4 - 2y^2 + 1)\right),\right.$$
$$\left. z\left(\tfrac{4}{3}x^2y^2 - 3(z^4 - 2z^2 + 1)\right)\right]$$

$$h_1^9 = \left[yz\left(-2 - \tfrac{4}{3}x^2 + y^2 + z^2\right),\, xz\left(-2 + x^2 - \tfrac{4}{3}y^2 + z^2\right),\, xy\left(-2 + x^2 + y^2 - \tfrac{4}{3}z^2\right)\right]$$

From the structure of $\mathcal{H}$ it follows that h_{11}^1 is the minimal degree invariant polynomial (up to a constant multiple) of $\mathcal{I}$. In Figure 1b we show the zero surface of h_{11}^1 which is O_h invariant.



(a) Points in Ξ divided in orbits **(b) Variety of the minimal degree invariant polynomial h_{11}^1 of $\mathcal{I}$**

Figure 1: Lowest degree invariant algebraic surface through an invariant set of the points Ξ

Example 6.3. Lets consider the cyclic group C_3, and its action over R^3. It has order 3 and 3 inequivalent irreducible representations of dimension 1, one absolutely irreducible representation and a pair of conjugate irreducible representations of complex type. We analyze the cyclic n–th roots system [3], which has been widely used as a benchmark. The cyclic 3–th roots system is defined by:

$$C(3) : x + y + z, \quad xy + yz + zx, \quad xyz - 1.$$

The associated ideal $\mathcal{I} = \langle C(3)\rangle$ of $C(3)$ is invariant under C_3. The reduced Gröbner basis $\mathcal{G}$ of $\mathcal{I}$ w.r.t the graded reverse lexicographic order and its corresponding normal set $\mathcal{N}$ are given by $\mathcal{G} := \left\{x + y + z, y^2 + yz + z^2, z^3 - 1\right\}$ and $\mathcal{N} := \left\{1, z, y, z^2, yz, yz^2\right\}$. Applying Algorithm 3 to the linear forms given by the coefficients of the normal forms w.r.t $\mathcal{N}$, we obtain a symmetry adapted H-basis $\mathcal{H} = \left\{x + y + z, x^2 + y^2 + z^2, x^3 + y^3 + z^3 - 3\right\}$ as well as a symmetry preserving and robust representation of the quotient $\mathcal{P} = \{1, (y - z)(x - z)(x - y), x - z, y - z, (x - y)(x - 2z + y), (y - z)(2x - y - z)\}$.

REFERENCES

[1] J. Berthomieu, B. Boyer, and J.-C. Faugère. 2017. Linear algebra for computing Gröbner bases of linear recursive multidimensional sequences. *Journal of Symbolic Computation* 83 (2017), 36 – 67.
[2] G. Birkhoff. 1979. The algebra of multivariate interpolation. *Constructive approaches to mathematical models* (1979), 345–363.
[3] G. Björck. 1990. Functions of modulus 1 on Z n whose Fourier transforms have constant modulus, and "cyclic n-roots". In *Recent Advances in Fourier Analysis and its Applications*. Springer, 131–140.
[4] M. Collowald and E. Hubert. 2015. A moment matrix approach to computing symmetric cubatures. (2015). https://hal.inria.fr/hal-01188290.
[5] C. De Boor. 1994. Gauss elimination by segments and multivariate polynomial interpolation. In *Approximation and Computation: A Festschrift in Honor of Walter Gautschi*. Springer, 1–22.
[6] C. De Boor and A. Ron. 1990. On multivariate polynomial interpolation. *Constructive Approximation* 6, 3 (1990).
[7] C. De Boor and A. Ron. 1992. The least solution for the polynomial interpolation problem. *Mathematische Zeitschrift* 210, 1 (1992).
[8] C. Fassino and H.M. Möller. 2016. Multivariate polynomial interpolation with perturbed data. *Numerical Algorithms* 71, 2 (2016), 273–292.
[9] A. Fässler and E. Stiefel. 1992. *Group theoretical methods and their applications*.
[10] J.-C Faugere, P. Gianni, D. Lazard, and T. Mora. 1993. Efficient computation of zero-dimensional Gröbner bases by change of ordering. *Journal of Symbolic Computation* 16, 4 (1993), 329–344.
[11] J.-C. Faugère and C. Mou. 2017. Sparse FGLM algorithms. *Journal of Symbolic Computation* 80, 3 (2017), 538 – 569.
[12] J.-C. Faugere and J. Svartz. 2013. Grobner bases of ideals invariant under a commutative group: the non-modular case. In *Proc. ISSAC 2013*. ACM, 347–354.
[13] K. Gatermann. 1990. Symbolic solution of polynomial equation systems with symmetry. In *ISSAC'90 Tokyo, Japan*. ACM-Press, 112–119.
[14] K. Gatermann. 1992. Linear representations of finite groups and the ideal theoretical construction of G-invariant cubature formulas. In *Numerical integration (Bergen, 1991)*. NATO Adv. Sci. Inst. Ser. C Math. Phys. Sci., Vol. 357. Kluwer Acad. Publ., Dordrecht, 25–35.
[15] K. Gatermann. 2000. *Computer algebra methods for equivariant dynamical systems*. Lecture Notes in Mathematics, Vol. 1728. Springer-Verlag, Berlin.
[16] K. Gatermann and F. Guyard. 1999. Gröbner bases, invariant theory and equivariant dynamics. *J. Symbolic Comput.* 28, 1-2 (1999), 275–302.
[17] K. Gatermann and P. A. Parrilo. 2004. Symmetry groups, semidefinite programs, and sums of squares. *J. Pure Appl. Algebra* 192, 1-3 (2004), 95–128.
[18] G. Golub and C. Van Loan. 1996. *Matrix Computations (3rd Ed.)*.
[19] E. Hubert and G. Labahn. 2012. Rational invariants of scalings from Hermite normal forms. In *Proc. ISSAC 2012*. ACM, 219–226.
[20] E. Hubert and G. Labahn. 2013. Scaling invariants and symmetry reduction of dynamical systems. *Found. Comput. Math.* 13, 4 (2013), 479–516.
[21] E. Hubert and G. Labahn. 2016. Computation of the Invariants of Finite Abelian Groups. *Mathematics of Computations* 85, 302 (2016), 3029–3050.
[22] M. Javanbakht and T. Sauer. 2019. Numerical computation of H-bases. *BIT Numerical Mathematics* 59, 2 (2019), 417–442.
[23] T. Krick, A. Szanto, and M. Valdettaro. 2017. Symmetric interpolation, Exchange Lemma and Sylvester sums. *Comm. Algebra* 45, 8 (2017), 3231–3250.
[24] F.S. Macaulay. 1916. The algebraic theory of modular systems. *Cambridge Tracts in Mathematics and Mathematical Physics* 19 (1916).
[25] M.G. Marinari, H.M. Möller, and T. Mora. 1991. Gröbner bases of ideals given by dual bases. In *ISSAC'91*. ACM, 55–63.
[26] H.M. Möller and B. Buchberger. 1982. The construction of multivariate polynomials with preassigned zeros. In *European Computer Algebra Conference*.
[27] H.M. Möller and T. Sauer. 2000. H-bases for polynomial interpolation and system solving. *Advances in Computational Mathematics* 12, 4 (2000), 335–362.
[28] B. Mourrain. 2017. Fast algorithm for border bases of Artinian Gorenstein algebras. In *ISSAC'17 Kaiserslautern, Germany*. ACM Press, 333–340.
[29] G. Pistone, E. Riccomagno, and H.P. Wynn. 2000. *Algebraic statistics: Computational commutative algebra in statistics*. Chapman and Hall/CRC.
[30] G. Pistone and H.P. Wynn. 1996. Generalised confounding with Gröbner bases. *Biometrika* 83, 3 (1996), 653–666.
[31] C. Riener and M. Safey El Din. 2018. Real root finding for equivariant semi-algebraic systems. In *Proc. ISSAC 2018*. ACM, 335–342.
[32] C. Riener, T. Theobald, L. J. Andrén, and J. B. Lasserre. 2013. Exploiting symmetries in SDP-relaxations for polynomial optimization. *Math. Oper. Res.* 38, 1 (2013).
[33] E. Rodriguez Bazan and E. Hubert. 2019. Symmetry Preserving Interpolation. In *ISSAC'19*. https://hal.inria.fr/hal-01994016
[34] T. Sauer. 2001. Gröbner bases, H–bases and interpolation. *Trans. Amer. Math. Soc.* 353, 6 (2001), 2293–2308.
[35] T. Sauer. 2017. Prony's method in several variables. *Numer. Math.* 136, 2 (2017).
[36] T. Sauer. 2018. Prony's method in several variables: symbolic solutions by universal interpolation. *J. Symbolic Comput.* 84 (2018), 95–112.
[37] J. P. Serre. 1977. *Linear representations of finite groups*. Springer.
[38] J. Verschelde and K. Gatermann. 1995. Symmetric Newton polytopes for solving sparse polynomial systems. *Adv. in Appl. Math.* 16, 1 (1995), 95–127.

Generalizing The Davenport-Mahler-Mignotte Bound – The Weighted Case

Vikram Sharma
vikram@imsc.res.in
The Institute of Mathematical Sciences, HBNI
Chennai, India

ABSTRACT

Root separation bounds play an important role as a complexity measure in understanding the behaviour of various algorithms in computational algebra, e.g., root isolation algorithms. A classic result in the univariate setting is the Davenport-Mahler-Mignotte (DMM) bound. One way to state the bound is to consider a directed acyclic graph (V, E) on a subset of roots of a degree d polynomial $f(z) \in \mathbb{C}[z]$, where the edges point from a root of smaller absolute value to one of larger absolute, and the in-degrees of all vertices is at most one. Then the DMM bound is an amortized lower bound on the following product: $\prod_{(\alpha,\beta)\in E} |\alpha - \beta|$. However, the lower bound involves the discriminant of the polynomial f, and becomes trivial if the polynomial is not square-free. This was resolved by Eigenwillig, 2008, by using a suitable subdiscriminant instead of the discriminant. Escorcielo-Perrucci, 2016, further dropped the in-degree constraint on the graph by using the theory of finite differences. Emiris et al., 2019, have generalized their result to handle the case where the exponent of the term $|\alpha - \beta|$ in the product is at most the multiplicity of either of the roots. In this paper, we generalize these results by allowing arbitrary positive integer weights on the edges of the graph, i.e., for a weight function $w : E \to \mathbb{Z}_{>0}$, we derive an amortized lower bound on $\prod_{(\alpha,\beta)\in E} |\alpha - \beta|^{w(\alpha,\beta)}$. Such a product occurs in the complexity estimates of some recent algorithms for root clustering (e.g., Becker et al., 2016), where the weights are usually some function of the multiplicity of the roots. Because of its amortized nature, our bound is arguably better than the bounds obtained by manipulating existing results to accommodate the weights.

KEYWORDS

Root separation bounds, confluent Vandermonde matrix, finite differences, sub-discriminants, nuclear norm.

ACM Reference Format:
Vikram Sharma. 2020. Generalizing The Davenport-Mahler-Mignotte Bound – The Weighted Case. In *International Symposium on Symbolic and Algebraic Computation (ISSAC '20), July 20–23, 2020, Kalamata, Greece.* ACM, New York, NY, USA, 8 pages. https://doi.org/10.1145/3373207.3404016

ISSAC '20, July 20–23, 2020, Kalamata, Greece

ACM ISBN 978-1-4503-7100-1/20/07…$15.00
https://doi.org/10.1145/3373207.3404016

1 INTRODUCTION

Given a *monic* univariate polynomial $f(z) \in \mathbb{C}[z]$, of degree d with roots $\alpha_1, \ldots, \alpha_d$, not all distinct, a **root separation bound** is a lower bound on the smallest distance $\mathrm{sep}(f)$ between any distinct pair of roots of f. A classic result [11] states that

$$\mathrm{sep}(f) > d^{-(d+2)/2}\Delta(f)^{1/2}M(f)^{1-d},$$

where

$$\Delta(f) := \prod_{i<j}(\alpha_i - \alpha_j)^2$$

is the **discriminant** of f, and

$$M(f) := \prod_{i=1}^{d} \max\{1, |\alpha_i|\} \tag{1}$$

is the **Mahler measure** of f.

The parameter $\mathrm{sep}(f)$ naturally occurs in the complexity analysis of many algorithms; examples are the (real or complex) root isolation algorithms ([13], [3], [5], [2]). However, most of these algorithms need a lower bound on the product of certain pairs of roots and not just the worst case separation. To capture these pairs, we consider a simple (i.e., no loops and multiple edges) undirected graph $G = (V, E)$, whose vertices are a subset of the distinct roots of f. Then we want a lower bound on $\prod_{(\alpha_i,\alpha_j)\in E} |\alpha_i - \alpha_j|$. One straightforward lower bound is $\mathrm{sep}(f)^{|E|}$, but Davenport [3] used the amortized nature of the Mahler measure to derive a lower bound for real roots that essentially matches the lower bound on $\mathrm{sep}(f)$ given above; the argument was later modified by Mignotte to complex roots [12]. A consequence of these results is a straightforward improvement in the complexity bounds on the running time of algorithms for root isolation algorithms by a multiplicative factor of the degree.

Both these lower bounds, nevertheless, rely on the discriminant $\Delta(f)$ and are trivial when the polynomial is not square-free, i.e., it has multiple roots. A remedy is to work with the square-free part $\widehat{f}$ of f, but this again blows the bound by exponential factors because of the growth in the coefficients of $\widehat{f}$ as compared to f. An alternative presented by Eigenwillig [4] used the $(d - r)$-th subdiscriminant of f instead of the the discriminant, where r is the number of distinct roots of f. However, there are some constraints on the graph G for the bound to be applicable, namely, in the directed acyclic graph obtained by directing the edges of G from a root of smaller absolute value to one of larger absolute value, the in-degree of all the vertices is at most one. Escorcielo-Perrucci [7] dropped this in-degree constraint by using the theory of finite differences. Despite this, their result gives weaker bounds on products

of the form

$$\prod_{(\alpha_i,\alpha_j)\in E} |\alpha_i - \alpha_j|^{w(\alpha_i,\alpha_j)}, \tag{2}$$

where $w : E \longrightarrow \mathbb{N}$ is a **weight function** that assigns a positive integer to all the edges.[1] In the special case where the weight function is such that the weight of an edge is bounded by the multiplicity of one of its vertices, [10] and [6] have derived lower bounds when the coefficients of f are real and complex numbers, respectively. To state their bound, let f have r distinct roots $\alpha_1, \ldots, \alpha_r$ with multiplicities $m_1, \ldots, m_r$, respectively, $\widehat{f}$ denote the square-free part of f, and for a root α_i let Δ_i denote the distance to the nearest distinct root. Then the bound in [6] is the following: If $K \subseteq [r]$ and $w_i \in \mathbb{N}$ is such that $w_i \leq m_i$, for $i \in K$, then

$$\prod_{i\in K} \Delta_i^{w_i} \geq 2^{-d(r+2)}(\|f\|_\infty \|\widehat{f}\|_\infty)^{-d} M(f)^{1-r} |\mathrm{res}(f, \widehat{f}')|, \tag{3}$$

here $\|\cdot\|_\infty$ is the maximum absolute value over the coefficient sequence of the polynomial, and $\mathrm{res}(\cdot,\cdot)$ is the univariate resultant. These bounds, though useful, fail to provide amortized lower bounds when the w_i's exceed the multiplicity. Such a scenario, for instance, occurs in the complexity analysis of some recent root clustering algorithms [1, 2], where the following product occurs, for some subsets $K_i \subseteq [r]$:

$$\prod_{i\in K} \Delta_i^{\sum_{j\in K_i} m_j}.$$

One way to derive a lower bound on this product is to exponentiate the left-hand side of (3) to the degree d (since the sum of the multiplicities over K_i is bounded by d), move the extraneous factors to the denominator in the right-hand side, and upper bound these to get a lower bound on the desired product. But, just as was the case with $\mathrm{sep}(f)^{|E|}$ earlier, such an approach loses the amortization property and gives exponentially worse bounds.

In this paper, we derive a lower bound on the product in (2) for arbitrary weight functions. The restrictions on the weights in the earlier approaches was an outcome of the choice of the symmetric function (either the discriminant, sub-discriminant or the resultant). We instead choose a symmetric function based on the weights and try to optimize over all valid choices of the function. This is done by constructing a confluent Vandermonde matrix to get the desired weight structure in the exponents. The choice of the confluent Vandermonde is especially helpful when the weights are skewed in distribution, because this means we can pick a different multiplicity structure on the roots and obtain better bounds. The spectral structure of the weighted adjacency matrix $A_w := [w_{i,j}]_{i,j=1,\ldots,r}$ plays an important role in the choice of the multiplicity structure for constructing the confluent Vandermonde matrix. For ease of comprehension, we state our result when f is an integer polynomial (since then the absolute value of the non-zero symmetric function is at least one, which is how the bounds are used in practice) and is also monic (otherwise divide $M(f)$ by the absolute value of the leading coefficient). Let $\|A_w\|_\star$ denote the **nuclear norm** of A_w, i.e., the sum of its singular values, $n := r\left\lceil \sqrt{\|A_w\|_\star} \right\rceil$, and $w(E)$ be the sum of the weights over the edges of G. Then we show that

$$\prod_{(\alpha_i,\alpha_j)\in E} |\alpha_i - \alpha_j|^{w(\alpha_i,\alpha_j)} > M(f)^{-2r\|A_w\|_\star} \left(\frac{n}{\sqrt{3}}\right)^{-\frac{3r\|A_w\|_\star}{2} - w(E)} n^{-n/2}. \tag{4}$$

The bound is amortized because the exponent of the Mahler measure does not contain $w(E)$, which would be the case if we try to derive the lower bound by modifying the earlier results (see (11) below).

In the next section, we give the requisite details and properties of the confluent Vandermonde matrix; Section 3 contains the statement of our main result Theorem 3.2 and its comparison with a modification of an existing bound; Section 4 contains a proof of the main result, and in Section 4.1 we specialize it to obtain the form given above in (4).

2 CONFLUENT VANDERMONDE

Consider the column vector

$$v(x)^t := \begin{bmatrix} 1 & x & x^2 & \cdots & x^n \end{bmatrix}.$$

Define the vector obtained by differentiating each entry in the column above i times and dividing by $i!$, ie.,

$$v_i(x)^t := \begin{bmatrix} \binom{0}{i}x^{-i} & \binom{1}{i}x^{1-i} & \binom{2}{i}x^{2-i} & \cdots & \binom{n-1}{i}x^{n-1-i} \end{bmatrix}, \tag{5}$$

with the natural convention that $\binom{j}{i} = 0$ if $j < i$. Let

$$\boldsymbol{\beta} := (\beta_1, \ldots, \beta_r) \in \mathbb{C}^r$$

be an r-dimensional vector of complex numbers,

$$\boldsymbol{\mu} := (\mu_1, \ldots, \mu_r) \in \mathbb{N}^r$$

be a sequence of positive integers, and $n := \sum_i \mu_i$. Then the **confluent Vandermonde matrix** $V(\boldsymbol{\beta}; \boldsymbol{\mu})$ is the $n \times n$ matrix with columns $(v_j(\beta_i))$, where $1 \leq i \leq r$ and $0 \leq j \leq \mu_i - 1$. We will also use the notation $V(\beta_1, \ldots, \beta_r; \mu_1, \ldots, \mu_r)$ when we want to emphasize the β_i's and μ_i's. We illustrate it below for $r = 3$ and $\mu_1 = 2, \mu_2 = 3$.

$$V(\boldsymbol{\beta}, \boldsymbol{r}) = \begin{bmatrix} 1 & 0 & 1 & 0 & 0 \\ \beta_1 & 1 & \beta_2 & 1 & 0 \\ \beta_1^2 & \binom{2}{1}\beta_1 & \beta_2^2 & 2\beta_2 & 1 \\ \beta_1^3 & \binom{3}{1}\beta_1^2 & \beta_2^3 & 3\beta_2^2 & \binom{3}{2}\beta_2 \\ \beta_1^4 & \binom{4}{1}\beta_1^3 & \beta_2^4 & 4\beta_2^3 & \binom{4}{2}\beta_2^2 \end{bmatrix}.$$

The **block, $B(\beta_i)$, corresponding to a** β_i, is the set of columns $(v_j(\beta_i))$, for $j = 0, \ldots, \mu_i - 1$. If all the μ_i's are one then we obtain the standard Vandermonde matrix denoted as $V(\boldsymbol{\beta})$. A key observation in understanding the determinant of the matrix above is to consider the matrix obtained by replacing the last column $v_{\mu_i - 1}(\beta_i)$, corresponding to some β_i with $\mu_i > 1$, with the column $v(y)$, for some variable y, which gives us the matrix

$$V(y) := V(\beta_1, \ldots, \beta_i, y, \beta_{i+1}, \ldots, \beta_r; \mu_1, \ldots, \mu_i - 1, 1, \mu_{i+1}, \ldots, \mu_r). \tag{6}$$

Let $\mathcal{V}(y) := \det(V(y))$. By expanding along the column corresponding to y, we can express $\mathcal{V}(y)$ as a polynomial in y with degree at most $n - 1$. If we differentiate this polynomial $(\mu_i - 1)$ times, divide by $(\mu_i - 1)!$ and substitute $y = \beta_i$, then we will recover the

[1] Throughout, we use $\mathbb{N}$ to denote the set of positive integers and $\mathbb{Z}_{\geq 0}$ the set of non-negative integers.

determinant of $V(\boldsymbol{\beta};\boldsymbol{\mu})$ expanded along the last column of the block $B(\beta_i)$. More precisely,

$$\det(V(\boldsymbol{\beta};\boldsymbol{\mu})) = \left.\frac{\mathcal{V}^{(\mu_i-1)}(y)}{(\mu_i-1)!}\right|_{y=\beta_i}. \tag{7}$$

This result is crucial in deriving the following explicit form for the determinant [9].

Proposition 2.1. *The determinant of the confluent Vandermonde matrix satisfies*

$$\det(V(\boldsymbol{\beta};\boldsymbol{\mu})) = \prod_{1\le i<j\le r}(\beta_j-\beta_i)^{\mu_i\mu_j}.$$

3 THE DAVENPORT-MAHLER-MIGNOTTE BOUND

The following variant of the bound appears in [7].

Proposition 3.1. *Let $\boldsymbol{\alpha}$:=$(\alpha_1,\ldots,\alpha_r)$ be a sequence of distinct complex numbers, and*

$$M(\boldsymbol{\alpha}) := \prod_{i=1}^{r}\max\{1,|\alpha_i|\}. \tag{8}$$

If $G(V,E)$ is an undirected simple graph (i.e., with no multi edges and self-loops) with vertices $V\subseteq\{\alpha_1,\ldots,\alpha_r\}$, then

$$\prod_{(\alpha_i,\alpha_j)\in E}|\alpha_i-\alpha_j| \ge |\det(V(\boldsymbol{\alpha}))|M(\boldsymbol{\alpha})^{-(r-1)}\left(\frac{r}{\sqrt{3}}\right)^{-|E|} r^{-r/2}.$$

Remark: The result in [7] actually uses the sub-discriminant. Given a degree d polynomial

$$f(z)=\prod_{i=1}^{r}(z-\alpha_i)^{m_i},$$

with distinct roots α_i of multiplicity m_i, for $1\le i\le r$, the $(d-r)$ discriminant of f is given by

$$\mathrm{sDisc}_{d-r}(f) := \det(V(\boldsymbol{\alpha}))^2\prod_{j=1}^{r}m_i. \tag{9}$$

Taking absolute values and substituting the expression for the absolute value of the determinant into Proposition 3.1 we get

$$\prod_{(\alpha_i,\alpha_j)\in E}|\alpha_i-\alpha_j| \ge |\mathrm{sDisc}_{d-r}(f)|^{1/2}M(f)^{-(r-1)} \times\left(\frac{r}{\sqrt{3}}\right)^{-|E|}\frac{r^{-r/2}}{\prod_{i=1}^{r}\sqrt{m_i}}. \tag{10}$$

Escorcielo-Perrucci [7] then use the following upper bound by Eigenwillig [4] to derive the final form of their result: If $m_1,\ldots,m_r$ are natural numbers such that $\sum_{i=1}^{r}m_i=d$ then

$$\prod_{i=1}^{r}\sqrt{m_i}\le 3^{\min\{d,2(d-r)\}/6}.$$

Instead, if we use the AM-GM inequality then we get a sharper bound, namely

$$\prod_{i=1}^{r}\sqrt{m_i}\le\left(\frac{d}{r}\right)^{r/2}.$$

Substituting this in (10), we get the following improvement over [7]:

$$\prod_{(\alpha_i,\alpha_j)\in E}|\alpha_i-\alpha_j| \ge |\mathrm{sDisc}_{d-r}(f)|^{1/2}M(f)^{-(r-1)}\left(\frac{r}{\sqrt{3}}\right)^{-|E|}d^{-r/2}.$$

We will generalize Proposition 3.1 above to account for non-zero integer weights on the edges, i.e., a lower bound on the product given in (2). To illustrate the advantage of our approach, we first give the details of a lower bound obtained by a straightforward modification of Proposition 3.1.

Let $w_{\max}$ be the largest weight over all the edges in G. Then we can raise the bound in the Proposition 3.1 to this weight and move the extraneous factor to the right-hand side, and replace them with an upper bound. For any edge $(\alpha_i,\alpha_j)\in E$, we have

$$|\alpha_i-\alpha_j|^{w_{\max}-w(\alpha_i,\alpha_j)}\le(2M(f))^{w_{\max}}.$$

Therefore, we obtain the following lower bound as a modification of Proposition 3.1, which we will use to compare with the bound derived in this paper:

$$\prod_{(\alpha_i,\alpha_j)\in E}|\alpha_i-\alpha_j|^{w(\alpha_i,\alpha_j)} \ge |\det(V(\boldsymbol{\alpha}))|^{w_{\max}}M(\boldsymbol{\alpha})^{-((r-1)w_{\max}+|E|w_{\max})}\cdot 2^{-(|E|w_{\max})}\left(\frac{r}{\sqrt{3}}\right)^{-|E|w_{\max}}r^{-(rw_{\max})/2}. \tag{11}$$

In comparison, we obtain the following generalization:

Theorem 3.2. *Let $\alpha_1,\ldots,\alpha_r\in\mathbb{C}$ be distinct complex numbers. Let $G(V,E)$ be an undirected graph whose vertices V is a subset of $\{\alpha_1,\ldots,\alpha_r\}$, with an associated a weight function $w: E\to\mathbb{N}$. Denote by*

$$A_w=[w(\alpha_i,\alpha_j)]_{i,j=1,\ldots,r}$$

the associated weighted adjacency matrix. To every vertex $\alpha_i\in V$, we assign a potential $\mu_i\in\mathbb{N}$ such that for every edge $(\alpha_i,\alpha_j)\in E$, we have $w(\alpha_i,\alpha_j)\le\mu_i\mu_j$. Define $\boldsymbol{\mu}$ as the column-vector of these potentials, n:=$\sum_{i=1}^{r}\mu_i$, $M(\boldsymbol{\alpha})$ be as in (8), and $w(E)$ as the sum of the weights of the edges in the graph G, i.e.,

$$w(E) := \sum_{(\alpha_i,\alpha_j)\in E}w(\alpha_i,\alpha_j). \tag{12}$$

Then

$$\prod_{(\alpha_i,\alpha_j)\in E}|\alpha_i-\alpha_j|^{w(\alpha_i,\alpha_j)} > |\det(V(\boldsymbol{\alpha};\boldsymbol{\mu}))|\,M(\boldsymbol{\alpha})^{-\|\boldsymbol{\mu}\boldsymbol{\mu}^t-A_w\|_\infty}\left(\frac{n}{\sqrt{3}}\right)^{-\sum_i\binom{\mu_i}{2}-w(E)}n^{-n/2}, \tag{13}$$

where ∞-norm of a matrix is the maximum one-norm over all the rows of the matrix.

Remarks:

(1) Since we are dealing with symmetric matrices, we can replace the ∞-norm with the induced 1-norm, which is the maximum over the sum of the columns.
(2) If all the weights are one, then we can take μ_i's as 1, and obtain Proposition 3.1 as a corollary.

(3) There is an interesting trade-off between the absolute values of the exponent of $M(\boldsymbol{\alpha})$ and $n/\sqrt{3}$, namely, as the number of edges in G increases the former decreases whereas the latter increases.

In order to compare (10) and (13), we make three assumptions:

(i) G is connected, so $|E| \geq r-1$,
(ii) $\mu_i = \sqrt{w_{\max}}$, for all $i = 1, \ldots, r$, and
(iii) f is an integer polynomial.

From the last assumption, it follows that both $|\det(V(\boldsymbol{\alpha}))|$ and $|\det(V(\boldsymbol{\alpha};\boldsymbol{\mu}))|$ are at least one, and that is how we often use them in applications. The second assumption implies that $n = r\sqrt{w_{\max}}$. We now compare three analogous terms from both the bounds by taking logarithms.

From the assumption of connectivity, it follows that the absolute value of the exponent of $M(\boldsymbol{\alpha})$ in (10) is at least $2(r-1)w_{\max}$, whereas in (13) it is at most $rw_{\max}$. If $r \geq 2$, then it follows that the former is larger than the latter. The difference is because of the amortized property of the bound in (13).

Consider the negation of the logarithm of the term

$$n^{-\sum_i \binom{\mu_i}{2} - w(E)}$$

in (13). This is equal to

$$\left(\sum_i \binom{\mu_i}{2} + w(E)\right) \log n \leq \left(\sum_i \binom{\mu_i}{2} + |E| w_{\max}\right) \log(r\sqrt{w_{\max}}).$$

Since $\binom{\mu_i}{2} \leq \mu_i^2/2$, it follows that $\sum_i \binom{\mu_i}{2} \leq rw_{\max}/2$. Therefore, the right-hand side above is upper bounded by

$$2|E| w_{\max} \log(r\sqrt{w_{\max}})$$

which is somewhat larger than $(-\log r^{-|E|w_{\max}})$, the corresponding term in (10). It must be remarked, nevertheless, that the choice in the second assumption is not the best (see Section 4.1) and is only used for illustration at this point.

The negation of the logarithm of $n^{-n/2}$ in (13) is

$$r\sqrt{w_{\max}} \log(r\sqrt{w_{\max}}),$$

which is better than the corresponding term in (10), namely,

$$(rw_{\max}) \log r,$$

for sufficiently large $w_{\max}$.

3.1 Some Results from the Theory of Finite Differences

Let $f : \mathbb{C} \to \mathbb{C}$ be a function and $y_1, \ldots, y_n$ be n nodes. Then the divided difference of f on these n nodes is given by

$$f[y_1, \ldots, y_n] := \sum_{k=1}^{n} \prod_{\ell=1;\ell\neq k}^{n} \frac{1}{(y_k - y_\ell)} f(y_k). \tag{14}$$

If $f(z) := z^m$, for some $m \in \mathbb{Z}_{\geq 0}$, then we have the following closed form:

$$f[y_1, \ldots, y_n] = \begin{cases} \displaystyle\sum_{\substack{(t_1,\ldots,t_n)\in\mathbb{Z}^n_{\geq 0} \\ \sum_{i=1}^n t_i = m-n+1}} \prod_{j=1}^n y_j^{t_j} & \text{if } n \leq m+1 \\ 0 & \text{if } n > m+1. \end{cases} \tag{15}$$

Given $i_1, \ldots, i_n \in \mathbb{Z}_{\geq 0}$, denote by

$$f^{(i_1,\ldots,i_n)}[y_1, \ldots, y_n] := \frac{1}{i_1!}\frac{\partial^{i_1}}{\partial y_1^{i_1}} \cdots \frac{1}{i_n!}\frac{\partial^{i_n}}{\partial y_n^{i_n}} f[y_1, \ldots, y_n]. \tag{16}$$

Then the following claim is straightforward to show:

Lemma 3.3. *Given $i_1, \ldots, i_n \in \mathbb{Z}_{\geq 0}$, the quantity*

$$f^{(i_1,\ldots,i_n)}[y_1, \ldots, y_n]$$

is a linear combination of $f^{(k_j)}(y_j)$, where $j = 1, \ldots, n$ and $k_j = 0, \ldots, i_j$. Moreover, the coefficient of $f^{(i_j)}(y_j)$ in this linear combination is

$$\frac{1}{i_j!} \prod_{\ell=1:\ell\neq j}^{n} \frac{1}{(y_j - y_\ell)^{i_\ell+1}}.$$

Proof. For simplicity, we only argue for i_1; the argument is similar for other cases. Consider the effect of $\frac{1}{i_1!}\frac{\partial^{i_1}}{\partial y_1^{i_1}}$ on $f[y_1, \ldots, y_n]$. By linearity of the derivative operator, we only need to focus on the term $f(y_1)/\prod_{i\neq 1}(y_1 - y_i)$. From Leibniz's rule applied to this term we get the expression

$$\frac{1}{i_1!} \frac{f^{(i_1)}(y_1)}{\prod_{i\neq 1}(y_1 - y_i)}.$$

The effect of the other partial derivatives $\frac{1}{i_\ell!}\frac{\partial^{i_\ell}}{\partial y_\ell^{i_\ell}}$ is only on the terms in the denominator, which yields the desired expression for the coefficient of $f^{(i_1)}(y_1)$. □

If $f(z) := z^m$, for some $m \in \mathbb{Z}_{\geq 0}$, and $(i_1, \ldots, i_n) \in \mathbb{Z}^n_{\geq 0}$, then as a generalization of (15), we obtain the following

$$f^{(i_1,\ldots,i_n)}[y_1, \ldots, y_n] = \begin{cases} \displaystyle\sum_{\substack{(t_1,\ldots,t_n)\in\mathbb{Z}^n_{\geq 0} \\ \sum_{i=1}^n t_i = m-n+1}} \prod_{j=1}^n \binom{t_j}{i_j} y_j^{t_j - i_j} & \text{if } n \leq m+1 \\ 0 & \text{if } n > m+1 \end{cases} \tag{17}$$

with the natural convention that $\binom{t_j}{i_j} = 0$ if $t_j < i_j$.

4 A PROOF OF THE MAIN RESULT

The idea of the proof is similar to [7]. Given the undirected graph G, we first direct its edges to go from a root of smaller modulus to one of larger modulus; this way we obtain a directed acyclic graph $\mathcal{G}$; the in-degrees of the vertices in $\mathcal{G}$ can be larger than one, which is the case addressed in [7]. We consider the vertices of $\mathcal{G}$ in the reverse order of a topological sort on its vertices, i.e., in the order $(\alpha_1, \ldots, \alpha_r)$, where if (α_i, α_j) is an edge in $\mathcal{G}$ then $j < i$. Let $\mathrm{In}(\alpha_i)$ denote the set of all vertices that have an edge pointing to α_i, d_i be the cardinality of $\mathrm{In}(\alpha_i)$ (i.e., the in-degree of α_i), and

$$V_0 := V(\boldsymbol{\alpha};\boldsymbol{\mu}). \tag{18}$$

At the ith step we will process the block corresponding to α_i in V_{i-1}, where $i \geq 1$, to obtain a matrix V_i. The relation between the two matrices is the following:

$$\det V_{i-1} = \det(V_i) \prod_{\alpha_j \in \mathrm{In}(\alpha_i)} (\alpha_i - \alpha_j)^{w(\alpha_j,\alpha_i)}. \tag{19}$$

The matrix V_i is instead obtained from V_{i-1} in stages by modifying the columns in the block corresponding to α_i, that is, there are two

loops – one over the blocks $B(\alpha_i)$, and an inner loop processing the columns of the block $B(\alpha_i)$. The end result is a matrix V_r such that

$$\det(V_0) = \det(V_r) \prod_{i=1}^{r} \prod_{\alpha_j \in \mathrm{In}(\alpha_i)} (\alpha_i - \alpha_j)^{w(\alpha_j,\alpha_i)}.$$

The final step is to derive an upper bound on $|\det(V_r)|$; this is done by applying Hadamard's inequality, and obtaining upper bounds on the two-norms of the columns of V_r. In what follows, we will use α in place of α_i, μ_α as the size of the block $B(\alpha)$, $k{:=}d_i$, and $V{:=}V_{i-1}$.

Without loss of generality, let us assume that $\beta_1, \ldots, \beta_k$ are the k vertices in $\mathrm{In}(\alpha)$, with respective weights $w_1, \ldots, w_k$. Since we are processing the vertices in reverse topological order, we know that the blocks corresponding to these vertices have not been changed. Let $\mu_1, \ldots, \mu_k$ be the sizes of the blocks $B(\beta_1), \ldots, B(\beta_k)$, respectively. We will replace each column in the block $B(\alpha)$ by a suitable linear combination of the columns in the blocks $B(\alpha)$ and $B(\beta_i)$ for $i = 1, \ldots, k$. The linear combination will be obtained by taking a suitable partial derivative of the form given in (16) and then substituting y_i's appropriately. Ideally, we would have replaced, say the last column in $B(\alpha)$, by the partial derivative obtained by taking full weights, $w_1, \ldots, w_k$. However, there is a slight obstacle, namely, that the derivatives of $f(\beta_i)$, for $i = 1, \ldots, k$, cannot exceed beyond $\mu_i - 1$. To overcome this we assign each edge (β_i, α) with corresponding weight w_i to a column in the block $B(\alpha)$, namely to the $\lceil w_i/\mu_i \rceil$-th column in $B(\alpha)$; since $w_i \leq \mu_i\mu_\alpha$ by assumption on weights, the edge will be assigned to a column in $B(\alpha)$. Let $S_j \subseteq [k]$, for $j = 1, \ldots, \mu_\alpha$, denote the set of all indices assigned to the jth column of $B(\alpha)$, i.e.,

$$S_j{:=} \{i \in [k] : \lceil w_i/\mu_i \rceil = j\}. \tag{20}$$

By assignment it follows that S_j's form a partition of $[k]$. The reason why this assignment works is the following: each column in $B(\alpha)$, along with its preceding columns in $B(\alpha)$ and the blocks $B(\beta_1), \ldots, B(\beta_k)$, can be used to factor out $(\beta_i - \alpha)^{\mu_i}$; therefore, $\lceil w_i/\mu_i \rceil$ columns will be required to get to $(\beta_i - \alpha)^{w_i}$. An illustrative aid for the subsequent proof is provided in Figure 1.

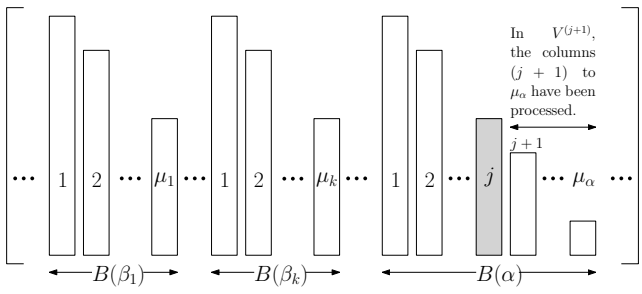


Figure 1: The matrix V_{i-1} and the block $B(\alpha)$ at stage i of the proof. At the jth step in processing V, the columns $(j+1)$ to μ_α of the block $B(\alpha)$ have been processed to obtain $V^{(j+1)}$. In $V^{(j+1)}$, the jth column is processed to obtain $V^{(j)}$.

We will now process the columns of $B(\alpha)$ starting from *the last column to the first in* V; it will help the reader to note that the columns will be counted from 1 to μ_α. Suppose we have already processed the columns of $B(\alpha)$ from μ_α down to $(j+1)$ in V; let $V^{(j+1)}$ be the resulting matrix; initially, define $V^{(\mu_\alpha+1)}{:=}V$. For β_ℓ, $\ell = 1, \ldots, k$, define

$$r_\ell{:=} \begin{cases} \mu_\ell & \text{if } w_\ell \text{ is divisible by } \mu_\ell, \\ (w_\ell \bmod \mu_\ell) & \text{otherwise.} \end{cases} \tag{21}$$

We inductively claim the following relation for $j \leq \mu_\alpha$:

$$\det(V) = \det(V^{(j+1)}) \prod_{\kappa=j+1}^{\mu_\alpha} \prod_{\ell \in S_\kappa} (\beta_\ell - \alpha)^{(\kappa-j-1)\mu_\ell + r_\ell}. \tag{22}$$

The proof is by reverse induction on decreasing values of j; the base case trivially holds for $j = \mu_\alpha$, since the product vanishes and $V = V^{(\mu_\alpha+1)}$ by choice.

To complete the inductive claim (22), we have to obtain the following terms from the jth column in $B(\alpha)$:

(1) the residue terms $(\beta_\ell - \alpha)^{r_\ell}$, for each index $\ell \in S_j$, and
(2) a factor of $(\beta_\ell - \alpha)^{\mu_\ell}$ for all the indices $\ell \in S_\kappa$, where $\kappa > j$.

This is done by taking a suitable partial derivative of the finite difference. Let

$$N_j{:=}|\cup_{\kappa=j}^{\mu_\alpha} S_\kappa|, \tag{23}$$

that is the total number of indices assigned to column j or greater; clearly $N_j \leq k$. We will introduce N_j variables for each of these indices, and a variable y_0 for α. Note that the jth column of the block $B(\alpha)$ in $V^{(j+1)}$ is obtained by substituting $z = \alpha$ in $v_{j-1}(z)$ given in (5). The mth entry of this column, for $m = 1, \ldots, n$, is

$$\binom{m-1}{j-1} z^{m-j} = \frac{(z^{m-1})^{(j-1)}}{(j-1)!}. \tag{24}$$

Define $f_m(z){:=}z^{m-1}$, and consider the finite difference

$$f_m[y_0, y_1, \ldots, y_{N_j}],$$

where the y_ℓ's are variables. Since the order of y_i's in (16) does not matter, we can assume without loss of generality that $S_j = \{1, \ldots, |S_j|\}$, the indices in S_{j+1} are the next $|S_{j+1}|$ numbers, and so on S_{μ_α} is the last $|S_{\mu_\alpha}$ numbers smaller than N_j; thus the sets S_κ, for $\kappa = j, \ldots, \mu_\alpha$, form a partition of the set $\{1, \ldots, N_j\}$. Further define

$$i_0{:=}j - 1, \;\; i_\ell{:=}r_\ell - 1, \tag{25}$$

for $\ell = 1, \ldots, |S_j|$, and

$$i_\ell = \mu_\ell - 1, \tag{26}$$

for $\ell = |S_j| + 1, \ldots, N_j$. Then we replace the mth entry of the jth column $v_{j-1}(\alpha)$ in the matrix $V^{(j+1)}$ by

$$f_m^{(i_0,\ldots,i_{N_j})}[y_0, y_1, \ldots, y_{N_j}] \tag{27}$$

and substitute $y_0{:=}\alpha$, and $y_\ell{:=}\beta_\ell$, for $\ell = 1, \ldots, N_j$.This is done for all the n entries (that is, $m = 1, \ldots, n$) in the jth column. Let $V^{(j)}$ be the resulting matrix. From Lemma 3.3 we know that the coefficient of $f_m^{(i_0)}(y_0)$ is

$$\frac{1}{i_0!} \prod_{\ell=1}^{N_j} \frac{1}{(y_0 - y_\ell)^{i_\ell+1}} = \frac{1}{(j-1)!} \prod_{\kappa=j}^{\mu_\alpha} \prod_{\ell \in S_\kappa} \frac{1}{(\alpha - \beta_\ell)^{i_\ell+1}}, \tag{28}$$

which is same for all $m = 1, \ldots, n$. Therefore, the replacement of the entries of the jth column in the matrix $V^{(j+1)}$ by (27), for

$m = 1, \ldots, n$, is tantamount to obtaining the matrix $V^{(j)}$ from $V^{(j+1)}$ by replacing the jth column of $V^{(j+1)}$ by a linear combination of its other columns and a scaled version of the jth column, where the scaling factor is the product term in (28); the $1/(j-1)!$ is not part of the scaling as it already occurs in all the entries of the column (see (24)). In terms of the determinant, we obtain the following relation:

$$\begin{aligned}\det(V^{(j+1)}) &= \det(V^{(j)}) \prod_{\kappa=j}^{\mu_\alpha} \prod_{\ell \in S_\kappa} (\beta_\ell - \alpha)^{i_\ell+1} \\ &= \det(V^{(j)}) \prod_{\ell \in S_j} (\beta_\ell - \alpha)^{r_\ell} \prod_{\kappa=j+1}^{\mu_\alpha} \prod_{\ell \in S_\kappa} (\beta_\ell - \alpha)^{\mu_\ell}.\end{aligned}$$

Substituting this in (22), we have the desired inductive relation:

$$\det(V) = \det(V^{(j)}) \prod_{\kappa=j}^{\mu_\alpha} \prod_{\ell \in S_\kappa} (\beta_\ell - \alpha)^{(\kappa-j)\mu_\ell + r_\ell}.$$

We stop when $j = 1$, to get

$$\det(V) = \det(V^{(1)}) \prod_{\kappa=1}^{\mu_\alpha} \prod_{\ell \in S_\kappa} (\beta_\ell - \alpha)^{(\kappa-1)\mu_\ell + r_\ell}.$$

But recall from (20) that for an index $\ell \in S_\kappa$, we have $\lceil w_\ell/\mu_\ell \rceil = \kappa$. Furthermore, from (21) it follows that $w_\ell = (\kappa-1)\mu_\ell + r_\ell$. Since $\cup_{\kappa=1}^{\mu_\alpha} S_\kappa = [k]$, we have accounted for all the β_ℓ's, and so the equation above is the same as

$$\begin{aligned}\det(V) &= \det(V^{(1)}) \prod_{\ell=1}^{k} (\beta_\ell - \alpha)^{w_\ell} \\ &= \det(V^{(1)}) \prod_{\alpha_j \in \mathrm{In}(\alpha_i)} (\alpha_j - \alpha)^{w(\alpha_j, \alpha_i)}.\end{aligned}$$

Defining $V_i := V^{(1)}$ and recalling that $V = V_{i-1}$, we complete the proof of the inductive claim (19). Applying the claim for $i = 1$ to r, and making appropriate substitutions, we get the desired relation (29).

$$\det(V_0) = \det(V_r) \prod_{i=1}^{r} \prod_{\alpha_j \in \mathrm{In}(\alpha_i)} (\alpha_i - \alpha_j)^{w(\alpha_j, \alpha_i)}, \tag{29}$$

where $V_0 = V(\boldsymbol{\alpha}; \boldsymbol{\mu})$ (see (18)) The absolute value of the product on the right-hand side is the value that we need to lower bound; we know the determinant on the left-hand side from Proposition 2.1, so all that remains is to derive an upper bound on $|\det(V_r)|$. We will use Hadamard's inequality for this purpose, which requires us to derive an upper bound on the two-norms of the columns of the matrix V_r. Let $V_r(\alpha_i; j)$ **denote the jth column of the block of columns** $V_r(\alpha_i)$ corresponding to $B(\alpha_i)$ in V_0; note that $V_r(\alpha_i)$ may be the same as $B(\alpha_i)$ (this happens, for instance, when there are no edges incident on α_i in $\mathcal{G}$). In what follows, we derive an upper bound on $\|V_r(\alpha_i; j)\|_2$.

Recall the definition of N_j from (23), and that μ_i is the size of the block $B(\alpha_i)$. For convenience again, let the sets $S_j, S_{j+1}, \ldots, S_{\mu_i} \subseteq \mathrm{In}(\alpha_i)$ be indexed such that $S_j = \{1, \ldots, |S_j|\}$, the next $|S_{j+1}|$ numbers are in S_{j+1} and so on until S_{μ_i} is the last $|S_{\mu_i}|$ numbers smaller than N_j; thus these sets form a partition of the set $\{1, \ldots, N_j\}$. Now the mth entry in the column $V_r(\alpha_i; j)$ is (27). From (17), we have the following bound on the absolute value of (27) after substituting $n := N_j + 1$, $y_0 = \alpha_0 := \alpha_i$, $y_\ell := \alpha_\ell$, $\ell = 1, \ldots, N_j$, and the indices i_ℓ's are defined as in (25) and (26):

$$\sum_{\substack{(t_0, t_1, \ldots, t_{N_j}) \in \mathbb{Z}_{\geq 0}^{N_j+1} \\ t_0 + t_1 + \cdots + t_{N_j} = m-1-N_j}} \prod_{\ell=0}^{N_j} \binom{t_\ell}{i_\ell} |\alpha_\ell|^{t_\ell - i_\ell}.$$

Since $\alpha_1, \ldots, \alpha_{N_j}$ have edges directed to α_i, their absolute values are smaller than $|\alpha_i|$. Therefore, the quantity above is upper bounded by

$$\sum_{\substack{(t_0, t_1, \ldots, t_{N_j}) \in \mathbb{Z}_{\geq 0}^{N_j+1} \\ t_0 + t_1 + \cdots + t_{N_j} = m-1-N_j}} \prod_{\ell=0}^{N_j} \binom{t_\ell}{i_\ell} |\alpha_i|^{m-1-N_j-i_\ell},$$

which is equal to

$$|\alpha_i|^{m-1-N_j-\sum_{\ell=0}^{N_j} i_\ell} \sum_{\substack{(t_0, t_1, \ldots, t_{N_j}) \in \mathbb{Z}_{\geq 0}^{N_j+1} \\ t_0 + t_1 + \cdots + t_{N_j} = m-1-N_j}} \prod_{\ell=0}^{N_j} \binom{t_\ell}{i_\ell}. \tag{30}$$

Define

$$M_j := N_j + \sum_{\ell=0}^{N_j} i_\ell = N_j + j - 1 + \sum_{\ell=1}^{N_j} i_\ell, \tag{31}$$

where the second equality follows from the fact that $i_0 = j - 1$ (see the definition in (25)). The binomial coefficients $\binom{t_\ell}{i_\ell}$ vanish for $t_\ell < i_\ell$, so we can assume that $t_\ell \geq i_\ell$. If $j_\ell := t_\ell - i_\ell$, then

$$\sum_{\ell=0}^{N_j} t_\ell = \sum_{\ell=0}^{N_j} i_\ell + \sum_{\ell=0}^{N_j} j_\ell,$$

and so the constraint $\sum_{\ell=0}^{N_j} t_\ell = m - 1 - N_j$ is equivalent to

$$\sum_{\ell=0}^{N_j} j_\ell = m - 1 - N_j - \sum_{\ell=0}^{N_j} = m - 1 - M_j$$

where the last step follows from the definition of M_j (31). Changing the indices from t_ℓ to j_ℓ in (30), we get the following bound the mth entry of $V_r(\alpha_i; j)$:

$$|\alpha_i|^{m-1-M_j} \sum_{\substack{(j_0, j_1, \ldots, j_{N_j}) \in \mathbb{Z}_{\geq 0}^{N_j+1} \\ j_0 + j_1 + \cdots + j_{N_j} = m-1-M_j}} \prod_{\ell=0}^{N_j} \binom{i_\ell + j_\ell}{i_\ell}. \tag{32}$$

We next derive a closed form for the summation term above.

Consider the generating function

$$\sum_{t_\ell \geq i_\ell} \binom{t_\ell}{i_\ell} x^{t_\ell - i_\ell} = \sum_{j_\ell \geq 0} \binom{i_\ell + j_\ell}{j_\ell} x^{j_\ell} = (1-x)^{-(i_\ell+1)}$$

for a given ℓ. Taking the product of these for different choices of ℓ, it follows that the summation term in the right-hand side of (32) is the coefficient of x^{m-1-M_j} in the generating function

$$(1-x)^{-\sum_{\ell=0}^{N_j}(i_\ell+1)} = (1-x)^{-(M_j+1)}$$

which is

$$\binom{m-1}{M_j}.$$

This implies that (32) is equal to $|\alpha_i|^{m-1-M_j}\binom{m-1}{M_j}$.

From the argument in the preceding paragraph, it follows that in the matrix V_r the two-norm of the jth column, in the block of columns corresponding to $B(\alpha_i)$, is

$$\|V_r(\alpha_i;j)\|_2 \leq \left(\sum_{m=1}^{n} |\alpha_i|^{2(m-1-M_j)}\binom{m-1}{M_j}^2\right)^{1/2}.$$

Since for $m-1 \leq M_j$ the binomial term vanishes, we can start the summation from M_j onwards to obtain the following equivalent form

$$\|V_r(\alpha_i;j)\|_2 \leq \left(\sum_{m=M_j}^{n-1} |\alpha_i|^{2(m-M_j)}\binom{m}{M_j}^2\right)^{1/2}.$$

Substituting $|\alpha_i|$ by

$$\max_1 |\alpha_i| := \max\{1, |\alpha_i|\} \tag{33}$$

and pulling out its largest power from the summation we have the following upper bound on the two-norm

$$\|V_r(\alpha_i;j)\|_2 \leq \max_1 |\alpha_i|^{(n-1-M_j)}\left(\sum_{m=M_j}^{n-1}\binom{m}{M_j}^2\right)^{1/2}.$$

Using the upper bound from [7, Lemma 7] on the summation term above, we get the following inequality

$$\|V_r(\alpha_i;j)\|_2 \leq \max_1 |\alpha_i|^{(n-1-M_j)}\left(\frac{n}{\sqrt{3}}\right)^{M_j}\sqrt{n}.$$

Taking the product of these quantities for $j = 1,\ldots,\mu_i$, we get the following upper bound on the product of the two-norms of the columns in the block $V_r(\alpha_i)$ in V_r:

$$\prod_{j=1}^{\mu_i}\|V_r(\alpha_i;j)\|_2 \leq \max_1 |\alpha_i|^{\sum_{j=1}^{\mu_i}(n-1-M_j)}\left(\frac{n}{\sqrt{3}}\right)^{\sum_{j=1}^{\mu_i}M_j} n^{\mu_i/2}. \tag{34}$$

Let us understand the term $\sum_{j=1}^{\mu_i} M_j$.

Lemma 4.1. *For a vertex α_i in the directed acyclic graph $\mathcal{G}$, define*

$$w_i := \sum_{\alpha_\ell \in In(\alpha_i)} w(\alpha_\ell, \alpha_i), \tag{35}$$

that is, the sum of the weights of all edges incident on α_i. Then

$$\sum_{j=1}^{\mu_i} M_j = \binom{\mu_i}{2} + w_i.$$

Proof. Recall the definition of the sets S_j, from (20), and the definition of M_j, from (31). Given a j, and $\ell \in S_j$, $i_\ell = r_\ell - 1$ from (25); for $\ell \in S_\kappa$, where $\kappa = j+1,\ldots,\mu_i$, $i_\ell = \mu_\ell - 1$. Therefore, we can rewrite (31) as

$$\begin{aligned} M_j &= N_j + j - 1 + \sum_{\ell\in S_j}(r_\ell - 1) + \sum_{\ell\in\cup_{\kappa>j}S_\kappa}(\mu_\ell - 1) \\ &= j - 1 + \sum_{\ell\in S_j} r_\ell + \sum_{\ell\in\cup_{\kappa>j}S_\kappa}\mu_\ell. \end{aligned}$$

The sum $\sum_j\sum_{\ell\in S_j} r_\ell$ is the sum of the residue terms over all indices in $\cup_{j=1}^{\mu_i} S_j$. Now consider the sum

$$\sum_{j=1}^{\mu_i}\sum_{\ell\in\cup_{\kappa>j}S_\kappa}\mu_\ell.$$

For two indices $j < \kappa$, the summation over j contributes an μ_ℓ for every $\ell \in S_\kappa$. Therefore,

$$\sum_{j=1}^{\mu_i}\left(\sum_{\ell\in S_j} r_\ell + \sum_{\ell\in\cup_{\kappa>j}S_\kappa}\mu_\ell\right) = w_i$$

□

Substituting the result in the lemma above into (34), we get the following upper bound on the two-norms of the columns in $V_r(\alpha_i)$

$$\prod_{j=1}^{\mu_i}\|V_r(\alpha_i;j)\|_2 \leq \max_1 |\alpha_i|^{(n-1)\mu_i - \binom{\mu_i}{2} - w_i}\left(\frac{n}{\sqrt{3}}\right)^{\binom{\mu_i}{2}+w_i} n^{\mu_i/2}.$$

Taking the product of this bound for $i = 1,\ldots,r$, along with Hadamard's inequality, gives us the following upper bound

$$|\det(V_r)| \leq \prod_{i=1}^{r}\left(\max_1 |\alpha_i|^{(n-1)\mu_i - \binom{\mu_i}{2} - w_i}\left(\frac{n}{\sqrt{3}}\right)^{\binom{\mu_i}{2}+w_i}\right) n^{n/2}. \tag{36}$$

where we use the fact that $n = \sum_{i=1}^{r}\mu_i$. The term

$$(n-1)\mu_i - \binom{\mu_i}{2} - w_i = \sum_{j=1;j\neq i}^{r}\mu_i\mu_j - w_i + \binom{\mu_i}{2} < \sum_{\substack{j=1\\ j\neq i}}^{r}\mu_i\mu_j - w_i + \mu_i^2.$$

If $\boldsymbol{\mu}$ be the column vector of all μ_i's, and A_w be the adjacency matrix with the (i,j)th entry as the weight $w(\alpha_i,\alpha_j)$ of the corresponding edge (α_i,α_j), then the last term in the inequality above is the one-norm of the ith row of the matrix $\boldsymbol{\mu}\boldsymbol{\mu}^t - A_w$. Since the ∞-norm of the matrix $\|\boldsymbol{\mu}\boldsymbol{\mu}^t - A_w\|_\infty$ is the maximum over all the row-sums, we have

$$(n-1)\mu_i - \binom{\mu_i}{2} - w_i \leq \|\boldsymbol{\mu}\boldsymbol{\mu}^t - A_w\|_\infty.$$

As for the term

$$\sum_{i=1}^{r}\left(\binom{\mu_i}{2} + w_i\right) = \sum_{i=1}^{r}\binom{\mu_i}{2} + w(E),$$

where $w(E)$ is defined in (12). Substituting these bounds in (36), we obtain the following upper bound

$$|\det(V_r)| \leq M(\boldsymbol{\alpha})^{\|\boldsymbol{\mu}\boldsymbol{\mu}^t - A_w\|_\infty}\left(\frac{n}{\sqrt{3}}\right)^{\sum_i\binom{\mu_i}{2}+w(E)} n^{n/2}. \tag{37}$$

Substituting this upper bound in (29) and moving it to the denominator in the left-hand side completes the proof of Theorem 3.2.

4.1 Choosing the best matrix

Theorem 3.2 leaves open the choice of the potentials $\mu_i \in \mathbb{N}$, $i = 1,\ldots,r$. Our aim here is to find the best possible choice of μ_i's satisfying the edge constraints $w(\alpha_i,\alpha_j) \leq \mu_i\mu_j$ and at the same time minimizing $\|\boldsymbol{\mu}\boldsymbol{\mu}^t - A_w\|_\infty$. For example, if all the weights are

one then it is clear that $\mu_i = 1$, for $i = 1, \ldots, r$, is the best possible assignment. In which case,

$$V(\boldsymbol{\alpha}; \boldsymbol{\mu}) = V(\boldsymbol{\alpha}),\ \ \|\boldsymbol{\mu}\boldsymbol{\mu}^t - A_w\|_\infty \leq (r-1),\ \ n = r,\ \ w(E) = |E|$$

and so Theorem 3.2 matches the bound given in Proposition 3.1.

Consider the relaxed version of the problem where μ_i's are positive reals; it is clear that rounding them up to the nearest integer would give a valid solution (though not an optimum solution) to the problem over the positive integers. Then the optimization problem is to minimize $\|\boldsymbol{\mu}\boldsymbol{\mu}^t - A_w\|_\infty$ such that

$$\boldsymbol{\mu}\boldsymbol{\mu}^t \geq A_w$$

where '$\geq$' here means entry wise; note that the non-edge constraints are trivially satisfied since no μ_i is ever assigned to zero. Since A_w is non-negative, we know from the Perron-Frobenius theory [9] that the spectrum of A_w is an eigenvalue $\rho(A_w)$ of A_w. Moreover, as A_w is symmetric it can be orthogonally diagonalized, i.e., $A_w = Q\Lambda Q^t$, where Q is the $r \times r$ orthogonal matrix whose columns $\boldsymbol{q}_k$, $k = 1, \ldots, r$, are the eigenvectors of A_w and Λ is a diagonal matrix that has the corresponding eigenvalues of A_w. Another way to express the relation is that A_w is the sum of some rank one matrices obtained by its eigenvectors, i.e.,

$$A_w = \sum_{k=1}^{r} \lambda_k \boldsymbol{q}_k \boldsymbol{q}_k^t.$$

We can also assume that the $\|\boldsymbol{q}_k\|_2 = 1$ for $k = 1, \ldots, r$. Combined with the equation above it follows that the (i, j)-th entry of A_w

$$w(\alpha_i, \alpha_j) = \sum_{k=1}^{r} \lambda_k \boldsymbol{q}_{k,i} \boldsymbol{q}_{k,j}.$$

Since by assumption $\|\boldsymbol{q}_k\|_2 = 1$, taking absolute values we get

$$w(\alpha_i, \alpha_j) \leq \sum_{k=1}^{r} |\lambda_k| = \|A_w\|_\star,$$

where $\|A_w\|_\star$ is the **nuclear norm** of A_w. Therefore, we can take $\boldsymbol{\mu}$ in Theorem 3.2 as the vector

$$\boldsymbol{\mu} := \left\lceil \sqrt{\|A_w\|_\star} \right\rceil \overbrace{(1, 1, \ldots, 1)}^{r}, \tag{38}$$

which implies that

$$n = r \left\lceil \sqrt{\|A_w\|_\star} \right\rceil$$

in the theorem. The error in the approximation can be shown to be bounded by

$$\|\boldsymbol{\mu}\boldsymbol{\mu}^t - A_w\|_\infty \leq 2r\|A_w\|_\star,$$

and

$$\sum_i \binom{\mu_i}{2} \leq \frac{3r\|A_w\|_\star}{2},$$

where in the last inequality we use the observation that as A_w has entries in $\mathbb{Z}_{\geq 0}$, its spectrum is greater than one, and hence $\|A_w\|_\star \geq 1$. By making these substitutions in Theorem 3.2, we obtain the result, namely (4), mentioned in Section 1.

5 CONCLUSION AND FUTURE WORK

Our derivation using the confluent Vandermonde matrix to get the desired weights in the exponents has the advantage of optimizing over the various choices of the matrix. We have given a first attempt at exploiting this choice. Whereas rank-one approximations to matrices are well studied [8], the challenge in our context is to derive a symmetric rank-one matrix that also *dominates* A_w.

One would also like to derive a lower bound on the absolute value of $\det(V(\boldsymbol{\alpha}; \boldsymbol{\mu}))$ in terms of the polynomial f, to get a more direct comparison with the earlier results. Perhaps an algorithm to compute the determinant from the coefficients would also be interesting; a related recent result is an algorithm to compute the $D^+(f)$-root function defined as $\prod_{1 \leq i < j \leq r} (\alpha_i - \alpha_j)^{m_i + m_j}$, i.e., G is the complete graph on the roots and the weight of an edge is the sum of the multiplicity of its vertices [14]. Similar to [6], one would like to derive weighted version of the results for the more general setting of polynomial systems.

REFERENCES

[1] Prashant Batra and Vikram Sharma. 2019. Complexity of a Root Clustering Algorithm. *CoRR* abs/1912.02820 (2019). https://arxiv.org/abs/1912.02820

[2] Ruben Becker, Michael Sagraloff, Vikram Sharma, and Chee Yap. 2018. A near-optimal subdivision algorithm for complex root isolation based on the Pellet test and Newton iteration. *Journal of Symbolic Computation* 86 (2018), 51 – 96. https://doi.org/10.1016/j.jsc.2017.03.009

[3] James H. Davenport. 1985. *Computer algebra for Cylindrical Algebraic Decomposition.* Tech. Rep. The Royal Inst. of Technology, Dept. of Numerical Analysis and Computing Science, S-100 44, Stockholm, Sweden. Reprinted as Tech. Report 88-10 , School of Mathematical Sci., U. of Bath, Claverton Down, Bath BA2 7AY, England. URL http://www.bath.ac.uk/~masjhd/TRITA.pdf.

[4] Arno Eigenwillig. 2008. *Real Root Isolation for Exact and Approximate Polynomials Using Descartes' Rule of Signs.* Ph.D. Thesis. University of Saarland, Saarbruecken, Germany.

[5] Arno Eigenwillig, Vikram Sharma, and Chee Yap. 2006. Almost Tight Complexity Bounds for the Descartes Method. In *Proc. of the 31st Intl. Symp. on Symbolic and Algebraic Computation.* 71–78. Genova, Italy. Jul 9-12, 2006.

[6] Ioannis Emiris, Bernard Mourrain, and Elias Tsigaridas. 2019. Separation bounds for polynomial systems. *Journal of Symbolic Computation* (2019). https://doi.org/10.1016/j.jsc.2019.07.001

[7] Paula Escorcielo and Daniel Perrucci. 2017. On the Davenport-Mahler bound. *J. Complexity* 41 (2017), 72–81. https://doi.org/10.1016/j.jco.2016.12.001

[8] Shmuel Friedland. 2013. Best rank one approximation of real symmetric tensors can be chosen symmetric. *Frontiers of Mathematics in China* 8 (2013), 19–40. https://doi.org/10.1007/s11464-012-0262-x.

[9] R. Horn and C. Johnson. 1991. *Topics in Matrix Analysis.* Cambridge University Press, Cambridge.

[10] Alexander Kobel and Michael Sagraloff. 2015. On the complexity of computing with planar algebraic curves. *J. Complexity* 31, 2 (2015), 206–236. https://doi.org/10.1016/j.jco.2014.08.002

[11] Maurice Mignotte. 1992. *Mathematics for Computer Algebra.* Springer-Verlag, Berlin.

[12] Maurice Mignotte. 1995. On the Distance Between the Roots of a Polynomial. *Applicable Algebra in Engineering, Commun., and Comput.* 6 (1995), 327–332.

[13] Victor Y. Pan. 2002. Univariate polynomials: Nearly optimal algorithms for numerical factorization and root-finding. *Journal of Symbolic Computation* 33, 5 (2002), 701–733.

[14] Jing Yang and Chee K. Yap. 2020. On mu-Symmetric Polynomials. *CoRR* abs/2001.07403 (2020). https://arxiv.org/abs/2001.07403

General Witness Sets for Numerical Algebraic Geometry

Frank Sottile, Department of Mathematics, Texas A&M University, College Station, Texas 77843, USA

Frank Sottile
sottile@math.tamu.edu

ABSTRACT

Numerical algebraic geometry has a close relationship to intersection theory from algebraic geometry. We deepen this relationship, explaining how rational or algebraic equivalence gives a homotopy. We present a general notion of witness set for subvarieties of a smooth complete complex algebraic variety using ideas from intersection theory. Under appropriate assumptions, general witness sets enable numerical algorithms such as sampling and membership. These assumptions hold for products of flag manifolds. We introduce Schubert witness sets, which provide general witness sets for Grassmannians and flag manifolds.

CCS CONCEPTS

• **Computing methodologies → Hybrid symbolic-numeric methods**.

KEYWORDS

numerical algebraic geometry, intersection theory, witness set, Schubert variety

ACM Reference Format:
Frank Sottile. 2020. General Witness Sets for Numerical Algebraic Geometry. In *International Symposium on Symbolic and Algebraic Computation (ISSAC '20), July 20–23, 2020, Kalamata, Greece.* ACM, New York, NY, USA, 8 pages. https://doi.org/10.1145/3373207.3403995

INTRODUCTION

Numerical algebraic geometry uses numerical analysis to study algebraic varieties. Its foundations rest on numerical homotopy continuation, which enables the numerical computation of solutions to systems of polynomial equations [26]. It relies on the fundamental concept of a witness set [24, 25], which is a data structure for representing a subvariety of affine or projective space on a computer. Witness sets also appear in symbolic computation under the term lifting fiber, and have been used to establish the complexity of computing points on an algebraic variety [10].

A witness set for an irreducible variety V of dimension k is a triple, (F, Λ, W), where F is a system of polynomial equations whose zero set contains V as a component and Λ is a general linear space of codimension k (represented by k general linear polynomials) which meets V transversally in the finite set W of points. Numerical continuation of the points W when Λ is moved allows one to, for example, sample points from V. Consequently, W may be considered to be a generic point of V in the sense of Weil [28].

A witness set for a subvariety also represents its fundamental cycle in homology. The homology of projective space has a basis given by classes $[L]$ of linear spaces. Since linear spaces satisfy duality—$L \cap \Lambda$ is a point when L and Λ are general linear spaces of complementary dimension—the homology class $[V]$ of a subvariety V of dimension k is determined by its degree, which is the number of points in its intersection with a general linear space Λ of codimension k. That is, if L is a linear space of dimension k, then

$$[V] \;=\; \deg(V \cap \Lambda) \cdot [L]\,.$$

In a witness set, we replace the number $\deg(V \cap \Lambda)$ by the set $W := V \cap \Lambda$ and require that the intersection be transverse, which we may, by Bertini's Theorem.

The concept of witness sets and their manipulation is linked to ideas from intersection theory [6, 7]. A witness set W is a concrete representation of the localized intersection product $[V] \bullet [\Lambda] \in H_0(V \cap \Lambda)$ [6, Ch. 8]. As W is a set of $\deg(V)$ points of V, we are implicitly working in the group of cycles modulo numerical equivalence. As a homotopy is a family of varieties (or points) over $\mathbb{C}$, homotopies are connected to the notion of rational equivalence.

We propose a notion of witness set for subvarieties of a smooth algebraic variety X, based on ideas from intersection theory. This requires an equivalence relation, such as numerical equivalence, on algebraic cycles such that the resulting group of cycles on X is a finitely generated free abelian group on which the intersection pairing is nondegenerate. Choosing an additive basis of cycles gives general witness sets for subvarieties of X. With additional assumptions (given in Section 3) this notion is refined, and there are algorithms using general witness sets such as changing a witness set, sampling, and membership testing.

Products of projective spaces satisfy these additional assumptions, and these ideas for such products were proposed in [12]. These assumptions hold for flag manifolds, where the natural general witness sets are Schubert witness sets. We explain how Schubert witness sets enable numerical continuation algorithms for sampling and membership.

Numerical algebraic geometry operates on the geometric side of algebraic geometry, with algorithms based on geometric constructions, such as fiber products [27], images of maps [13], and monodromy [26, §15.4]. It is also suited for intersection theory, using excess intersection formulas to compute Chern numbers [4]. Understanding witness sets in terms of intersection theory is a natural continuation.

Research of Sottile supported in part by NSF grant DMS-1501370 and Simons Foundation Collaboration Grant for Mathematics Number 636314.

ISSAC '20, July 20–23, 2020, Kalamata, Greece

ACM ISBN 978-1-4503-7100-1/20/07…$15.00
https://doi.org/10.1145/3373207.3403995

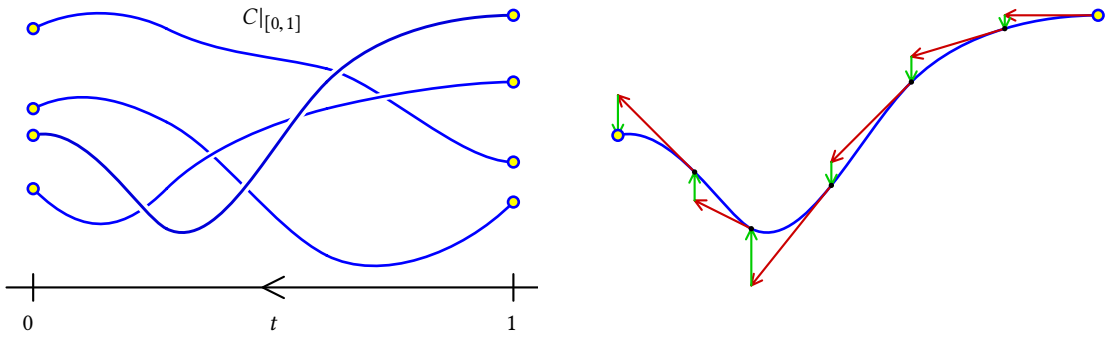


Figure 1: Paths over $[0,1]$ and predictor-corrector steps.

This paper is organized as follows. Section 1 gives background from numerical algebraic geometry, including numerical continuation, witness sets, and some fundamental algorithms. Section 2 gives background from intersection theory and explains the connection of rational equivalence to numerical homotopy continuation. We present general witness sets in Section 3, and explain how additional hypotheses enable algorithms for sampling and membership. In Section 4 we introduce Schubert witness sets, which are the natural general witness sets for flag manifolds and explain the fundamental algorithms for Schubert witness sets.

1 CLASSICAL WITNESS SETS

We review aspects of numerical algebraic geometry as may be found in [2, 26].

1.1 Homotopy continuation

A *homotopy* is a polynomial map

$$H \;=\; H(x;t) \;:\; \mathbb{C}^n \times \mathbb{C} \;\longrightarrow\; \mathbb{C}^N\,, \tag{1}$$

where $H^{-1}(0) \subset \mathbb{C}^n \times \mathbb{C}$ defines an algebraic curve C with a dominant projection to the distinguished (t) coordinate, $\mathbb{C}$. We suppose that 1 is a regular value of the projection to $\mathbb{C}$ and we know the points of the fiber, and we use them to obtain the points of the fiber over 0.

For example, suppose that $F = (f_1, \ldots, f_n)$ with f_i a polynomial of degree d_i. Then the *Bézout homotopy*

$$H(x;t) \;:=\; (1-t)F \;+\; t(x_i^{d_i}-1 \mid i = 1, \ldots, n)$$

connects the points over $t = 1$, $(x_1, \ldots, x_n)$ where x_i is a d_i-th root of unity, to the unknown solutions to $F = 0$.

Restricting t to the interval $[0,1] \subset \mathbb{C}$ (more generally to a path in $\mathbb{C}$ connecting 1 to 0 [2, § 2.1]), the algebraic curve C becomes a collection of real paths in $\mathbb{C}^n \times [0,1]$ connecting points in the fiber at $t = 1$ to those at $t = 0$. A point $(x, 1)$ in the fiber of C at $t = 1$ lies on a unique path, and standard predictor-corrector methods construct a sequence $(x_1, t_1), \ldots, (x_s, t_s)$ of points along that path with $1 > t_1 > \cdots > t_s = 0$ so that x_s is a solution to $H(x, 0) = 0$. This is illustrated in Figure 1.

These numerical algorithms do not compute points on paths or on varieties, but rather refinable approximations to such points. This uses Newton's method which replaces a point $x \in \mathbb{C}^n$ by the result of a Newton step

$$N_F(x) \;:=\; x \;-\; (DF(x))^{-1}(x)\,,$$

where $F\colon \mathbb{C}^n \to \mathbb{C}^n$ is a polynomial map and $DF(x)$ is its Jacobian derivative. When x is sufficiently close to a solution x^* of F, the sequence of iterations defined by $x_0 := x$ and $x_i = N_F(x_{i-1})$ for $i > 0$ satisfies

$$\|x_i - x^*\| \;<\; \left(\frac{1}{2}\right)^{2^i-1} \|x - x^*\|\,.$$

When this occurs, we say that x *converges quadratically* to x^*. Smale's α-theory [22] involves a computable [14] constant $\alpha(F,x)$ such that if $\alpha(F,x) < (13-3\sqrt{17})/4$, then x converges quadratically to a solution. Other approaches to certification (e.g. Krawcyzk's Method [18, 20]) use interval arithmetic [21].

We ignore the question of whether our approximations lie in the basin of quadratic convergence under Newton iterations and simply refer to them as solutions, state that they lie on paths or on varieties, *et cetera*.

1.2 Witness sets and algorithms

Algorithms based on numerical homotopy continuation can compute the isolated solutions to a system of polynomial equations $F(x) = 0$ and follow solutions along homotopy paths. Sommese, Verschelde, and Wampler [24, 25] introduced the notion of a witness set, which enables the representation and manipulation of algebraic subvarieties of $\mathbb{C}^n$ using numerical homotopy continuation.

Let $F\colon \mathbb{C}^n \to \mathbb{C}^N$ be a polynomial map and $V \subset \mathbb{C}^n$ a union of irreducible components of $F^{-1}(0)$ of dimension k. A *witness set* for V is a triple (F, Λ, W) where $\Lambda\colon \mathbb{C}^n \to \mathbb{C}^k$ is k general affine forms and $W = V \cap \Lambda^{-1}(0)$. As Λ is general, W consists of $\deg(V)$ points and the intersection is transverse.

We may use a witness set W to compute other witness sets. If Λ' is another set of k independent affine forms, the convex combination $\Lambda(t) := (1-t)\Lambda' + t\Lambda$ may be used with F to define a homotopy that connects the points W at $t = 1$ to points $W' := V \cap (\Lambda')^{-1}(0)$ at $t = 0$. Numerical continuation along this homotopy computes the points W' (when finite) from the points W. When Λ' is general, we obtain another witness set (F, Λ', W') for V.

As every point of V lies on some affine subspace of codimension k which meets V properly, continuation of a witness set along such homotopies samples points of V. Moreover, if $p \in \mathbb{C}^n$ and we

choose Λ' such that $\Lambda'(p) = 0$, but Λ' is otherwise general, then $p \in V$ if and only if $p \in W'$. These three algorithms, moving a witness set, sampling points of a variety, and the membership test, are fundamental methods to study a variety V given a witness set, and form the basis for more sophisticated algorithms.

2 INTERSECTION THEORY

We recall aspects of algebraic cycles and intersection theory, and then discuss how rational equivalence leads to homotopies as in Subsection 1.1. This material, with proofs, is found in Chapters 1 and 19 of [6]. Other sources include [5, 7].

Let X be a smooth algebraic variety of dimension n. If $V, \Lambda \subset X$ are subvarieties of dimensions k and l with $k+l \geq n$, then either $V \cap \Lambda$ is empty or it has dimension at least $k+l-n$. It is *proper* if it has this expected dimension. The intersection $V \cap \Lambda$ is *transverse* at a point $p \in V \cap \Lambda$ when both V and Λ are smooth at p and their tangent spaces at p span the tangent space of X at p, $T_pV + T_p\Lambda = T_pX$. The intersection $Y \cap Z$ is *generically transverse* if it is transverse at a dense set of points $p \in V \cap \Lambda$. Generically transverse is necessary as any of V, Λ, or $V \cap \Lambda$ may have singular points. Generically transverse intersections are proper.

2.1 Intersection theories

Let X be a connected, complete, smooth, irreducible complex algebraic variety of dimension n. For each $0 \leq k \leq n$, the group Z_kX of k-cycles on X is the free abelian group generated by the k-dimensional irreducible subvarieties of X. The *fundamental cycle* $[V]$ of an irreducible subvariety V of X is the corresponding generator of Z_kX. A subscheme $V \subset X$ of dimension k also has a fundamental cycle. For each irreducible component Λ of V of dimension k, let $m_{\Lambda,V}$ be its multiplicity in V, which is the generic multiplicity of V along Λ. The fundamental cycle of V is

$$[V] \;:=\; \sum_{\Lambda} m_{\Lambda,V}[\Lambda]\,.$$

A cycle $\sum \alpha_V[V]$ with $\alpha_V \geq 0$ is *effective*. If $\alpha_V \in \{0,1\}$, it is *multiplicity-free*. The fundamental cycle of a generically transverse intersection is multiplicity-free. A map $\iota\colon Y \to X$ of varieties induces a map $\iota_*\colon Z_kY \to Z_kX$. When ι is an inclusion and $V \subset Y$ is a subscheme, $\iota_*[V] = [\iota(V)]$.

Sending a subvariety V to its fundamental cycle in homology induces the *cycle class map* $\mathrm{cl}\colon Z_kX \to H_{2k}(X,\mathbb{Q})$. Its kernel is the group $\mathrm{Hom}_k\, X$ of k-cycles with an integer multiple homologically equivalent to zero and its image is the k-th *algebraic homology* $H_k^{\mathrm{alg}}X$ of X. (The shift in homological degree from $2k$ to k is for notational consistency.) The group $H_k^{\mathrm{alg}}X$ is a finitely generated free abelian group. As X is smooth, homology has an intersection product which induces a bilinear map $H_k^{\mathrm{alg}}X \times H_l^{\mathrm{alg}}X \to H_{k+l-n}^{\mathrm{alg}}X$, where $(\alpha,\beta) \mapsto \alpha \cdot \beta$. When $k+l=n$, this gives the *intersection pairing* $H_{n-k}^{\mathrm{alg}}X \times H_k^{\mathrm{alg}}X \to H_0^{\mathrm{alg}}X = \mathbb{Z}$.

Suppose that $Y \subset X \times \mathbb{P}^1$ is an irreducible subvariety of dimension $k+1$ with projections ι to X and f to $\mathbb{P}^1$,

$$\begin{array}{ccc} & Y \subset X \times \mathbb{P}^1 & \\ {\scriptstyle\iota}\swarrow & & \searrow{\scriptstyle f} \\ X & & \mathbb{P}^1 \end{array} \tag{2}$$

where f is surjective. The fibers of f are naturally subschemes of X of dimension k. Call the cycle

$$[\iota(f^{-1}(0))] \;-\; [\iota(f^{-1}(1))] \;\in\; Z_kX \tag{3}$$

an *elementary rational equivalence*. Elementary rational equivalences generate the subgroup $\mathrm{Rat}_k\, X \subset Z_kX$ of k-cycles rationally equivalent to zero. The quotient $A_kX := Z_kX/\,\mathrm{Rat}_k\, X$ is the k-th *Chow group* of X. As X is smooth, there is an intersection product as with homology.

Let V and Λ be subvarieties of X of dimension k and l. The localized intersection product [6, Ch. 8] of their fundamental cycles is a cycle class

$$[V] \bullet [\Lambda] \;\in\; A_{k+l-n}(V \cap \Lambda)\,. \tag{4}$$

Its image in $A_{k+l-n}X$ under the map induced by the inclusion $V \cap \Lambda \hookrightarrow X$ is the intersection product $[V] \cdot [\Lambda]$. When the intersection is proper, the localized intersection product is the fundamental cycle of the scheme-theoretic intersection, $[V \cap \Lambda]$.

Let $\deg\colon A_0X \to \mathbb{Z}$ be the degree map on 0-cycles

$$\deg:\; \sum m_p[p] \;\longmapsto\; \sum m_p\,,$$

the sum over $p \in X$. Note that only finitely many coefficients m_p are non-zero. Composing with the product gives an intersection pairing $A_{n-k}X \times A_kX \to A_0X \to \mathbb{Z}$ as before.

If we replace $\mathbb{P}^1$ by an irreducible curve T and $0, 1 \in \mathbb{P}^1$ by two smooth points of T in the definition of rational equivalence, we obtain *algebraic equivalence*. Let $\mathrm{Alg}_k\, X \subset Z_kX$ be the group generated by differences of algebraically equivalent k-cycles, the group of cycles algebraically equivalent to zero. Let $B_kX := Z_kX/\,\mathrm{Alg}_k\, X$ be the group of cycles modulo algebraic equivalence. This has an intersection product and pairing $B_{n-k}X \times B_kX \to B_0X = \mathbb{Z}$ as before.

A cycle $\beta \in Z_kX$ is *numerically equivalent to zero* if, for every $\alpha \in A_{n-k}X$, we have $\deg(\alpha \cdot \overline{\beta}) = 0$, where $\overline{\beta}$ is the image of β in A_kX. Let $\mathrm{Num}_k\, X \subset Z_kX$ be the subgroup of k-cycles numerically equivalent to zero. Set $N_kX := Z_kX/\,\mathrm{Num}_k\, X$, which is a finitely generated free abelian group. The intersection pairing is nondegenerate by the definition of numerical equivalence.

Proposition 1. *For every $0 \leq k \leq n$ we have*

$$\mathrm{Rat}_k\, X \;\subset\; \mathrm{Alg}_k\, X \;\subset\; \mathrm{Hom}_k\, X \;\subset\; \mathrm{Num}_k\, X\,, \tag{5}$$

as subgroups of Z_kX. The maps $A_kX \to B_kX \to H_k^{\mathrm{alg}}X \to N_kX$ are compatible with the intersection product. The groups $H_k^{\mathrm{alg}}X$ and N_kX are finitely generated free abelian groups and the intersection pairing $N_{n-k}X \times N_kX \to \mathbb{Z}$ is nondegenerate.

Define A_*X to be the direct sum of the A_kX and the same for B_*X, H_*^{alg}, and N_*X.

Remark 2. The first two inclusions in (5) are strict in general. A conjectured equality of $\mathrm{Hom}_k\, X$ and $\mathrm{Num}_k\, X$ is a consequence of Grothendieck's 'standard conjectures' [17, § 5]. The question of when the intersection pairing on N_*X is perfect is related to the representability of integral homology classes by algebraic cycles. ⋄

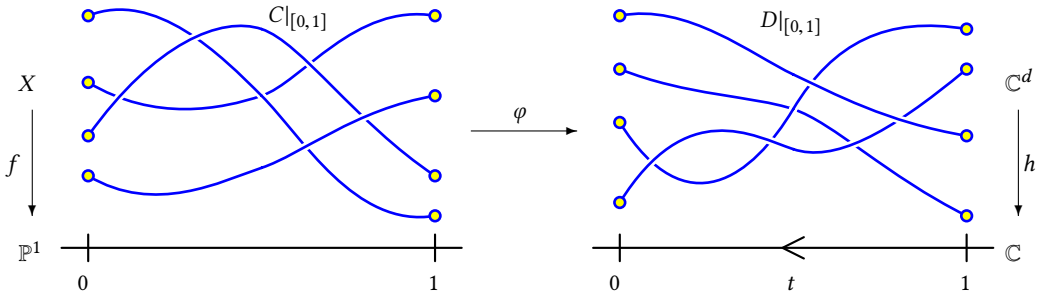


Figure 2: An elementary rational equivalence defines a homotopy.

2.2 Intersection theory and homotopy continuation

Elementary rational and algebraic equivalences give homotopies in the sense of Subsection 1.1.

Let $Y \subset X \times \mathbb{P}^1$ be an irreducible subvariety of dimension $n-k+1$ having projections ι to X and f to $\mathbb{P}^1$ with f surjective as in (2). This gives an elementary rational equivalence (3) in $\mathrm{Rat}_{n-k} X$. Suppose that $V \subset X$ has dimension k and meets $\iota f^{-1}(1)$ transversally. Then $(V \times \mathbb{P}^1) \cap Y$ contains a curve C passing through $V \cap \iota f^{-1}(1)$. Writing g for the restriction of f to C gives a surjective map $g\colon C \to \mathbb{P}^1$. Then $g^{-1}[0,1]$ gives arcs in C connecting the points of $g^{-1}(1)$ to $g^{-1}(0)$ as in Figure 1. Choosing coordinates and equations for the varieties, we obtain a homotopy as in Subsection 1.1.

Theorem 3. *Let $Y \subset X \times \mathbb{P}^1$ give an elementary rational equivalence in $\mathrm{Rat}_{n-k} X$ (3) and $V \subset X$ be a subvariety of dimension k meeting $\iota f^{-1}(1)$ transversally with g the restriction of f to the curve $C = (V \times \mathbb{P}^1) \cap Y$. Let $U \subset X$ be an affine open set containing $\iota g^{-1}[0,1]$. For any embedding $\varphi\colon U \to \mathbb{C}^d$ there is a homotopy $H(x;t)$ defining a curve $D \subset \mathbb{C}^d \times \mathbb{C}$ with $\varphi^{-1}(D) = C \cap (U \times \mathbb{P}^1)$ and $\varphi^{-1}(D|_{[0,1]}) = g^{-1}[0,1]$.*

Proof. Let $\mathbb{C} \subset \mathbb{P}^1$ be an affine line containing $[0,1]$. Then the arcs $g^{-1}[0,1]$ lie in the curve $C^\circ := C \cap (U \times \mathbb{C})$. Let $\varphi\colon U \to \mathbb{C}^d$ be a map realizing U as a subvariety of $\mathbb{C}^d$. Then $\varphi \times id_{\mathbb{C}}$ realizes C° as a subvariety of $\mathbb{C}^d \times \mathbb{C}$. Let D be its closure and $h\colon D \to \mathbb{C}$ the projection map. Choosing any system of equations $H\colon \mathbb{C}^d \times \mathbb{C} \to \mathbb{C}^N$ with $D = H^{-1}(0)$ gives a homotopy. See Figure 2. □

Remark 4. This leads to a numerical homotopy algorithm to find the points of $\iota g^{-1}(0)$, given those of $\iota g^{-1}(1)$. Write $h\colon D \to \mathbb{C}$ for the projection. As $\varphi\iota g^{-1}(1) = h^{-1}(1)$, we may use the homotopy to trace these points along the arcs of $h^{-1}[0,1]$ to obtain the points of $h^{-1}(0)$. Since $h^{-1}(0) = \varphi\iota g^{-1}(0)$, applying φ^{-1} to $h^{-1}(0)$ gives $\iota g^{-1}(0)$. ⋄

Remark 5. Theorem 3 used rational equivalence as homotopy continuation assumes that t is rational ($t \in \mathbb{C}$). Given an elementary algebraic equivalence, replace $\mathbb{C}$ by a smooth affine curve T, the points 0 and 1 by points $p, q \in T$, and the interval $[0,1]$ by an arc γ on T connecting p to q. This gives arcs connecting points of $(V \times \mathbb{P}^1) \cap Y$ above p to points above q. Choosing coordinates (φ) gives a homotopy $H(x;t)$, but the parameter t is not rational, as it takes values in $\gamma \subset T$. This becomes a traditional homotopy by choosing a map $\psi\colon T \to \mathbb{P}^1$ with $\psi(p) = 1$ and $\psi(q) = 0$, and then the path γ from p to q gives a path $\psi(\gamma)$ between 1 and 0, which is followed in the homotopy. This is not a rational equivalence as only a subset of the points in a fiber $(\psi \circ \phi)^{-1}$ are followed along $\psi(\gamma)$ from 1 to 0 (these are the points above $\gamma \subset T$). ⋄

3 GENERAL WITNESS SETS

Let X be an algebraic variety and fix C_* to be an intersection theory as in Proposition 1 such that C_*X is a finitely generated free abelian group with nondegenerate intersection pairing. A basis for C_*X gives a normal form (6) for a fundamental cycle $[V]$, leading to general witness sets. We discuss when general witness sets may be moved and may be used for sampling and membership.

3.1 General witness sets

The kth Betti number b_k of X is the rank of the free $\mathbb{Z}$-module C_kX. While it has a $\mathbb{Z}$-basis of cycles $\alpha_1, \ldots, \alpha_{b_k} \in Z_kX$, these need not be effective. There are however, independent effective cycles $[L_1^{(k)}], \ldots, [L_{b_k}^{(k)}] \in Z_kX$, with each $L_i^{(k)}$ an irreducible subvariety of dimension k. These form a basis for the $\mathbb{Q}$-vector space $C_kX \otimes_{\mathbb{Z}} \mathbb{Q}$, called an *effective $\mathbb{Q}$-basis*. We work with a fixed choice of cycles $\{L_b^{(a)}\}$ that form an effective $\mathbb{Q}$-basis for C_*X.

For a subvariety V of X of dimension k, there are rational numbers $c_j(V)$ for $j = 1, \ldots, b_k$ defined by the expansion of the fundamental cycle of V in this basis,

$$[V] \;=\; \sum_{j=1}^{b_k} c_j(V)[L_j^{(k)}]\,. \tag{6}$$

The intersection pairing on $C_{n-k}X \times C_kX$ is encoded by the $b_{n-k} \times b_k$ integer matrix $M^{(k)}$ whose entries are

$$M^{(k)}_{i,j} \;:=\; \deg\Big([L_i^{(n-k)}] \cdot [L_j^{(k)}]\Big)\,, \tag{7}$$

where $i = 1, \ldots, b_{n-k}$ and $j = 1, \ldots, b_k$. As the intersection pairing is nondegenerate, $b_{n-k} = b_k$ and $M^{(k)}$ is invertible.

Consequently, if $c(V) := (c_j(V) \mid j = 1, \ldots, b_k)^T$ is the vector of coefficients in the representation (6) of V, then the vector of intersection multiplicities,

$$d(V) \;:=\; \Big(\deg([V] \cdot [L_1^{(n-k)}]), \ldots, \deg([V] \cdot [L_{b_{n-k}}^{(n-k)}])\Big)^T\,,$$

satisfies $d(V) = M^{(k)}c(V)$, and so we may recover the class (6) of V from these intersection multiplicities by inverting this relation, $c(V) = (M^{(k)})^{-1}d(V)$.

As $\dim V + \dim L_i^{(n-k)} = \dim X$, the product $[V] \cdot [L_i^{(n-k)}]$ is the image in C_0X of the localized product $[V] \bullet [L_i^{(n-k)}]$ in $C_0(V \cap L_i^{(n-k)})$. This in turn is the image of the localized intersection product (4) in $A_0(V \cap L_i^{(n-k)})$ under the map $A_* \to C_*$ of Proposition 1. When the intersection is proper (has dimension 0), $[V] \bullet [L_i^{(n-k)}]$ is the fundamental cycle $[V \cap L_i^{(n-k)}]$ of the intersection, which is

$$\sum_{p \in V \cap L_i^{(n-k)}} m_p \, p \,,$$

where m_p is the intersection multiplicity of $V \cap L_i^{(n-k)}$ at p.

Definition 6. Let $V \subset X$ be a subvariety of dimension k. A *general witness set* for V is a triple $(V, \Lambda_\bullet, W_\bullet)$, where $\Lambda_\bullet$ is a list $(\Lambda_1, \ldots, \Lambda_{b_{n-k}})$ of subvarieties of X such that $[\Lambda_i] = [L_i^{(n-k)}]$ and $W_\bullet$ is a list $(W_1, \ldots, W_{b_{n-k}})$ of cycles such that $W_i \in Z_0(V \cap \Lambda_i)$ represents the localized product $[V] \bullet [\Lambda_i]$. We call each component W_i a general witness set. ◇

By the preceeding discussion, general witness sets encode fundamental cycles.

Theorem 7. *Suppose that $(V, \Lambda_\bullet, W_\bullet)$ is a general witness set for V. The vector $c(V)$ of coefficients of $[V]$ in the basis $[L_j^{(k)}]$ of C_kX is obtained from the vector $\deg(W_\bullet) := (\deg(W_1), \ldots, \deg(W_{b_{n-k}}))^T$ of the degrees of the W_i by the formula $c(V) = (M^{(k)})^{-1} \deg(W_\bullet)$.*

Example 8. The cycles W_i are not necessarily effective. If $X := Bl_p\mathbb{P}^2$, the blow up of $\mathbb{P}^2$ in a point p, then $C_1X = [\ell]\mathbb{Z} + [E]\mathbb{Z}$ (this holds in any intersection theory), where ℓ is the proper transform of a line in $\mathbb{P}^2$ and E is the exceptional divisor. In this case, $M^{(1)} = \left(\begin{smallmatrix} 1 & 0 \\ 0 & -1 \end{smallmatrix}\right)$ as $[\ell]^2 = 1$, $[\ell] \cdot [E] = 0$, and $[E]^2 = -1$. A general witness set for E is $W_\bullet = (0, -[q])$, where $q \in E$. ◇

Example 9. Projective space $\mathbb{P}^n$ has free abelian Chow groups $A_*\mathbb{P}^n$. Here, $b_k = 1$ for $0 \le k \le n$ and $L^{(k)}$ is any k-dimensional linear subspace (k-plane). By Bertini's Theorem, a general $(n-k)$-plane Λ meets a k-dimensional subvariety V of $\mathbb{P}^n$ transversally in $\deg(V)$ points $W = V \cap \Lambda$. Thus classical witness sets are general witness sets. ◇

Chow groups are not typically finitely generated free abelian groups with a nondegenerate intersection pairing—e.g. if $\mathcal{E}$ is an elliptic curve, then $A_0\mathcal{E} = \mathcal{E} \times \mathbb{Z}$ and $A_1\mathcal{E} = \mathbb{Z}$. (This is remedied for $\mathcal{E}$ by algebraic equivalence as $B_0\mathcal{E} = B_1\mathcal{E} = \mathbb{Z}$.) Nevertheless, for many common varieties X, rational equivalence and numerical equivalence coincide. A sufficient condition is that X admits an action of a solvable linear algebraic group with finitely many orbits [9]. This class of varieties includes projective space, toric varieties, Grassmannians, flag manifolds, and spherical varieties. If X is such a space and Y any variety, then there is a Künneth isomorphism $A_*X \otimes A_*Y \xrightarrow{\sim} A_*(X \times Y)$, so products with these spaces preserve these properties.

Example 10. Hauenstein and Rodriguez [12] developed multiprojective witness sets for subvarieties of products of projective spaces, which are general witness sets for these varieties. Let $m, n \ge 1$. The Chow group of $\mathbb{P}^m \times \mathbb{P}^n$ is a free abelian group that is isomorphic to its cohomology. To describe a basis, for each $0 \le a \le m$ and $0 \le b \le n$, let $K_a \subset \mathbb{P}^m$ and $L_b \subset \mathbb{P}^n$ be linear subspaces of dimensions a and b, respectively. The classes $[K_a] \otimes [L_b] = [K_a \times L_b]$ with $a + b = k$ form a basis for $A_k(\mathbb{P}^m \times \mathbb{P}^n)$.

A subvariety $V \subset \mathbb{P}^m \times \mathbb{P}^n$ of dimension k has *bidegrees* $d_{a,b} = d_{a,b}(V)$ for $a + b = k$ defined by

$$[V] \;=\; \sum_{a+b=k} d_{a,b} \, [K_a \times L_b] \,,$$

where $0 \le a \le m$ and $0 \le b \le n$. A *multihomogeneous witness set* for V is a triple $(V, \Lambda_\bullet, W_\bullet)$ where

(i) For each (a, b) with $a + b = k$, $\Lambda_{a,b} = M^a \times N^b$ where $M^a \subset \mathbb{P}^m$ and $N^b \subset \mathbb{P}^n$ are linear subspaces of codimension a and b, respectively, such that
(ii) $W_{a,b} := V \cap \Lambda_{a,b}$ is transverse and therefore consists of $d_{a,b}$ points, and
(iii) $\Lambda_\bullet = \{\Lambda_{a,b} \mid a + b = k\}$ and $W_\bullet = \{W_{a,b} \mid a + b = k\}$.

Hauenstein and Rodriguez enrich this structure by representing V as a component of the solution set of a system of bihomogeneous polynomials and the linear subspaces M^a and N^b by general linear forms on their ambient projective spaces. They give algorithms based on multihomogeneous witness sets for moving a witness set, membership, sampling, regeneration, and numerical irreducible decomposition using a trace test [23]. An alternative trace test for multihomogeneous witness sets is developed in [19], and extensions to more than two factors are given in [11]. ◇

3.2 Moving, sampling and membership

While general witness sets provide a representation of a cycle class $[V]$, without further assumption, their utility is limited. We first describe how rational or algebraic equivalence allows a general witness set to be moved, and then discuss conditions on subvarieties Λ_i in an effective $\mathbb{Q}$-basis that allow sampling and a membership test. The moving lemma is essential for actual computations.

If $(V, \Lambda_\bullet, W_\bullet)$ is a general witness set for a subvariety $V \subset X$ of dimension k and $V \cap \Lambda_i$ is transverse, then we may move the general witness set W_i using any elementary rational or algebraic equivalence involving Λ_i.

Theorem 11. *Suppose that Λ_i is an effective cycle with $[\Lambda_i] = [L_i^{(n-k)}]$ in C_*X that meets a subvariety V transversally in a general witness set $W_i = V \cap \Lambda_i$. For any elementary rational equivalence $[\Lambda_i] - [\Lambda_i'] \in \mathrm{Rat}_{n-k} X$, homotopy continuation of W_i along this rational equivalence as in Theorem 3 computes a general witness set $W_i' \in C_0(V \cap \Lambda_i')$.*

Proof. Suppose that $Y \subset X \times \mathbb{P}^1$ as in (2) gives an elementary rational equivalence $[\Lambda_i] - [\Lambda_i'] \in \mathrm{Rat}_{n-k} X$ (so that $\Lambda_i = \iota f^{-1}(1)$ and $\Lambda_i' = \iota f^{-1}(0)$). Since V meets Λ_i transversally in W_i, by Theorem 3, there is a homotopy connecting W_i with points on $V \cap \Lambda_i'$. If $V \cap \Lambda_i'$ is transverse, then numerical homotopy continuation may be used to compute the points $W_i' = V \cap \Lambda_i'$. If it is not transverse, so that homotopy paths become singular at $t = 0$, then endgames [1, 15]

may be used to compute the endpoints of the paths and the corresponding multiplicities. These points and multiplicities give a cycle $W_i' \in Z_0(V \cap \Lambda_i')$ representing $[V] \bullet [\Lambda_i']$. □

Remark 12. By Remark 5, an elementary algebraic equivalence also gives a homotopy. ◇

The exceptional divisor E of $X = Bl_p\mathbb{P}^2$ does not move. Thus it may not be possible to move a generator L_i of $C_{n-k}X$ in a rational or algebraic family, and thus move a general witness set W_i for a subvariety V of X. Even when a generator L_i moves, it may not move with sufficient freedom.

While $E \subset Bl_p\mathbb{P}^2$ does not move, the proper transform ℓ of a line moves fairly freely. For any curve $C \subset Bl_p\mathbb{P}^2$ and any smooth point x of C with $x \notin E$, there is a proper transform ℓ' of a line in $\mathbb{P}^2$ (so $[\ell] - [\ell']$ is an elementary rational equivalence) that contains x and meets $C \smallsetminus E$ transversally. This suggests the following definition.

Definition 13. A subvariety Λ of X satisfies the *moving lemma* with respect to a dense open subset U of X if for any subvariety V of X and smooth point $x \in V \cap U$, there is a subvariety Λ' of X containing x with $V \cap \Lambda'$ transverse in U and $[\Lambda] - [\Lambda']$ is an elementary rational equivalence. ◇

Remark 14. Suppose that a member $L_i^{(n-k)}$ of an effective $\mathbb{Q}$-basis for $C_{n-k}X$ satisfies the moving lemma with respect to U. Given a general witness set $W_i = V \cap L_i^{(n-k)}$, the algorithm of Theorem 11 for moving W_i may be used to sample points of $V \cap U$ and test membership in V for points $x \in U$. ◇

It is always possible to choose an effective $\mathbb{Q}$-basis for $C_{n-k}X$ with one member satisfying a generic moving lemma.

Proposition 15. *Let X be a smooth variety and $V \subset X$ any affine open subset. Then there is a dense open subset $U \subset V$ such that for every k with $1 \le k \le n = \dim X$, there exists a subvariety Λ of X of dimension k that satisfies the moving lemma with respect to U.*

Proof. Let $\pi\colon V \to \mathbb{A}^n$ be a finite map given by Noether normalization and $U \subset V$ be the set of points x where $d_x\pi$ is unrammified. Then the inverse images $\pi^{-1}(L)$ of affine k-planes L in $\mathbb{A}^n$ form a family of rationally equivalent subvarieties which satisfy the moving lemma with respect to U. □

4 SCHUBERT WITNESS SETS

While an elementary rational equivalence gives rise to homotopies (Theorem 3), the Chow ring A_*X of cycles on X modulo rational quivalence does not typically satisfy hypotheses which allow general witness sets as in Section 3. Even when A_*X satisfies these hypotheses, a general witness set W_i might not be an effective cycle or it might not be possible to use W_i for sampling or testing membership, even generically on an open subset $U \subset X$.

Nevertheless, for the important class of flag varieties, the theory of witness sets for subvarieties of projective spaces extends optimally. Flag varieties include projective spaces, Grassmannians, and products thereof. The Chow ring of a flag variety has an integer basis of effective Schubert cycles, which are the fundamental classes of Schubert varieties (defined below), and the intersection pairing is nondegenerate. Consequently, subvarieties of X have general witness sets. Also, the intersection matrix M^k (7) is a permutation matrix, implying that the coefficients (6) are positive integers. Finally, each Schubert variety satisfies the moving lemma on the whole flag variety. We expain all this below.

There is a well-known classification of flag varieties [3]. Let G be a semisimple reductive algebraic group, such as $SL_m\mathbb{C}$, a symplectic or complex orthogonal group, or a product of such groups. A flag variety for G is a compact homogeneous space for G. It has the form G/P for P a subgroup of G containing a maximal solvable (Borel) subgroup B of G. The orbit of B (or of any conjugate of B) on G/P gives an algebraic cell decomposition of G/P. Closures of these cells are Schubert varieties whose fundamental cycles give a $\mathbb{Z}$-basis for the Chow ring A_*G/P. This has a detailed combinatorial structure, which may be found in [3] or in [8], the latter for $G = SL_m\mathbb{C}$. We summarize its salient features, which imply that the natural general witness sets for flag varieties—Schubert witness sets—have the optimal properties of classical witness sets. We describe them for the Grassmannian of lines in $\mathbb{P}^4$, and show how to determine a Schubert witness set for the set of lines on a quadric $\mathbb{P}^4$.

A *partially ordered set* (*poset*) is a set S with a binary relation $\le$ that is reflexive, antisymmetric, and transitive. If S has a minimal and a maximal element, and all maximal chains in S have the same length, then S is *ranked*. The rank $\mathrm{rk}(\alpha)$ of an element $\alpha \in S$ is the number of elements below α in any maximal chain containing α and the rank of S is the rank of its maximal element. Write S_k for the set of elements of S of rank k.

We summarize some of the structure of a flag variety. Proofs are found in [3, 8].

Proposition 16. *For a flag variety G/P of dimension n, rational equivalence coincides with numerical equivalence, and we have the following.*

(i) *G/P has an algebraic cell decomposition,*

$$G/P \;=\; \coprod_{\alpha\in S} X_\alpha^\circ\,,$$

where S is a ranked poset of rank n. We have $X_\alpha^\circ \simeq \mathbb{C}^{\mathrm{rk}(\alpha)}$, and if X_α is the Zariski closure of X_α°, then

$$X_\alpha \;=\; \coprod_{\beta\le\alpha} X_\beta^\circ\,.$$

(ii) *We have $A_*G/P = \bigoplus_\beta [X_\beta]\cdot\mathbb{Z}$ and*

$$A_kG/P \;=\; \bigoplus_{\mathrm{rk}(\beta)=k} [X_\beta]\cdot\mathbb{Z}\,,$$

so that $\{[X_\beta] \mid \beta \in S_k\}$ is a $\mathbb{Z}$-basis for A_kG/P.

(iii) *For any subvarieties $Y, Z \subset G/P$, there is a dense open subset $\mathcal{O}$ of G such that $gY \cap Z$ is generically transverse for $g \in \mathcal{O}$.*

(iv) *For every $\beta \in S_k$, there is a $\hat\beta \in S_{n-k}$ and a dense open subset $\mathcal{O}$ of G such that for any $\beta \in S_{n-k}$ and $g \in \mathcal{O}$, the intersection $gX_\alpha \cap X_\beta$ is empty if $\alpha \ne \hat\beta$, and if $\alpha = \hat\beta$, then the intersection is transverse and consists of a single point.*

Remark 17. Part (iii), the moving lemma for subvarieties of G/P, is Kleiman's Bertini Theorem [16]. ◇

Remark 18. When $G/P = \mathbb{P}^n$, $S = [0, n]$ is a chain of length n and X_a is a linear subspace of dimension a.

When $G/P = \mathbb{P}^m \times \mathbb{P}^n$, $S = [0,m] \times [0,n]$ and for $(a,b) \in S$, $X_{(a,b)} = K_a \times L_b$ where $K_a \subset \mathbb{P}^m$ and $L_b \subset \mathbb{P}^n$ are linear subspaces of dimensions a and b, respectively.

We describe the poset S and Schubert varieties for the Grassmannian of lines in $\mathbb{P}^4$ in Subsection 4.1. ◇

A flag variety G/P has general witness sets, as rational and numerical equivalence coincide. Using classes of Schubert varieties for a basis of A_*G/P, we obtain *Schubert witness sets*. For a subvariety $V \subset G/P$ of dimension k, a Schubert witness set $(V, gX_\bullet, W_\bullet)$ has the form $gX_\bullet = (gX_\alpha \mid \alpha \in S_{n-k})$ with $g \in G$ and $W_\bullet = (W_\alpha \mid \alpha \in S_{n-k})$ where

$$W_\alpha \;=\; V \cap gX_\alpha \quad \text{for } \alpha \in S_{n-k} \tag{8}$$

is transverse for all $\alpha \in S_{n-k}$. By (iii) for each α, a general translate gX_α meets V transversally, and we may use the same group element g for every $\alpha \in S_{n-k}$.

By (iv), the intersection matrix $M^{(k)}$ is

$$M^{(k)}_{\alpha,\beta} \;=\; \begin{cases} 1 & \text{if } \alpha = \hat{\beta} \\ 0 & \text{if } \alpha \neq \hat{\beta} \end{cases},$$

and thus

$$[V] \;=\; \sum_{\beta \in S_{n-k}} \deg(W_{\hat{\beta}})[X_\beta]\,. \tag{9}$$

We summarize some properties of Schubert witness sets.

Theorem 19. *Let $V \subset G/P$ be a subvariety of dimension k. Each component W_α of a Schubert witness set $W_\bullet$ is a multiplicity-free cycle. Any component W_α of a Schubert witness set* (8) *may be moved to any other Schubert witness set $W'_\alpha = V \cap hX_\alpha$ along an elementary rational equivalence. A non-zero Schubert witness set W_α may be used to sample points of V and to test membership in V for any point $x \in G/P$.*

Proof. For $\alpha \in S_{n-k}$, $W_\alpha = V \cap gX_\alpha$ (8) is transverse, so W_α is a multiplicity-free cycle. Suppose that $W'_\alpha = V \cap hX_\alpha$ is the αth component of another Schubert witness set for V. Let $\varphi\colon \mathbb{C} \to G$ be a smooth rational map with $\varphi(0) = g$ and $\varphi(1) = h$ (e.g. $\varphi(t) = \psi(t)g$ where ψ is a one-parameter subgroup with $\psi(0) = 1$ and $\psi(1) = hg^{-1}$). Then

$$Y \;=\; \overline{\{(x,t) \mid x \in \varphi(t)X_\alpha\}} \;\subset\; G/P \times \mathbb{P}^1$$

as in (2) with $\imath f^{-1}(0) = gX_\alpha$ and $\imath f^{-1}(1) = hX_\alpha$. By Theorem 3, $(V \times \mathbb{P}^1) \cap Y$ is a homotopy between W_α and W'_α.

Since translates of X_α cover G/P (and thus V), these homotopies may be used to sample points of V, and to test membership in V for any $x \in G/P$ as in Subsection 1.2. □

Remark 20. Property (iv) of Proposition 16, that the Schubert basis is self-dual under the intersection pairing, simplifies the use of general witness sets. A variety with an intersection theory C_* that is finitely generated and has the property that every subvariety V satisfies the moving lemma for $U = X$ is a *duality space* if the basis is self-dual under the intersection pairing as in (*iv*). ◇

General witness sets simplify when X is a duality space. If $\{L_i^{(k)} \mid i = 1, \ldots, b_k\}$ are subvarieties whose cycles form a basis for $C_k X$ and $\{L_i^{(n-k)} \mid i = 1, \ldots, b_{n-k} = b_k\}$ subvarieties representing the dual basis in that

$$\deg\big([L_i^{(k)}] \bullet [L_j^{(n-k)}]\big) \;=\; \begin{cases} 1 & i = j \\ 0 & \text{otherwise} \end{cases}$$

Then if $V \subset X$ has dimension k with general witness sets $W_i = V \cap \Lambda_i$, where the intersection is transverse and $[\Lambda_i] = [L_i^{(n-k)}]$, then

$$[V] \;=\; \sum \deg(W_i) \cdot [L_i^{(k)}]\,. \tag{10}$$

as in (9). ◇

4.1 Schubert witness sets for $\mathbb{G}(1, \mathbb{P}^4)$

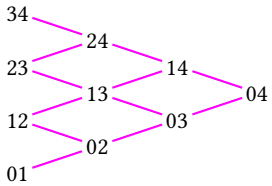
Let $\mathbb{G}(1, \mathbb{P}^4)$ be the Grassmannian of lines in $\mathbb{P}^4$. This is a homogeneous space of dimension 6 for $SL_5\mathbb{C}$. Its Schubert decomposition is in terms of a flag of linear spaces

$$M_\bullet : M_0 \in M_1 \subset M_2 \subset M_3 \subset \mathbb{P}^4\,,$$

where $\dim M_i = i$. Schubert varieties are parametrized by pairs i, j with $0 \le i < j \le 4$. Then $X_{ij} = X_{ij}M_\bullet$ is

$$X_{ij}M_\bullet \;:=\; \{\ell \in \mathbb{G}(1,\mathbb{P}^4) \mid \emptyset \neq \ell \cap M_i, \ell \subset M_j\}\,.$$

The dimension of X_{ij} is $i + j - 1$ and $X_{ij} \subset X_{ab}$ if $i \le a$ and $j \le b$. Duality is given by $\widehat{ij} = 4{-}j, 4{-}i$. We display the partial order S for $\mathbb{G}(1, \mathbb{P}^4)$ below.



Duality is obtained by reflecting in the horizontal line of symmetry with $\widehat{13} = 13$ and $\widehat{04} = 04$.

Let $Q \subset \mathbb{P}^4$ be a smooth quadric which is the zero set of a quadratic polynomial f and let $V_Q \subset \mathbb{G}(1, \mathbb{P}^4)$ be the set of lines that lie on Q. This has codimension 3 in $\mathbb{G}(1, \mathbb{P}^4)$. Indeed, consider the parametrization $\ell(t) = tp + (1-t)q$ of the line through the points $p, q \in \mathbb{P}^4$. Then $f(\ell(t))$ is a quadratic polynomial in t whose coefficients are polynomials in the coordinates of p and q. This line lies on Q when the three coefficients of $f(\ell(t))$ vanish.

A Schubert witness set for V_Q has the form

$$\big(V_Q, (W_{13}, W_{04}), (gX_{13}, gX_{04})\big)\,,$$

where $W_\alpha = V_Q \cap gX_\alpha$ is transverse. Let $M_\bullet$ be the flag in $\mathbb{P}^4$ that defines gX_α. Since

$$X_{04}M_\bullet \;=\; \{\ell \mid M_0 \in \ell\}$$

is the set of lines that contain the point M_0 and $M_0 \notin Q$ (as $M_\bullet$ is general), we have $W_{04} = V_Q \cap X_{04}M_\bullet = \emptyset$. As

$$X_{13}M_\bullet \;=\; \{\ell \mid M_1 \cap \ell \neq \emptyset \text{ and } \ell \subset M_3\}\,,$$

we see that $V_Q \cap X_{13}M_\bullet$ is the set of lines ℓ on $Q \cap M_3$ that meet M_1. Because M_3 is a general $\mathbb{P}^3$, $Q \cap M_3$ is a quadratic hypersurface in $\mathbb{P}^3$. This contains two families of lines, and each point of $Q \cap M_3$

lies on one line from each family. Since M_1 meets Q in two points, $W_{13} = V_Q \cap X_{13}M_\bullet$ consists of four lines. As

$$[V_Q] \;=\; \deg(W_{13})[X_{13}] + \deg(W_{04})[X_{04}] \;=\; 4[X_{13}]\,,$$

if Q' is a second quadric, then

$$[V_Q \cap V_{Q'}] \;=\; [V_Q]^2 \;=\; 16[X_{13}]^2 \;=\; 16[X_{01}]\,,$$

which recovers the well-known fact that 16 lines lie on a general quartic surface $Q \cap Q'$ in $\mathbb{P}^4$.

REFERENCES

[1] D.J. Bates, J.D. Hauenstein, and A.J. Sommese. 2011. A parallel endgame. In *Randomization, relaxation, and complexity in polynomial equation solving*. Contemp. Math., Vol. 556. Amer. Math. Soc., 25–35.

[2] D.J. Bates, J.D. Hauenstein, A.J. Sommese, and C.W. Wampler. 2103. *Numerically Solving Polynomial Systems with Bertini.* SIAM.

[3] M. Brion. 2005. Lectures on the geometry of flag varieties. In *Topics in cohomological studies of algebraic varieties*. Birkhäuser, Basel, 33–85.

[4] S. Di Rocco, D. Eklund, C. Peterson, and A.J. Sommese. 2011. Chern numbers of smooth varieties via homotopy continuation and intersection theory. *J. Symbolic Comput.* 46, 1 (2011), 23–33.

[5] D. Eisenbud and J. Harris (Eds.). 2016. *3264 and All That.* Cambridge University Press.

[6] Wm. Fulton. 1984. *Intersection Theory*. Number 2 in Ergebnisse der Math. Springer-Verlag.

[7] Wm. Fulton. 1984. *Introduction to Intersection Theory in Algebraic Geometry*. AMS.

[8] William Fulton. 1997. *Young tableaux*. London Mathematical Society Student Texts, Vol. 35. Cambridge University Press, Cambridge. x+260 pages.

[9] W. Fulton, R. MacPherson, F. Sottile, and B. Sturmfels. 1995. Intersection theory on spherical varieties. *J. Algebraic Geom.* 4, 1 (1995), 181–193.

[10] Marc Giusti and Joos Heintz. 1993. La détermination des points isolés et de la dimension d'une variété algébrique peut se faire en temps polynomial. In *Computational algebraic geometry and commutative algebra (Cortona, 1991)*. Cambridge Univ. Press, 216–256.

[11] J.D. Hauenstein, A. Leykin, J.I. Rodriguez, and F. Sottile. 2019. A numerical toolkit for multiprojective varieties. `ArXiv.org/1908.00899`.

[12] J.D. Hauenstein and J.I. Rodriguez. 2020. Numerical irreducible decomposition of multiprojective varieties. Advances in Geometry, to appear.

[13] J.D. Hauenstein and A.J. Sommese. 2010. Witness sets of projections. *Appl. Math. Comput.* 217, 7 (2010), 3349–3354.

[14] J.D. Hauenstein and F. Sottile. 2012. Algorithm 921: alphaCertified: certifying solutions to polynomial systems. *ACM Trans. Math. Software* 38, 4 (2012), Art. ID 28, 20.

[15] B. Huber and J. Verschelde. 1998. Polyhedral end games for polynomial continuation. *Numer. Algorithms* 18, 1 (1998), 91–108.

[16] S.L. Kleiman. 1974. The transversality of a general translate. *Compositio Math.* 28 (1974), 287–297.

[17] S.L. Kleiman. 1994. The standard conjectures. In *Motives (Seattle, WA, 1991)*. Proc. Sympos. Pure Math., Vol. 55. Amer. Math. Soc., Providence, RI, 3–20.

[18] R. Krawczyk. 1969. Newton-algorithmen zur bestimmung von nullstellen mit fehler-schranken. *Computing* 4, 3 (1969), 187–201.

[19] A. Leykin, J.I. Rodriguez, and F. Sottile. 2018. Trace test. *Arnold Math. J.* 4, 1 (2018), 113–125.

[20] R.E. Moore and S.T. Jones. 1977. Safe starting regions for iterative methods. *SIAM J. Numer. Anal.* 14, 6 (1977), 1051–1065.

[21] R.E. Moore, R.B. Kearfott, and M.J. Cloud. 2009. *Introduction to interval analysis*. SIAM. xii+223 pages.

[22] S. Smale. 1986. Newton's method estimates from data at one point. In *The merging of disciplines: new directions in pure, applied, and computational mathematics (Laramie, Wyo., 1985)*. Springer, New York, 185–196.

[23] A.J. Sommese, J. Verschelde, and C.W. Wampler. 2002. Symmetric functions applied to decomposing solution sets of polynomial systems. *SIAM J. Numer. Anal.* 40, 6 (2002), 2026–2046 (2003).

[24] A.J. Sommese, J. Verschelde, and C.W. Wampler. 2005. Introduction to numerical algebraic geometry. In *Solving polynomial equations*. Algorithms Comput. Math., Vol. 14. Springer, Berlin, 301–335.

[25] A.J. Sommese and C.W. Wampler. 1996. Numerical algebraic geometry. In *The mathematics of numerical analysis*. Lectures in Appl. Math., Vol. 32. Amer. Math. Soc., 749–763.

[26] A.J. Sommese and C.W. Wampler. 2005. *The numerical solution of systems of polynomials*. World Scientific.

[27] A.J. Sommese and C.W. Wampler. 2008. Exceptional sets and fiber products. *Found. Comput. Math.* 8, 2 (2008), 171–196.

[28] A. Weil. 1962. *Foundations of algebraic geometry*. Amer. Math. Soc. xx+363 pages.

Parametric Standard System for Mixed Module and its Application to Singularity Theory

Hiroshi Teramoto*
teramoto@es.hokudai.ac.jp
Hokkaido University/Institute for Chemical Reaction Design and Discover
Sapporo, Japan
PRESTO, Department of Research Promotion
Tokyo, Japan

Katsusuke Nabeshima
Tokushima University
Tokushima, Japan
nabeshima@tokushima-u.ac.jp

ABSTRACT

We provide a concrete computational algorithm for computing the standard basis for a mixed module proposed by Gatermann and Hosten [1]. We extend it to parametric standard system for a mixed module and provide an algorithm to compute it. We demonstrate our algorithm by applying it to classification of map-germs relative to $\mathcal{A}$ in which complicated moduli structures appear.

CCS CONCEPTS

• **Computing methodologies** → **Algebraic algorithms**.

KEYWORDS

mixed-module, comprehensive standard basis, singularity theory

ACM Reference Format:
Hiroshi Teramoto and Katsusuke Nabeshima. 2020. Parametric Standard System for Mixed Module and its Application to Singularity Theory. In *International Symposium on Symbolic and Algebraic Computation (ISSAC '20), July 20–23, 2020, Kalamata, Greece*. ACM, New York, NY, USA, 8 pages. https://doi.org/10.1145/3373207.3404027

1 INTRODUCTION

Mixed modules are sums of two modules over two different rings. Mixed modules appear in various settings such as Stanley decomposition of a quotient ideal and singularity theory. In singularity theory, mixed modules appear in classification of map-germs relative to various equivalence relations such as $\mathcal{A}$ [2], $\mathcal{K}_{\mathcal{B}}$ [3, 4], and $\mathcal{A}[G]$-equivalence [5] for some Lie group G. There, the concept of (extended) tangent space plays an important role and an (extended) tangent space is a mixed module relative to these equivalences. Compared to the conventional module over a single ring, the algebraic structure of a mixed module can be highly complicated, which makes classification of map-germs relative to these equivalences difficult. This is thought of as one of the motivations of Mather [6] to reduce classification of stable map-germs relative to $\mathcal{A}$ to that of those relative to $\mathcal{K}$ since, in the latter case, (extended) tangent spaces of map-germs are modules.

One of the pioneering works for automation of classification relative to these equivalences is done by Kirk [7–9] based on the complete transversal theorem [10, 11]. Unlike the conventional module where efficient computation can be done by using the standard basis, there was no such a concept for mixed module at that time. In their algorithm, they handle mixed modules in jet spaces as a huge vector space over $\mathbb{R}$. It seems that their software is no longer available and it is difficult to assess the efficiency of their algorithm but it can be made much more efficient if mixed module structures are taken into account.

Since then, a possible generalization of standard bases for mixed modules is proposed by Gatermann and Hosten [1] and it is applied to solve classification of map-germs relative to $\mathcal{K}_{\mathcal{B}}$. In this paper, we extend it to parametric standard system for a mixed module (comprehensive standard system (CSS) for a mixed module), propose a computational algorithm (Algorithms 2-4) for it, and apply the algorithm to solve classification problems involving complicated moduli structure.

In Sec. 2, we review standard basis for a mixed module by [1] and introduce a concrete computational algorithm (Algorithm 1) for it. In Sec. 3, we extend it to CSS for a mixed module along with an example to demonstrate it. In Sec. 4, we demonstrate CSS for a mixed module in classification of map-germs relative to $\mathcal{A}$. In Sec. 5, we provide conclusions and remarks.

2 STANDARD BASIS FOR MIXED MODULE [1]

Let K be a field and let $\lambda = (\lambda_1, \cdots, \lambda_{n_\lambda})$ and $x = (x_1, \cdots, x_{n_x})$ be variables such that they are disjoint with each other. Let $K[x, \lambda]$ be the polynomial ring with variables x and λ, $\langle x, \lambda \rangle$ be the ideal generated by x and λ, and $K[x, \lambda]_{\langle x, \lambda \rangle}$ be the localization of $K[x, \lambda]$ with respect to $\langle x, \lambda \rangle$.

ISSAC '20, July 20–23, 2020, Kalamata, Greece

ACM ISBN 978-1-4503-7100-1/20/07…$15.00
https://doi.org/10.1145/3373207.3404027

Definition 2.1. *A (x,λ)-mixed module $M \subset \left(K[x,\lambda]_{\langle x,\lambda\rangle}\right)^n$ is a $K[\lambda]_{\langle\lambda\rangle}$-module which may be written as a sum $M = N + Q$, where $N \subset \left(K[x,\lambda]_{\langle x,\lambda\rangle}\right)^n$ is a $K[x,\lambda]_{\langle x,\lambda\rangle}$-module of finite codimension as a K-vector space in $\left(K[x,\lambda]_{\langle x,\lambda\rangle}\right)^n$ and $Q \subset \left(K[x,\lambda]_{\langle x,\lambda\rangle}\right)^n$ is a $K[\lambda]_{\langle\lambda\rangle}$-module.*

Let $\prec_{x,\lambda}$ be a local ordering in the set of monomials in x and λ. Let $\prec_{x,\lambda,m}$ be a module ordering in the monomials in $\left(K[x,\lambda]_{\langle x,\lambda\rangle}\right)^n$ which is compatible with the ordering $\prec_{x,\lambda}$, i.e., a module ordering satisfying:

(1) $x^\alpha\lambda^\beta e_i \prec_{x,\lambda,m} x^{\alpha'}\lambda^{\beta'}e_j$
$\Rightarrow x^{\alpha+\alpha''}\lambda^{\beta+\beta''}e_i \prec_{x,\lambda,m} x^{\alpha'+\alpha''}\lambda^{\beta'+\beta''}e_j$,
(2) $x^\alpha\lambda^\beta \prec_{x,\lambda} x^{\alpha'}\lambda^{\beta'} \Rightarrow x^\alpha\lambda^\beta e_i \prec_{x,\lambda,m} x^{\alpha'}\lambda^{\beta'}e_i$,

for all $\alpha,\alpha',\alpha'' \in \mathbb{Z}_{\geq 0}^{n_x}$, $\beta,\beta',\beta'' \in \mathbb{Z}_{\geq 0}^{n_\lambda}$, and $i,j \in \{1,\cdots,n\}$, where $e_i = (0,\cdots,0,1,0,\cdots,0)$ is the i-th canonical basis vector of $\left(K[x,\lambda]_{\langle x,\lambda\rangle}\right)^n$ with 1 at the iâĂŞth place. Let $\mathrm{LM}_{\prec_{x,\lambda,m}}(f)$, $\mathrm{LT}_{\prec_{x,\lambda,m}}(f)$, and $\mathrm{LC}_{\prec_{x,\lambda,m}}(f)$ be the leading monomial, leading term and leading coefficient of $f \in \left(K[x,\lambda]_{\langle x,\lambda\rangle}\right)^n$, respectively.

Definition 2.2 (Initial Module). *The initial module $in_{\prec_{x,\lambda,m}}(M)$ is defined as the $K[\lambda]_{\langle\lambda\rangle}$-module*

$$in_{\prec_{x,\lambda,m}}(M) = \langle \mathrm{LM}_{\prec_{x,\lambda,m}}(f) \,|\, \forall g \in K[x,\lambda]_{\langle x,\lambda\rangle}, gf \in M \rangle_{K[x,\lambda]_{\langle x,\lambda\rangle}} + \langle \mathrm{LM}_{\prec_{x,\lambda,m}}(f) \,|\, f \in M \rangle_{K[\lambda]_{\langle\lambda\rangle}}.$$

Definition 2.3 ((x,λ)-mixed standard basis). *An (x,λ)-mixed standard basis of M is a pair $\left(S^{(1)},S^{(2)}\right)$ of two finite sets $S^{(1)}$ and $S^{(2)}$ such that*

$$M = \langle S^{(1)}\rangle_{K[x,\lambda]_{\langle x,\lambda\rangle}} + \langle S^{(2)}\rangle_{K[\lambda]_{\langle\lambda\rangle}}$$

and

$$in_{\prec_{x,\lambda,m}}(M) = \langle \mathrm{LM}_{\prec_{x,\lambda,m}}\left(S^{(1)}\right)\rangle_{K[x,\lambda]_{\langle x,\lambda\rangle}} + \langle \mathrm{LM}_{\prec_{x,\lambda,m}}\left(S^{(2)}\right)\rangle_{K[\lambda]_{\langle\lambda\rangle}}.$$

Lemma 37 in [1] guarantees the existence of an (x,λ)-mixed standard basis with respect to an arbitrary local order $\prec_{x,\lambda}$. A brief sketch of the algorithm for computing standard basis for a given pair of finite number of generators in N and Q is given in [1]. Here, we provide a concrete algorithm for computing a pair $\left(S^{(1)},S^{(2)}\right)$ for a given pair of finite number of generators in N and Q. We define the S-polynomial $\mathrm{spoly}(f,g)$ for non-zero $f,g \in K[x,\lambda]^n$ as follows: Suppose $\mathrm{LM}_{\prec_{x,\lambda,m}}(f) = x^\alpha\lambda^\beta e_i$ and $\mathrm{LM}_{\prec_{x,\lambda,m}}(g) = x^{\alpha'}\lambda^{\beta'}e_j$ ($\alpha,\alpha' \in \mathbb{Z}_{\geq 0}^{n_x}$, $\beta,\beta' \in \mathbb{Z}_{\geq 0}^{n_\lambda}$ and $i,j \in \{1,\cdots,n\}$). The S-polynomial $\mathrm{spoly}(f,g)$ is defined as

$$\mathrm{LCM}\left(x^\alpha\lambda^\beta, x^{\alpha'}\lambda^{\beta'}\right)\left(\frac{f}{\mathrm{LC}_{\prec_{x,\lambda}}(f)\,x^\alpha\lambda^\beta} - \frac{g}{\mathrm{LC}_{\prec_{x,\lambda}}(g)\,x^{\alpha'}\lambda^{\beta'}}\right) \tag{1}$$

if $i=j$ and 0 in the other cases, where $x^\alpha = x_1^{\alpha_1}x_2^{\alpha_2}\cdots x_{n_x}^{\alpha_{n_x}}$ and $\lambda^\beta = \lambda_1^{\beta_1}\lambda_2^{\beta_2}\cdots\lambda_{n_\lambda}^{\beta_{n_\lambda}}$. For $\alpha,\alpha' \in \mathbb{Z}_{\geq 0}^{n_x}$, we write $\alpha \leq \alpha'$ if $\alpha_i \leq \alpha_i'$ holds for all $i = 1,\cdots,n_x$.

Algorithm 1. *Compute Standard Basis for Mixed Module*

Input: $N,Q \subset K[x,\lambda]^n$: finite sets of generators of the mixed module $\langle N\rangle_{K[x,\lambda]_{\langle x,\lambda\rangle}} + \langle Q\rangle_{K[\lambda]_{\langle\lambda\rangle}}$
Output: $\left(S^{(1)},S^{(2)}\right)$: standard basis

1: $S^{(1)} \leftarrow$ the reduced standard basis of N;
2: $S^{(2)} \leftarrow$ the non-zero reduced normal forms of the elements of Q with respect to $S^{(1)}$;
3: $P_1 \leftarrow \left\{ \mathrm{spoly}(f,g) \,\middle|\, \begin{array}{c} f \in S^{(1)}, g \in S^{(2)}, i=j \text{ and } \alpha \leq \alpha' \\ \mathrm{LM}_{\prec_{x,\lambda,m}}(f) = x^\alpha\lambda^\beta e_i, \\ \text{and } \mathrm{LM}_{\prec_{x,\lambda,m}}(g) = x^{\alpha'}\lambda^{\beta'}e_j \end{array} \right\}$;
4: $P_2 \leftarrow \left\{ \mathrm{spoly}(f,g) \,\middle|\, \begin{array}{c} f \in S^{(2)}, g \in S^{(2)}, i=j \text{ and } \alpha = \alpha' \\ \mathrm{LM}_{\prec_{x,\lambda,m}}(f) = x^\alpha\lambda^\beta e_i, \\ \text{and } \mathrm{LM}_{\prec_{x,\lambda,m}}(g) = x^{\alpha'}\lambda^{\beta'}e_j \end{array} \right\}$;
5: $P = P_1 \cup P_2$;
6: **while** $P \neq \emptyset$ **do**
7: $f \leftarrow$ one of the elements in P;
8: $P \leftarrow P \setminus \{f\}$;
9: $f \leftarrow$ the reduced normal form of f in **Algorithm 32** in [1] with respect to $\left(S^{(1)},S^{(2)}\right)$;
10: **if** $f \neq 0$ $\left(\mathrm{LM}_{\prec_{x,\lambda,m}}(f) = x^\alpha\lambda^\beta e_i\right)$ **then**
11: $P \leftarrow P \cup \left\{ \mathrm{spoly}(f,g) \,\middle|\, \begin{array}{c} g \in S^{(1)}, i=j \text{ and } \alpha \geq \alpha', \\ \mathrm{LM}_{\prec_{x,\lambda,m}}(g) = x^{\alpha'}\lambda^{\beta'}e_j \end{array} \right\}$;
12: $P \leftarrow P \cup \left\{ \mathrm{spoly}(f,g) \,\middle|\, \begin{array}{c} g \in S^{(2)}, i=j \text{ and } \alpha = \alpha', \\ \mathrm{LM}_{\prec_{x,\lambda,m}}(g) = x^{\alpha'}\lambda^{\beta'}e_j \end{array} \right\}$;
13: $S^{(2)} \leftarrow S^{(2)} \cup \{f\}$;
14: **end if**
15: **end while**

end

Algorithm 1 terminates in finite steps. It can be shown as follows. The set of monomials in $\left(k[x,\lambda]_{\langle x,\lambda\rangle}\right)^n$ that do not belong to $\langle S^{(1)}\rangle_{K[x,\lambda]_{\langle x,\lambda\rangle}}$ is finite since the codimension of N in $\left(k[x,\lambda]_{\langle x,\lambda\rangle}\right)^n$ is finite. Let $\mathrm{Mon}_{\mathrm{can}}$ be the set of finite number of monomials which are not divisible by any element of $\mathrm{LM}_{\prec_{x,\lambda}}\left(S^{(1)}\right)$. By Algorithm 1, in the while loop (6-15th line), it is guaranteed that $\mathrm{LM}\left(S^{(2)}\right) \subset \mathrm{Mon}_{\mathrm{can}}$ holds. It is because the set $S^{(2)}$ is initiated in line 2 of Algorithm 1 as the non-zero reduced normal forms of the elements of Q with respect to $S^{(1)}$ and all the new elements added to $S^{(2)}$ are non-zero reduced normals forms in **Algorithm 32** in [1] with respect to $\left(S^{(1)},S^{(2)}\right)$, whose leading terms are not multiples of any $\mathrm{LM}_{\prec_{x,\lambda}}\left(S^{(1)}\right)$ by **Lemma 33** in [1]. Every time a non-zero element f is found in the while loop, the number of elements in $\mathrm{LM}\left(S^{(2)}\right)$ increases by one. Since the number of elements in $\mathrm{Mon}_{\mathrm{can}}$ is finite, the number of elements in $\mathrm{LM}_{\prec_{x,\lambda,m}}\left(S^{(2)}\right)$ is saturated in finite steps. After the step, a non-zero element f cannot appear in the while loop and thus the number of elements in P decreases by one in each iteration of the while loop. Since the number of elements in P is finite, the iteration terminates in finite steps.

Theorem 41 in [1] guarantees the output of Algorithm 1, i.e., $\left(S^{(1)},S^{(2)}\right)$ is a standard basis of the mixed module $M = N + Q$.

In summary, we get the following.

THEOREM 2.1. *For a given finite set of generators $N, Q \subset K[x,\lambda]^n$, Algorithm 1 terminates in finite steps and outputs an (x,λ)-mixed standard basis $\left(S^{(1)}, S^{(2)}\right)$ of*

$$\langle N\rangle_{K[x,\lambda]_{\langle x,\lambda\rangle}} + \langle Q\rangle_{K[\lambda]_{\langle\lambda\rangle}}.$$

2.1 Example

In this example, we compute the $\mathcal{A}$-codimension of a map-germ $f\colon \left(\mathbb{R}^2,0\right) \to \left(\mathbb{R}^2,0\right)$, defined as

$$(x_1,x_2) \mapsto \left(y_1 = x_1, y_2 = x_1x_2 + x_2^5 + x_2^7\right),$$

which is Type 6 in [12], by using a mixed standard basis. In this example and in the forthcoming examples, we use the variables (x,y) instead of (x,λ) as in [1] since in this context y is supposed to be a coordinate in the target space of the map-germ and it is not common to use λ for that.

Let $\mathcal{E}_n$ be the set of function-germs $f\colon (\mathbb{R}^n,0) \to \mathbb{R}$, $\mathcal{M}_n$ be its maximal ideal, and $\mathcal{M}_n^k$ for $k \in \mathbb{N}$ is recursively defined as: $\mathcal{M}_n^1 = \mathcal{M}_n$ and $\mathcal{M}_n^{k+1} = \mathcal{M}_n \cdot \mathcal{M}_n^k$. Let the tangent space of f relative to $\mathcal{A}$ be

$$T\mathcal{A}(f) = \mathcal{M}_2\langle (1,x_2), \left(0, x_1 + 5x_2^4 + 7x_2^6\right)\rangle_{\mathcal{E}_2} + f^*\langle \mathcal{M}_2\mathcal{E}_2^2\rangle_{f^*\mathcal{E}_2},$$

where $f^*\mathcal{E}_2 = \{\eta \circ f \,|\eta \in \mathcal{E}_2\}$ and

$$\begin{aligned} f^*\langle \mathcal{M}_2\mathcal{E}_2^2\rangle_{f^*\mathcal{E}_2} &= f^*\langle (y_1,0),(y_2,0),(0,y_1),(0,y_2)\rangle_{f^*\mathcal{E}_2} \\ &= \langle (f_1(x),0),(f_2(x),0),(0,f_1(x)),(0,f_2(x))\rangle_{f^*\mathcal{E}_2}. \end{aligned}$$

Since f is 7-$\mathcal{A}$-determined [12],

$$\frac{\mathcal{M}_2\mathcal{E}_2^2}{T\mathcal{A}(f)} \cong \frac{\mathcal{M}_2\mathcal{E}_2^2}{T\mathcal{A}(f) + \mathcal{M}_n^8\mathcal{E}_2^2}$$

which is isomorphic to $\dfrac{\left(\langle x_1,x_2\rangle \mathbb{R}[x,y]_{\langle x,y\rangle}\right)^2}{M}$, as an $\mathbb{R}$-vector space where M is an (x,y)-mixed module with

$$\begin{aligned} N = \langle x_1,x_2\rangle \cdot \langle \frac{\partial f}{\partial x_1}, \frac{\partial f}{\partial x_2}\rangle_{\mathbb{R}[x,y]_{\langle x,y\rangle}} \\ + \langle y_1 - f_1(x), y_2 - f_2(x)\rangle \cdot \left(\mathbb{R}[x,y]_{\langle x,y\rangle}\right)^2 \\ + \langle x_1,x_2\rangle^8 \cdot \left(\mathbb{R}[x,y]_{\langle x,y\rangle}\right)^2 \end{aligned} \quad (2)$$

and

$$Q = \langle y_1,y_2\rangle \cdot \left(\mathbb{R}[x,y]_{\langle x,y\rangle}\right)^2. \quad (3)$$

By computing an (x,y)-mixed standard basis of M, we can get the $\mathcal{A}$-codimension of f.

In this example, we use the following module ordering:

$$x_1^{\alpha_1}x_2^{\alpha_2}y_1^{\beta_1}y_2^{\beta_2}e_i \prec x_1^{\alpha_1'}x_2^{\alpha_2'}y_1^{\beta_1'}y_2^{\beta_2'}e_j$$

iff one of the following holds:

1. $\alpha_1+\alpha_2+\beta_1+\beta_2 > \alpha_1'+\alpha_2'+\beta_1'+\beta_2'$
2. $\alpha_1+\alpha_2+\beta_1+\beta_2 = \alpha_1'+\alpha_2'+\beta_1'+\beta_2'$ and $\beta_1 < \beta_1'$
3. $\alpha_1+\alpha_2+\beta_1+\beta_2 = \alpha_1'+\alpha_2'+\beta_1'+\beta_2'$ and $\beta_1 = \beta_1'$ and $\beta_2 < \beta_2'$
4. $\alpha_1+\alpha_2+\beta_1+\beta_2 = \alpha_1'+\alpha_2'+\beta_1'+\beta_2'$ and $\beta_1 = \beta_1'$ and $\beta_2 = \beta_2'$ and $\alpha_1 < \alpha_1'$
5. $\alpha = \alpha'$ and $\beta = \beta'$ and $i > j$.

For the module ordering, an (x,y)-mixed standard basis $\left(S^{(1)}, S^{(2)}\right)$ of $N+Q$ is computed by Algorithm 1 implemented in SINGULAR [13]:

$$\begin{aligned} S^{(1)} = \{ &\left(0, y_2 + 4y_2^5 + 6x_2^7\right), \left(y_2, -4y_2^6 + 6x_2^8\right), (0, y_1 - x_2), \\ &\left(y_1, -5x_2^5 - 7x_2^7\right), \left(x_2, x_2^2\right), \left(x_1, -5x_2^5 - 7x_2^7\right), \\ &\left(0, x_1x_2 + 5x_2^5 + 7x_2^7\right), \left(0, x_1^2 - 25x_2^8\right), \left(x_2^9, 0\right)\}, \end{aligned}$$

$$\begin{aligned} S^{(2)} = \{ &\left(0, x_2^5 + 7/5x_2^7\right), (0, x_1), \left(0, x_2^6 + 3/2x_2^8\right), \\ &\left(0, x_2^5 + 3/2x_2^7\right), \left(0, x_2^7\right), \left(0, x_2^8\right)\}. \end{aligned}$$

The quotient vector space $\dfrac{\left(\langle x_1,x_2\rangle \mathbb{R}[x,y]_{\langle x,y\rangle}\right)^2}{M}$ is spanned by monomials in $\left(\langle x_1,x_2\rangle \mathbb{R}[x,y]_{\langle x,y\rangle}\right)^2$ that are neither multiples of $\mathrm{LM}_{\prec_{x,\lambda,m}}(f)$ for $f \in S^{(1)}$ nor involutive multiples of $\mathrm{LM}_{\prec_{x,\lambda,m}}(f)$ for $f \in S^{(2)}$, where a monomial $x^\alpha y^\beta e_i$ is an involutive multiple of $x^{\alpha'}y^{\beta'}e_j$ if $i=j$, $\alpha=\alpha'$ and $\beta \geq \beta'$, i.e., $\beta_i \geq \beta_i'$ for all $i = 1,\cdots,n_y$. In this case,

$$\frac{\mathcal{M}_2\mathcal{E}_2^2}{T\mathcal{A}(f)} \cong \langle (0,x_2), \left(0,x_2^2\right), \left(0,x_2^3\right), \left(0,x_2^4\right)\rangle_{\mathbb{R}}$$

and the $\mathcal{A}$-codimension of f is 4, which coincides with the result in Table 1 in [12].

In the next section, we extend the result to that of mixed modules with parameters.

3 COMPREHENSIVE STANDARD SYSTEM FOR MIXED MODULE

Let $\mathbb{K}$ be a field and let $\lambda = (\lambda_1,\cdots,\lambda_{n_\lambda})$, $a = (a_1,\cdots,a_{n_a})$, and $x = (x_1,\cdots,x_{n_x})$ be variables such that they are disjoint with each other. Let $\mathbb{K}[a][x,\lambda]$ be the polynomial ring with variables x, λ, and a. Let K be the algebraic closure of $\mathbb{K}$. Let $t = (t_1,\cdots,t_{n_a}) \in K^{n_a}$ and $\sigma_t\colon \mathbb{K}[a][x,\lambda] \to K[x,\lambda]$ be a specialization morphism defined as $\sigma_t(f) = f|_{a=t}$. In the same manner, we define $\sigma_t(f)$ for $f \in \mathbb{K}(a)[x,\lambda]$ and $t \in K^{n_a}$ if the denominators of all the coefficients of the terms of f are specialized to be nonzero at t. Let $V(E) = \{t \in K^{n_a} \,|\forall h \in E, h(t) = 0\}$ be an affin algebraic set of an ideal $E \subset \mathbb{K}[a]$.

DEFINITION 3.1 (COMPREHENSIVE STANDARD SYSTEM FOR MIXED MODULE). *Let $N, Q \subset \mathbb{K}[a][x,\lambda]^n$ be finite sets such that $\langle \sigma_t(N)\rangle_{K[x,\lambda]_{\langle x,\lambda\rangle}}$ has a finite codimension as a K-vector space in $\left(K[x,\lambda]_{\langle x,\lambda\rangle}\right)^n$ for all $t \in V$. Let $S_i^{(1)}, S_i^{(2)} \subset \mathbb{K}[a][x,\lambda]^n$ be a finite subset, and $(E_i, N_i) \subset \mathbb{K}[a] \times \mathbb{K}[a]$ for $i = 1,\cdots,\ell$. The triple set $G = \left\{\left(E_i, N_i, \left(S_i^{(1)}, S_i^{(2)}\right)\right)\right\}_{i=1,\cdots,\ell}$ is called Comprehensive Standard System (CSS) for N, Q with respect to $\prec_{x,\lambda,m}$ over $V \subset K^{n_a}$ if the following conditions hold:*

1. $V \subset \bigcup_{i=1}^{\ell} V(E_i) \setminus V(N_i)$.
2. *For any $t \in V$ and $i \in \{1,\cdots,\ell\}$ such that $t \in V(E_i) \setminus V(N_i)$ holds, the pair $\left(\sigma_t\left(S_i^{(1)}\right), \sigma_t\left(S_i^{(2)}\right)\right)$*

is an (x,λ)-mixed standard basis of $\langle\sigma_t(N)\rangle_{K[x,\lambda]_{\langle x,\lambda\rangle}}+\langle\sigma_t(Q)\rangle_{K[\lambda]_{\langle\lambda\rangle}}$.

Like in Algorithm 1, the first step is to compute the comprehensive standard basis of N. By Definition 3.1, for any $t\in K^{n_a}$, the specialization $\sigma_t(N)$ is of finite codimension in $\left(K[x,\lambda]_{\langle x,\lambda\rangle}\right)^n$, the comprehensive standard basis of N can be computed by using the algebraic local cohomology (ALC) [14, 15]. Another algorithm such as [16] can be used for that purpose but there is at least one benefit to using ALC in this part, that is, reduction by ALC does not require any division algorithm and can be made quite efficient. In Algorithm 3, reduction by a standard basis of N occurs many times and this part can be made quite efficient if ALC is used. In our implementation, we implemented ALC for finite-codimension modules with parameters in Singular.

In what follows, we provide our algorithm to compute CSS for given pairs of finite generators N and Q in $\mathbb{K}[a][x,\lambda]^n$.

Algorithm 2. *Compute* CSS

Input: $N,Q\subset\mathbb{K}[a][x,\lambda]^n$, $E_{in},N_{in}\subset\mathbb{K}[a]$
Output: G : CSS on $V(E_{in})\setminus V(N_{in})$
1: $G\leftarrow\emptyset$;
2: $\left\{\left(E_i,N_i,S_i^{(1)}\right)\right\}_{i=1,\cdots,\ell'}\leftarrow$ comprehensive standard system of N on $V(E_{in})\setminus V(N_{in})$;
3: **for** $i\in\{1,\cdots,\ell'\}$ **do**
4: $\left\{\left(E_{ij},N_{ij},S_i^{(1)},S_{ij}^{(2)}\right)\right\}_{j=1,\cdots,\ell''}$ $\leftarrow$ CSSMain $\left(E_i,N_i,S_i^{(1)},Q\right)$;(See Algorithm 3)
5: $G\leftarrow G\cup\left\{\left(E_{ij},N_{ij},S_i^{(1)},S_{ij}^{(2)}\right)\right\}_{j=1,\cdots,\ell''}$;
6: **end for**
end

Algorithm 3. CSSMain $\left(E_i,N_i,S_i^{(1)},Q\right)$

Input: $E_i,N_i\subset\mathbb{K}[a]$, $S_i^{(1)},Q\subset\mathbb{K}[a][x,\lambda]^n$
Output: G : CSS on $V(E_i)\setminus V(N_i)$
1: $G\leftarrow\emptyset$;
2: $Q\leftarrow$ the reduced normal form of Q in terms of $S_i^{(1)}$ in $\left(\mathbb{K}(a)[x,\lambda]_{\langle x,\lambda\rangle}\right)^n$, keep non-zero elements only and multiply each non-zero element to the least common multiple of the demominators of the coefficients of its terms in $\mathbb{K}[a]$;
3: $S^{(1)}\leftarrow S_i^{(1)}$;
4: $S^{(2)}\leftarrow$ the reduced normal form of Q in terms of $E_i\mathbb{K}[a][x,\lambda]^n$, keep non-zero elements only;
5: $h\leftarrow$ the square-free part of LCM $\left(\mathrm{LC}_{\prec_{x,\lambda,m}}\left(S^{(2)}\right)\right)$;
6: $\left(h_1,\cdots,h_{n_f}\right)\leftarrow$ the irreducible factors of h;
7: $G\leftarrow G\cup\bigcup_{j=1}^{n_f}$ CSSMain $\left(E_i+\langle h_j\rangle,\left(\prod_{l=1}^{j-1}h_l\right)N_i,S^{(1)},S^{(2)}\right)$; *
8: $P_1\leftarrow\left\{\mathrm{spoly}(f,g)\middle|\begin{array}{c}f\in S^{(1)},g\in S^{(2)},i=j\text{ and }\alpha\le\alpha',\\ \mathrm{LM}_{\prec_{x,\lambda,m}}(f)=x^\alpha\lambda^\beta e_i,\\ \text{and }\mathrm{LM}_{\prec_{x,\lambda,m}}(g)=x^{\alpha'}\lambda^{\beta'}e_j\end{array}\right\}$; †
9: $P_2\leftarrow\left\{\mathrm{spoly}(f,g)\middle|\begin{array}{c}f\in S^{(2)},g\in S^{(2)},i=j\text{ and }\alpha=\alpha',\\ \mathrm{LM}_{\prec_{x,\lambda,m}}(f)=x^\alpha\lambda^\beta e_i,\\ \text{and }\mathrm{LM}_{\prec_{x,\lambda,m}}(g)=x^{\alpha'}\lambda^{\beta'}e_j\end{array}\right\}$;
10: $P\leftarrow P_1\cup P_2$;
11: $G\leftarrow G\cup\mathrm{CSSSub}\left(E_i,hN_i,S_i^{(1)},S_i^{(2)},P\right)$; (See Algorithm 4)
12: **return**;
end

Algorithm 4. CSSSub $\left(E_i,N_i,S^{(1)},S^{(2)},P\right)$

Input: $E_i,N_i\subset\mathbb{K}[a]$, $S^{(1)},S^{(2)}\subset\mathbb{K}[a][x,\lambda]^n$, $P\subset\mathbb{K}(a)[x,\lambda]^n$
Output: G : CSS on $V(E_i)\setminus V(N_i)$
1: $G\leftarrow\emptyset$;
2: **while** $P\neq\emptyset$ and $N_i\not\subset\sqrt{E_i}$ **do**
3: $f\leftarrow$ one of the elements in P;
4: $P\leftarrow P\setminus\{f\}$;
5: $f\leftarrow$ the reduced normal form of f in **Algorithm 32** in [1] with respect to $\left(S^{(1)},S^{(2)}\right)$ in $\mathbb{K}(a)[x,\lambda]^n$ multiplied by the least common multiple of the denominators of the coefficients of all the terms of f so that $f\in\mathbb{K}[a][x,\lambda]$ holds;
6: $f\leftarrow$ the reduced normal form of f in terms of $E_i\mathbb{K}[a][x,\lambda]^n$;
7: **while** $f\neq0$ **do**
8: $P_1\leftarrow\left\{\mathrm{spoly}(f,g)\middle|\begin{array}{c}g\in S^{(1)},i=j\text{ and }\alpha\ge\alpha',\\ \mathrm{LM}_{\prec_{x,\lambda,m}}(f)=x^\alpha\lambda^\beta e_i,\\ \text{and }\mathrm{LM}_{\prec_{x,\lambda,m}}(g)=x^{\alpha'}\lambda^{\beta'}e_j\end{array}\right\}$;
9: $P_2\leftarrow\left\{\mathrm{spoly}(f,g)\middle|\begin{array}{c}g\in S^{(2)},i=j\text{ and }\alpha=\alpha',\\ \mathrm{LM}_{\prec_{x,\lambda,m}}(f)=x^\alpha\lambda^\beta e_i,\\ \text{and }\mathrm{LM}_{\prec_{x,\lambda,m}}(g)=x^{\alpha'}\lambda^{\beta'}e_j\end{array}\right\}$;
10: $P'\leftarrow P\cup P_1\cup P_2$;
11: $G\leftarrow G\cup$ CSSSub $\left(E_i,\mathrm{LC}_{\prec_{x,\lambda,m}}(f)N_i,S^{(1)},S^{(2)}\cup\{f\},P'\right)$;
12: $E_i\leftarrow E_i+\langle\mathrm{LC}_{\prec_{x,\lambda,m}}(f)\rangle$;
13: $f\leftarrow f-\mathrm{LT}_{\prec_{x,\lambda,m}}(f)$;
14: **end while**
15: **end while**
16: **if** $N_i\not\subset\sqrt{E_i}$ **then**
17: $G\leftarrow G\cup\left\{\left(E_i,N_i,S^{(1)},S^{(2)}\right)\right\}$;
18: **end if**
end

For a given input $E_i,N_i\subset\mathbb{K}[a]$, $S_i^{(1)},Q\subset\mathbb{K}[a][x,\lambda]^n$, CSSMain outputs a CSS for N,Q over $V(E_i)\setminus V(N_i)$. In Algorithm 2, in line 2, a comprehensive standard system of N over $V(E_{in})\setminus V(N_{in})$ is computed. This computation can be done by using [14, 15]. By letting $\left\{\left(E_i,N_i,S_i^{(1)}\right)\right\}_{i=1,\cdots,\ell'}$ be a comprehensive standard system over $V(E_{in})\setminus V(N_{in})$,

*If $j=1$, we suppose $\prod_{l=1}^{j-1}h_l=1$.

†Here, we suppose $P_i\subset\mathbb{K}(a)[x,\lambda]^n$ and compute spoly(f,g) for f,g regarded as elements of $\mathbb{K}(a)[x,\lambda]^n$ by using Eq. (1).

the algorithm computes $S^{(2)}$ for each locally closed set $V(E_i)\setminus V(N_i)$ for $i\in\{1,\cdots,\ell'\}$ in line 4 and outputs a comprehensive standard system for N and Q.

In Algorithm 3, initialization of the set $S^{(2)}$ and the set of S-polynomials for Algorithm 4 is done. In lines 5 and 6 in Algorithm 3, the irreducible factors $\left(h_1,\cdots,h_{n_f}\right)$ and their product h of the product of $\mathrm{LC}_{\prec_{x,\lambda,m}}\left(S^{(2)}\right)$ are computed. If $\sigma_t(h)\neq 0$ for $t\in K^{n_a}$, all the leading coefficients of $\sigma_t\left(S^{(2)}\right)$ are non-zero. Algorithm 3 decomposes the locally closed set $V(E_i)\setminus V(N_i)$ such as

$$V(E_i)\setminus V(N_i)=[V(E_i)\setminus V(hN_i)]\cup\bigcup_{j=1}^{n_f}\left[V(E_i+\langle h_j\rangle)\setminus V\left(\prod_{l=1}^{j-1}h_lN_i\right)\right],$$

and recursively call CSSMain for each locally closed set except to the first one $V(E_i)\setminus V(hN_i)$. On the locally closed set $V(E_i)\setminus V(hN_i)$, all the leading coefficients of the elements in $S^{(2)}$ are non-zero and thus the S-polynomials of the elements in between $S^{(1)}$ and $S^{(2)}$ or that of the elements among $S^{(2)}$ are well-defined on $V(E_i)\setminus V(hN_i)$. The set of the S-polynomials P is initiated in lines 8 - 10 of Algorithm 3 and forwarded to CSSSub in line 11.

In Algorithm 4, CSS on $V(E_i)\setminus V(N_i)$ (Put $E_i=E_i$ and $N_i=hN_i$ to match it with the previous context.) is computed. Note that all the leading coefficients of $S^{(2)}$ are supposed to be non-zero on $V(E_i)\setminus V(N_i)$. For each element f in the set of the S-polynomials P, its reduced normal form with respect to $\left(S^{(1)},S^{(2)}\right)$ and $E_i\mathbb{K}[a][x,\lambda]$ are computed in lines 5 and 6, respectively. If the reduced normal form of f is non-zero, Algorithm 4 enters into the while loop starting from line 7 to line 18. In the while loop, the locally closed set $V(E_i)\setminus V(N_i)$ is decomposed into

$$V(E_i)\setminus V(N_i)=\left[V(E_i)\setminus V\left(\mathrm{LC}_{\prec_{x,\lambda,m}}(f)N_i\right)\right]\cup\left[V\left(E_i+\langle\mathrm{LC}_{\prec_{x,\lambda,m}}(f)\rangle\right)\setminus V(N_i)\right].$$

For the first locally closed set $V(E_i)\setminus V\left(\mathrm{LC}_{\prec_{x,\lambda,m}}(f)N_i\right)$, the leading coefficient of f is non-zero. In this case, Algorithm 4 updates the set of the S-polynomials and $S^{(2)}$ and recursively call CSSSub. For the second locally closed set $V\left(E_i+\langle\mathrm{LC}_{\prec_{x,\lambda,m}}(f)\rangle\right)\setminus V(N_i)$, the leading coefficient of f is zero and thus $\mathrm{LT}_{\prec_{x,\lambda,a}}(f)$ is subtracted from f, E_i is updated to $E_i+\langle\mathrm{LC}_{\prec_{x,\lambda,m}}(f)\rangle$ and the loop continues while $P\neq\emptyset$ and $N_i\not\subset\sqrt{E_i}$. In the end, if $P=\emptyset$ but $N_i\not\subset\sqrt{E_i}$, Algorithm 4 adds the resulting $\left(E_i,N_i,S^{(1)},S^{(2)}\right)$ to G. This is the flow of Algorithms 2-4. For Algorithms 2-4, we can prove the following theorem.

Theorem 3.1 (Correctness and Termination in Finite Steps). *For a given finite set of generators $N,Q\subset\mathbb{K}[a][x,\lambda]^n$ such that $\langle\sigma_t(N)\rangle_{K[x,\lambda]_{\langle x,\lambda\rangle}}$ has a finite codimension as a K vector space in $\left(K[x,\lambda]_{\langle x,\lambda\rangle}\right)^n$ for all $t\in K^{n_a}$, Algorithms 2-4 terminate in finite steps and output a Comprehensive Standard System (*CSS*) for N,Q with respect to $\prec_{x,\lambda,m}$ over $V(E_{in})\setminus V(N_{in})$.*

Proof. First, we prove the correctness of Algorithms 2-4. All the outputs are ones in Algorithm 4, it is enough to prove that the output of Algorithm 4 is a comprehensive standard system for N and Q over $V(E_i)\setminus V(N_i)$. We prove that by showing that all the possible S-polynomials among the elements in $\sigma_t\left(S^{(1)}\right)$ and $\sigma_t\left(S^{(2)}\right)$ are reduced to 0 by $\left(\sigma_t\left(S^{(1)}\right),\sigma_t\left(S^{(2)}\right)\right)$ for all $t\in V(E_i)\setminus V(N_i)$, which implies that $\left(\sigma_t\left(S^{(1)}\right),\sigma_t\left(S^{(2)}\right)\right)$ is an (x,λ)-mixed standard basis for the mixed module generated by $\sigma_t(N)$ and $\sigma_t(Q)$ (Theorem 41 in [1]).

All the S-polynomials between the elements in $\sigma_t\left(S^{(1)}\right)$ are reduced to 0 with respect to $\sigma_t\left(S^{(1)}\right)$ for all $t\in V(E_i)\setminus V(N_i)$ since the triple $\left\{\left(E_i,N_i,S^{(1)}\right)\right\}$ is a comprehensive standard basis over $V(E_i)\setminus V(N_i)$.

All the leading coefficients of the elements in $S^{(1)}$ and $S^{(2)}$ in the input are non-zero over $V(E_i)\setminus V(N_i)$. Therefore, $\sigma_t(\mathrm{spoly}(f,g))=\mathrm{spoly}(\sigma_t(f),\sigma_t(g))$ holds for all $f,g\in S^{(1)}\cup S^{(2)}$ and for all $t\in V(E_i)\setminus V(N_i)$. Under the setting, suppose $\mathrm{spoly}(f,g)$ is reduce to r by $\left(S^{(1)},S^{(2)}\right)$ in $\left(\mathbb{K}(a)[x,\lambda]_{\langle x,\lambda\rangle}\right)^n$ modulo E_i as is done for f in lines 5 and 6 in Algorithm 4. Note that all the divisions occuring in **Algorithm 32** in [1] are divisions by the leading coefficients and thus $\sigma_t(r)$ is well-defined for all $t\in V(E_i)\setminus V(N_i)$. If either $r=0$ or $\sigma_t\left(\mathrm{LC}_{\prec_{x,\lambda,m}}(r)\right)\neq 0$ for $t\in K^{n_a}$, $\sigma_t(r)$ coincides with the normal form of $\sigma_t(\mathrm{spoly}(f,g))$ with respect to $\left(\sigma_t\left(S^{(1)}\right),\sigma_t\left(S^{(2)}\right)\right)$ by a similar argument as in **Theorem 1** in [17]. This means that if $\mathrm{spoly}(f,g)$ is reduced to 0 with respect to $\left(S^{(1)},S^{(2)}\right)$ in $\left(\mathbb{K}(a)[x,\lambda]_{\langle x,\lambda\rangle}\right)^n$ modulo E_i, $\mathrm{spoly}(\sigma_t(f),\sigma_t(g))$ is reduced to 0 with respect to $\left(\sigma_t\left(S^{(1)}\right),\sigma_t\left(S^{(2)}\right)\right)$ in $\left(K[x,\lambda]_{\langle x,\lambda\rangle}\right)^n$ for all $t\in V(E_i)\setminus V(N_i)$. The S-polynomials

$$P_1=\left\{\mathrm{spoly}(f,g)\,\middle|\,\begin{array}{c}f\in S^{(1)},g\in S^{(2)},i=j\text{ and }\alpha\leq\alpha'\\ \mathrm{LM}_{\prec_{x,\lambda,m}}(f)=x^{\alpha}\lambda^{\beta}e_i,\\ \text{and }\mathrm{LM}_{\prec_{x,\lambda,m}}(g)=x^{\alpha'}\lambda^{\beta'}e_j\end{array}\right\}$$

and

$$P_2=\left\{\mathrm{spoly}(f,g)\,\middle|\,\begin{array}{c}f\in S^{(2)},g\in S^{(2)},i=j\text{ and }\alpha=\alpha'\\ \mathrm{LM}_{\prec_{x,\lambda,m}}(f)=x^{\alpha}\lambda^{\beta}e_i,\\ \text{and }\mathrm{LM}_{\prec_{x,\lambda,m}}(g)=x^{\alpha'}\lambda^{\beta'}e_j\end{array}\right\}$$

are reduced to 0 with respect to $\left(S^{(1)},S^{(2)}\right)$ modulo E_i in the end of Algorithm 4. This proves the correctness.

Second, we prove that Algorithms 2-4 terminate in finite steps. The computation of comprehensive standard system of N in line 2 of Algorithm 2 terminates in finite steps.

The recursion in line 7 of Algorithm 3 terminates in finite steps. It can be shown as follows: Every time CSSMain is called at least one of the leading terms of the elements in $S^{(2)}$ is made zero but the number of the terms in the elements in $S^{(2)}$ that can be made zero is finite because the K-codimension of $\sigma_t(N)$ is finite for any $t\in K^{n_a}$, the number of the terms appearing in the reduced normal form with respect to $S^{(1)}$ in $\left(\mathbb{K}(a)[x,\lambda]_{\langle x,\lambda\rangle}\right)^n$ is finite. Therefore, the recursion cannot continue in infinitely many steps and terminates either in the situation that no more leading terms

of the elements in $S^{(2)}$ can be made zero or all the elements in $S^{(2)}$ are zero.

Algorithm 4 terminates in finite steps because the two while loops in lines 2-18 and in lines 7-15 terminate in finite step and the recursion in line 11 terminates in finite steps because every time CSSSub is called the number of the elements in $\mathrm{LM}_{\prec_{x,\lambda,m}}\left(S^{(2)}\right)$ increases by one but the number cannot exceed the $\mathbb{K}(a)$ codimension of $\langle S^{(1)}\rangle_{\mathbb{K}(a)[x,\lambda]_{\langle x,\lambda\rangle}}$. This proves that Algorithms 2-4 terminate in finite steps. □

3.1 Example

Consider

$$f\colon (x_1,x_2)\mapsto \left(y_1=x_1, y_2=x_1^2x_2+x_1x_2^3+\alpha x_2^5+x_2^6+\beta x_2^7\right) \tag{4}$$

(Type 18 in Table 1 in [12]). Its $\mathcal{A}$-codimension depends on the moduli parameters $\alpha,\beta\in\mathbb{R}$. We would like to detect exceptional values of the moduli parameters (In the generic case, it has $\mathcal{A}$-codimension 8 [12].). In this example, the degree of determinacy also depends on the moduli parameters. By applying a result of du Plessis [18], **Lemma 2.6**, f is k-$\mathcal{A}$-determined if

$$\mathcal{M}_2^{k+1}\mathcal{E}_2^2\subset T\mathcal{A}_1(f)+\langle x_1\frac{\partial f}{\partial x_2}\rangle_{\mathbb{R}}+f^*\langle y_2e_1\rangle_{\mathbb{R}} +\mathcal{M}_2^{k+1}f^*(\mathcal{M}_2)\mathcal{E}_2^2+\mathcal{M}_2^{2k+2}\mathcal{E}_2^2 \tag{5}$$

holds. This condition is equivalent to $\langle x_1,x_2\rangle^{k+1}\left(\mathbb{R}[x,y]_{\langle x,y\rangle}\right)^2$ is contained in the (x,y)-mixed module $M=N+Q$ where

$$\begin{aligned}N=&\langle x_1,x_2\rangle\cdot\langle\frac{\partial f}{\partial x_1},\frac{\partial f}{\partial x_2}\rangle_{\mathbb{R}[x,y]_{\langle x,y\rangle}}+\langle x_1\frac{\partial f}{\partial x_2}\rangle_{\mathbb{R}[x,y]_{\langle x,y\rangle}}\\ &+\langle y_1-f_1(x),y_2-f_2(x)\rangle\cdot\left(\mathbb{R}[x,y]_{\langle x,y\rangle}\right)^2\\ &+\langle x_1,x_2\rangle^{k+1}\cdot\langle y_1,y_2\rangle\cdot\left(\mathbb{R}[x,y]_{\langle x,y\rangle}\right)^2\\ &+\langle x_1,x_2\rangle^{2k+2}\cdot\left(\mathbb{R}[x,y]_{\langle x,y\rangle}\right)^2\end{aligned}$$

and $Q=\langle y_1,y_2\rangle^2\cdot\left(\mathbb{R}[y]_{\langle y\rangle}\right)^2+\langle y_2e_1\rangle_{\mathbb{R}[x,y]_{\langle y\rangle}}$.

By computing CSS for the (x,y)-mixed module for $k=7$ by using the same module ordering in Example 2.1, the parameter space $\mathbb{C}^2$ is decomposed into 12 locally closed sets. Note that $\mathbb{C}$ is the algebraic closure of $\mathbb{R}$ and thus Algorithms 2-4 provide a decomposition of $\mathbb{C}^2$ instead of $\mathbb{R}^2$. However, Algorithms 2-4 are based upon arithmetic operations in the ground field only. This means that if the scalars in the input data are contained in $\mathbb{R}$, then all scalars in the output also lie in $\mathbb{R}$. This guarantees that the decomposition restricted to $\mathbb{R}^2$ provides a semi-algebraic decomposition of $\mathbb{R}^2$ such that the pair $\left(S^{(1)},S^{(2)}\right)$ corresponding to each semi-algebraic set specialized at any element in the semi-algebraic set is an (x,y)-mixed standard basis of $\langle\sigma_t(N)\rangle_{\mathbb{R}[x,y]_{\langle x,y\rangle}}+\langle\sigma_t(Q)\rangle_{\mathbb{R}[y]_{\langle y\rangle}}$ for $t\in\mathbb{R}$. By reducing the generators of $\langle x_1,x_2\rangle^{k+1}\left(\mathbb{R}[x,y]_{\langle x,y\rangle}\right)^2$ by the mixed standard basis for each locally closed set, Eq. (5) holds for parameter values in the locally closed set

$$\begin{aligned}&V(\langle 0\rangle)\setminus V\left(\langle 4000\alpha^5\beta-8600\alpha^4\beta-2500\alpha^4\right.\\ &\left.+4260\alpha^3\beta+7825\alpha^3-540\alpha^2\beta-2574\alpha^2+81\alpha\rangle\right)\end{aligned} \tag{6}$$

and thus f is 7-$\mathcal{A}$-determined for the parameter values in the locally closed set in Eq. (6). For the two parameter values, higher jets need to be investigated.

In a similar manner as in Example 2.1, for these parameter values, we can compute the $\mathcal{A}$-codimension of f as follows: We compute CSS for a pair of finite sets of generators of Eq. (2) and Eq. (3) by using the same module ordering in Example 2.1. The output is too complicated to be shown here but the locally closed set in Eq. (6) is decomposed into 4 locally closed sets and the $\mathcal{A}$-codimension of $M=\langle\sigma_t(N)\rangle_{K[x,y]_{\langle x,y\rangle}}+\langle\sigma_t(Q)\rangle_{K[y]_{\langle y\rangle}}$ is 8 for all t in the locally closed set in Eq. (6).

4 APPLICATION TO SINGULARITY THEORY

In this section, we apply CSS for a mixed module to classification of singularities relative to $\mathcal{A}$. In Example 3.1, we show that the map-germ in Eq. (4) is 7-$\mathcal{A}$-determined if the parameters are in the locally closed set in Eq. (6). To demonstrate CSS for a mixed module, we proceed the classification further than [12] up to $\mathcal{A}$-codimension 9. The result is summarized in Table 1. Let us introduce some terminology. Let $\mathcal{H}$ be the unipotent subgroup of $\mathcal{A}$ whose tangent space at f is defined as

$$T\mathcal{H}(f)=\mathcal{M}_2^2\langle\frac{\partial f}{\partial x_1},\frac{\partial f}{\partial x_2}\rangle_{\mathcal{E}_2}+f^*\left(\mathcal{M}_2^2\mathcal{E}_2^2\right)+\langle x_1\frac{\partial f}{\partial x_2},f_2e_1\rangle_{\mathbb{R}}.$$

Let $M_{r,s}(\mathcal{H})$ be filtration of $\mathcal{M}_2\mathcal{E}_2^2$ defined as

$$M_{r,s}(\mathcal{H})=\sum_{i\geq s}(T\mathcal{H})^i\cdot\left(\mathcal{M}_2^r\mathcal{E}_2^2\right)+\mathcal{M}_2^{r+1}\mathcal{E}_2^2$$

for integers $r\geq 1$ and $s\geq 0$ and $M_{0,0}(\mathcal{H})=\mathcal{M}_2\mathcal{E}_2^2$ [9]. The associated (r,s)-jet space $J^{r,s}(2,2)$ is then defined to be $\mathcal{M}_2\mathcal{E}_2^2/M_{r,s}(\mathcal{H})$ and $j^{r,s}\colon\mathcal{M}_2\mathcal{E}_2^2\to J^{r,s}(2,2)$ be the canonical projection. Let $H_{r,s}(\mathcal{H})=j^{r,s}(M_{r,s-1}(\mathcal{H}))$ be the image of $M_{r,s-1}(\mathcal{H})$ in $J^{r,s}(2,2)$. For example, the list of a basis of $H_{r,s}(\mathcal{H})$ regarded as a vector space over $\mathbb{R}$ for $r=8$ is shown in Table 2. For $\mathcal{H}$ and its compatible filtration $M_{r,s}(\mathcal{H})$, the following holds true.

(r,s)	Basis for $H_{r,s}(\mathcal{H})$
$(8,0)$	$\{(0,0)\}$
$(8,1)$	$\{(0,x_2^8)\}$
$(8,2)$	$\{(0,x_1x_2^7),(x_2^8,0)\}$
$(8,3)$	$\{(0,x_1^2x_2^6),(x_1x_2^7,0)\}$
$(8,4)$	$\{(0,x_1^3x_2^5),(x_1^2x_2^6,0)\}$
$(8,5)$	$\{(0,x_1^4x_2^4),(x_1^3x_2^5,0)\}$
$(8,6)$	$\{(0,x_1^5x_2^3),(x_1^4x_2^4,0)\}$
$(8,7)$	$\{(0,x_1^6x_2^2),(x_1^5x_2^3,0)\}$
$(8,8)$	$\{(0,x_1^7x_2),(x_1^6x_2^2,0)\}$
$(8,9)$	$\{(0,x_1^8),(x_1^7x_2,0)\}$
$(8,10)$	$\{(x_1^8,0)\}$

Table 2: List of basis of $H_{r,s}(\mathcal{H})$

Type	Representative	Range	$\mathcal{A}$-codimension
1	$\left(x_1, x_1^2x_2 + x_1x_2^3 + \alpha x_2^5 + x_2^6 + \beta x_2^7\right)$	(α, β) is in Eq. (6)	8
2	$\left(x_1, x_1^2x_2 + x_1x_2^3 + \alpha x_2^5 + x_2^6 + \beta x_2^7 + \gamma x_2^9\right)$	(α, β) is in Eq. (8), $(\alpha, \beta) \neq \left(\frac{9}{11}, \frac{1111}{36}\right)$ $(\alpha, \beta, \gamma) \neq \left(\frac{1}{4}, 9, -1080\right), \left(\frac{9}{5}, -\frac{4}{3}, \frac{1072}{729}\right)$	9
3	$\left(x_1, x_1^2x_2 + x_1x_2^3 + \frac{9}{11}x_2^5 + x_2^6 + \frac{1111}{306}x_2^7 + \gamma x_1x_2^7\right)$	$\gamma \in \mathbb{R}$	9

Table 1: List of representatives of a map-germ having the same 7 jet as Type 18 in [12] of $\mathcal{A}$-codimension less than 10

THEOREM 4.1 (**Theorem 2.9** IN [10]). *Let $f\colon \left(\mathbb{R}^2, 0\right) \to \left(\mathbb{R}^2, 0\right)$ be a smooth map-germ and let T be a vector subspace of $M_{r,s}\left(\mathcal{H}\right)$ such that*

$$M_{r,s}\left(\mathcal{H}\right) \subset T + T\mathcal{H}\left(f\right) + M_{r,s+1}\left(\mathcal{H}\right) \tag{7}$$

holds true. Then any map-germ $g\colon \left(\mathbb{R}^2, 0\right) \to \left(\mathbb{R}^2, 0\right)$ with $g - f \in M_{r,s}\left(\mathcal{H}\right)$ is $\mathcal{H}$-equivalent to a map-germ of the form $f + \tau + \phi$ with $\tau \in T$ and $\phi \in M_{r,s+1}\left(\mathcal{H}\right)$.

We call T complete transversal by following [10]. By using Theorem 4.1, we investigate the higher order jets of the map-germ in Eq. (4) for the parameter values in the locally closed set

$$V\left(\langle 4000\alpha^5\beta - 8600\alpha^4\beta - 2500\alpha^4 + 4260\alpha^3\beta + 7825\alpha^3 - 540\alpha^2\beta - 2574\alpha^2 + 81\alpha\rangle\right) \setminus V\left(\langle 1\rangle\right). \tag{8}$$

To apply Theorem 4.1, we need to find T such that Eq. (7) holds true for $(r, s) = (8, 0)$. Once we find such a vector subspace T, all the map-germs g having the same 7-jet as f are $\mathcal{H}$-equivalent to a map-germ of the form $f + \tau + \phi$ with $\tau \in T$ and $\phi \in M_{8,1}\left(\mathcal{H}\right)$ and thus we get an exhaustive list of representatives of $\mathcal{H}$-equivalence in $(8, 1)$-jet space having the same $(8, 0)$-jet (7-jet) as f.

As a candidate for T, it is enough to consider a basis for $H_{8,1}$, that is, $\left(0, x_2^8\right)$ in Table 2. Given the candidate for T, a basis for T can be computed by using CSS for a mixed module as follows: Let us consider an (x, y)-mixed module $M = N + Q$ where $r = 8$,

$$\begin{aligned} N = {} & \langle x_1, x_2\rangle^2 \cdot \langle \frac{\partial f}{\partial x_1}, \frac{\partial f}{\partial x_2}\rangle_{\mathbb{R}[x,y]_{\langle x,y\rangle}} + \langle x_1 \frac{\partial f}{\partial x_2}\rangle_{\mathbb{R}} \\ & + \langle y_1 - f_1\left(x\right), y_2 - f_2\left(x\right)\rangle \cdot \left(\mathbb{R}\left[x, y\right]_{\langle x,y\rangle}\right)^2 \\ & + \langle \sum_{s=2}^{r+2} H_{r,s}\left(\mathcal{H}\right)\rangle_{\mathbb{R}} + \langle x_1, x_2\rangle^{r+1} \cdot \left(\mathbb{R}\left[x, y\right]_{\langle x,y\rangle}\right)^2 \end{aligned}$$

and $Q = \langle y_1, y_2\rangle^2 \cdot \left(\mathbb{R}\left[y\right]_{\langle y\rangle}\right)^2 + \langle (y_2, 0)\rangle_{\mathbb{R}}$. CSS for finite sets of generators of N and Q over the locally closed set in Eq. (8) is computed as :

(1) $V\left(\langle 4\alpha - 5, 4200\beta - 16829\rangle\right) \setminus V\left(\langle 1\rangle\right)$:

$$\begin{aligned} S_1^{(1)} = \{ & \left(0, y_2 - x_1^2x_2 - x_1x_2^3 - \alpha x_2^5 - x_2^6 - \beta x_2^7\right), \\ & \left(y_2, (-8\alpha + 10)x_1x_2^6 + 16\alpha x_2^8\right), \left(0, y_1 - x_1\right), \left(y_1 - x_1, 0\right), \\ & \left(x_2^2, 2x_1x_2^3 + x_2^5\right), \left(x_1x_2, -5x_1x_2^4 - 10\alpha x_2^6 - 12x_2^7 - 14\beta x_2^8\right), \\ & \left(x_1^2, (-10\alpha + 15)x_1x_2^5 + 25\alpha x_2^7 - 12x_1x_2^6 + 30x_2^8\right), \left(0, x_1x_2^7\right), \\ & \left(0, x_1^3 + (5\alpha - 9)x_1x_2^4 - 15\alpha x_2^6 + 6x_1x_2^5 - 18x_2^7 + 7\beta x_1x_2^6 - 21\beta x_2^8\right), \\ & \left(0, x_1^2x_2^2 + 3x_1x_2^4 + 5\alpha x_2^6 + 6x_2^7 + 7\beta x_2^8\right), \left(0, x_2^9\right)\}, \end{aligned}$$

$$\begin{aligned} S_1^{(2)} = \{ & \left(0, -1/4x_1x_2^5 - 25/8x_2^7 + 6/5x_1x_2^6 - 3x_2^8\right), \left(0, x_1^2\right), \\ & \left(0, -1/4x_1x_2^5 - 25/8x_2^7 + 5/4x_1x_2^6 - 3x_2^8\right), \left(0, -5/2x_2^8\right), \\ & \left(0, x_1x_2^6\right), \left(0, x_1x_2^4 + 75/11x_2^6 + 372/11x_2^7\right)\} \end{aligned}$$

(2) $V\left(\langle 4000\alpha^4\beta - 8600\alpha^3\beta - 2500\alpha^3 + 4260\alpha^2\beta + 7825\alpha^2 - 540\alpha\beta - 2574\alpha + 81\rangle\right) \setminus V\left(\langle \alpha\left(40\alpha^3 - 182\alpha^2 + 273\alpha - 135\right)\rangle\right)$: In the following cases, $S_j^{(1)}$ is the same as $S_1^{(1)}$ for $j = 2, 3, 4$,

$$\begin{aligned} S_2^{(2)} = \{ & \left(0, (2\alpha - 3)/2x_1x_2^5 - 5\alpha/2x_2^7 + 6/5x_1x_2^6 - 3x_2^8\right), \\ & \left(0, (2\alpha - 3)/2x_1x_2^5 + (-5\alpha)/2x_2^7 + 5/4x_1x_2^6 - 3x_2^8\right), \\ & \left(0, (4\alpha - 5)/4x_1x_2^6 + (-2\alpha)x_2^8\right), \left(0, x_2^8\right), \left(0, x_1^2\right), \\ & \left(0, (10\alpha^2 - 33\alpha + 27)/10x_1x_2^4 + \right. \\ & \left. (-6\alpha^2 + 9\alpha)/2x_2^6 + (-3\alpha + 27)/5x_2^7\right)\} \end{aligned}$$

(3) $V\left(\langle 5\alpha - 9, 3\beta + 4\rangle\right) \setminus V\left(\langle 1\rangle\right)$: $S_3^{(1)} = S_1^{(1)}$,

$$\begin{aligned} S_3^{(2)} = \{ & \left(0, (2\alpha - 3)/2x_1x_2^5 + (-5\alpha)/2x_2^7 + 6/5x_1x_2^6 - 3x_2^8\right), \\ & \left(0, x_1^2\right), \left(0, (2\alpha - 3)/2x_1x_2^5 + (-5\alpha)/2x_2^7 + 5/4x_1x_2^6 - 3x_2^8\right), \\ & \left(0, (4\alpha - 5)/4x_1x_2^6 + (-2\alpha)x_2^8\right), \left(0, x_2^8\right), \\ & \left(0, (2\alpha^2 - 3\alpha)/2x_2^6 + (\alpha - 9)/5x_2^7\right)\} \end{aligned}$$

(4) $V\left(\langle a\rangle\right) \setminus V\left(\langle 1\rangle\right)$: $S_4^{(1)} = S_1^{(1)}$,

$$\begin{aligned} S_4^{(2)} = \{ & \left(0, (2\alpha - 3)/2x_1x_2^5 + (-5\alpha)/2x_2^7 + 6/5x_1x_2^6 - 3x_2^8\right), \\ & \left(0, x_1^2\right), \left(0, (2\alpha - 3)/2x_1x_2^5 + (-5\alpha)/2x_2^7 + 5/4x_1x_2^6 - 3x_2^8\right), \\ & \left(0, (4\alpha - 5)/4x_1x_2^6 + (-2\alpha)x_2^8\right), \left(0, x_1x_2^4 + 2x_2^7 + (7\beta + 4)/3x_2^8\right)\} \end{aligned}$$

$\left(0, x_2^8\right)$ is reduced to 0 by CSS in (1-3) and T can be set to $\langle 0\rangle_{\mathbb{R}}$ whereas that cannot be reduced to 0 by CSS in (4) and T needs to be set to $\langle\left(0, x_2^8\right)\rangle_{\mathbb{R}}$. By Theorem 4.1, we conclude

that any map-germ g having the same $(8,0)$-jet as f is $\mathcal{H}$(and thus $\mathcal{A}$)-equivalent to $f+\phi$ where $\phi \in M_{8,1}(\mathcal{H})$ except for $\alpha = 0$, whereas such a g is $\mathcal{A}$-equivalent to $f + \gamma\left(0, x_2^8\right) + \phi$ where $\gamma \in \mathbb{R}$ and $\phi \in M_{8,1}(\mathcal{H})$ for $\alpha = 0$.

Case (1) $\alpha \neq 0$**:** The next non-zero complete transversal in Eq. (7) appears for $(r,s) = (8,2)$ if $(\alpha,\beta) = (9/11, 1111/306)$. In that case, the map germ is $\mathcal{A}$-equivalent to

$$(x_1,x_2) \mapsto \left(x_1, x_1^2x_2 + x_1x_2^3 + \frac{9}{11}x_2^5 + x_2^6 + \frac{1111}{306}x_2^7 + \gamma x_1x_2^7\right)$$

where $\gamma \in \mathbb{R}$, is 8-$\mathcal{A}$-determined and has $\mathcal{A}$-codimension 9. If $(\alpha,\beta) \neq (9/11, 1111/306)$, the next non-zero complete transversal in Eq. (7) appears for $(r,s) = (9,0)$. In that case, the map germ is $\mathcal{A}$-equivalent to

$$(x_1,x_2) \mapsto \left(x_1, x_1^2x_2 + x_1x_2^3 + \alpha x_2^5 + x_2^6 + \beta x_2^7 + \gamma x_2^9\right)$$

where $\gamma \in \mathbb{R}$, is 9-$\mathcal{A}$-determined and has $\mathcal{A}$-codimension 9 except for $(\alpha,\beta,\gamma) = (1/4, 9, -1080), (9/5, -4/3, 1072/729)$.

Case (2) $\alpha = 0$**:** In this case, the next non-zero complete transversal in Eq. (7) appears for $(r,s) = (10,0)$ for parameters in the set $V(\langle \alpha, \beta+4\rangle) \setminus V(\langle 1\rangle)$, whereas it does not appear for parameters in the set $V(\langle\alpha\rangle) \setminus V(\langle\beta+4\rangle)$. By using the approximation lemma (**Lemma 1.3B** in [19]), it can be shown that the $\mathcal{A}$-codimension of f is equal to or greater than 10 in all the cases.

5 CONCLUSION AND REMARKS

We have provided a concrete computational algorithm (Algorithm 1) of standard basis for a mixed module proposed by Gatermann and Hosten [1]. We have extended it to parametric standard system for a mixed module and provided an algorithm to compute it (Algorithms 2-4). We have demonstrated our algorithm in classification of map -germs relative to $\mathcal{A}$ in which complicated moduli structures appear. To get geometrical interpretation of the exceptional moduli parameters, computation of invariants such as ones listed in [12] may be helpful. However, we did not put the information of these invariants in this paper because the main purpose of this paper is to develop an algorithm for comprehensive standard system for a mixed-module and its application to singularity theory. Based on our algorithm, we will report new classification of map-germs relative to $\mathcal{K}_{\mathcal{B}}$, $\mathcal{A}$, and $\mathcal{A}[G]$ for some Lie group G in the forthcoming paper.

ACKNOWLEDGMENTS

H. T. thanks Prof. Shinichi Tajima for his kind instruction on algebraic local cohomology and Prof. Shyūichi Izumiya and Prof. Yutaro Kabata for their fruitful comments on classification in singularity theory. H. T. was supported by JSPS KAKENHI Grant Number JP19K03484, JST PRESTO Grant Number JPMJPR16E8, Institute for Chemical Reaction Design and Discovery (ICReDD) sponsored by World Premier International Research Center Initiative (WPI Initiative), MEXT, Japan, the Research Institute for Mathematical Sciences, an International Joint Usage/Research Center located in Kyoto University. K. N. was supported by JSPS Grant-in-Aid for Scientific Research (C) (No 18K03214).

REFERENCES

[1] K. Gatermann and S. Hosten. Computational algebra for bifurcation theory. *J. Symb. Comput.*, 40:1180, 2005.
[2] J. Mather. Stability of C^∞-mappings III. Finitely determined map-germs. *Publications Mathématiques, Institute des Hautes Études Scientifiques (IHES)*, 35:127, 1968.
[3] M. Golubitsky and D. G. Schaeffer. *Singularities and Groups in Bifurcation Theory*, volume I of *Applied Mathematical Science*. Springer, 1985.
[4] M. Golubitsky and D. G. Schaeffer. *Singularities and Groups in Bifurcation Theory*, volume II of *Applied Mathematical Science*. Springer, 1985.
[5] S. Izumiya, M. Takahashi, and H. Teramoto. Geometric equivalence among smooth map germs. *Methods and Applications of Analysis*, 25:337, 2018.
[6] J. N. Mather. Stability of C^∞ mappings, IV: Classification of stable germs by $\mathbb{R}$ algebras. *Publ. Math. I. H. E. S.*, 37:223, 1969.
[7] N. P. Kirk. Transversal, A Maple Package For Singularity Theory, Version 3.1. 1998.
[8] N. P. Kirk. Computational Aspects of Singularity Theory. *Doctor Thesis*, 1993.
[9] N. P. Kirk. Computational aspects of classifying singularities. *LMS J. Comput. Math.*, 3:207, 2000.
[10] J. W. Bruce, N. P. Kirk, and A. A. du Plessis. Complete transversals and the classification of singularities. *Nonlinearity*, 10:253, 1997.
[11] D. Ratcliffe. Stems and series in $\mathcal{A}$-classification. *Proc. London Math. Soc.*, 70:183, 1995.
[12] J. H. Rieger. Families of maps from the plane to the plane. *J. London Math. Soc.*, 36:351, 1987.
[13] W. Decker, G.-M. Gruel, G. Pfister, and H. Schönemann. SINGULAR 4-0-2 — A computer algebra system for polynomial computations. http://www.singular.uni-kl.de, 2015.
[14] S. Tajima, Y. Nakamura, and K. Nabeshima. Standard bases and algebraic local cohomology for zero dimensional ideals. *Advanced Studies in Pure Mathematics*, 56:341, 2009.
[15] K. Nabeshima and S. Tajima. Algebraic local cohomology with parameters and parametric standard basis for zero-dimensional ideals. *J. Symb. Comp.*, 82:91, 2017.
[16] A. Hashemi and M. Kazemi. Parametric standard bases and their applications. *International Workshop on Computer Algebra in Scientific Computing, CASC 2019: Computer Algebra in Scientific Computing*, page 179, 2019.
[17] A. Montes. A new algorithm for discussing Gröbner bases with parameters. *J. Symb. Comput.*, 33:183, 2002.
[18] J. W. Bruce, A. A. DU Plessis, and C. T. C. Wall. Determinacy and unipotency. *Invent. Math.*, 88:521, 1987.
[19] C. T. C. Wall. Finite Determinacy of Smooth Map-Germs. *Bull. Lond. Math. Soc.*, 13:481, 1981.

Condition Numbers for the Cube. I: Univariate Polynomials and Hypersurfaces

Josué Tonelli-Cueto
Inria Paris & IMJ-PRG
Sorbonne Université
Paris, France
josue.tonelli.cueto@bizkaia.eu

Elias Tsigaridas
Inria Paris & IMJ-PRG
Sorbonne Université
Paris, France
elias.tsigaridas@inria.fr

ABSTRACT

The condition-based complexity analysis framework is one of the gems of modern numerical algebraic geometry and theoretical computer science. One of the challenges that it poses is to expand the currently limited range of random polynomials that we can handle. Despite important recent progress, the available tools cannot handle random sparse polynomials and Gaussian polynomials, that is polynomials whose coefficients are i.i.d. Gaussian random variables.

We initiate a condition-based complexity framework based on the norm of the cube, that is a step in this direction. We present this framework for real hypersurfaces. We demonstrate its capabilities by providing a new probabilistic complexity analysis for the Plantinga-Vegter algorithm, which covers both random sparse (alas a restricted sparseness structure) polynomials and random Gaussian polynomials. We present explicit results with structured random polynomials for problems with two or more dimensions. Additionally, we provide some estimates of the separation bound of a univariate polynomial in our current framework.

CCS CONCEPTS

• **Theory of computation → Computational geometry**; **Design and analysis of algorithms**; • **Mathematics of computing** → *Numerical analysis*; *Computations on polynomials*.

KEYWORDS

condition number; probabilistic complexity; sparse polynomials; subdivision methods; numerical algebraic geometry

ACM Reference Format:
Josué Tonelli-Cueto and Elias Tsigaridas. 2020. Condition Numbers for the Cube. I: Univariate Polynomials and Hypersurfaces. In *International Symposium on Symbolic and Algebraic Computation (ISSAC '20), July 20–23, 2020, Kalamata, Greece.* ACM, New York, NY, USA, 8 pages. https://doi.org/10.1145/3373207.3404054

ISSAC '20, July 20–23, 2020, Kalamata, Greece

ACM ISBN 978-1-4503-7100-1/20/07…$15.00
https://doi.org/10.1145/3373207.3404054

1 INTRODUCTION

The complexity of numerical algorithms is not uniform. It depends on a measure of the numerical sensitivity of the output with respect to perturbations of the input, called *condition number*. This motivates the condition-based complexity analysis of numerical algorithms. As this analysis is not input-independent, a usual technique is to randomize the input to obtain a probabilistic complexity analysis that reflects the behaviour of the algorithm in practice. We refer the reader to [3] for more details about this paradigm of complexity for numerical algorithms.

After the complete solution of Smale's 17th problem [17], the main challenge in numerical algebraic geometry is to extend the current algorithms and their analysis to more general inputs, sparse and structured polynomials. Regarding the solution of sparse polynomial systems over the complex numbers, there is the groundbreaking work of Malajovich [19, 20] and Malajovich and Rojas [21, 22]. Additionally, there is significant progress in the probabilistic analysis of the condition number for solving some structured polynomial systems by Armentano and Beltrán [1], by Beltrán and Kozhasov [2], and by Ergür, Paouris and Rojas [13, 14].

A common problem with many of the current techniques is that they rely on unitary/orthogonal invariance. Developing techniques that do not rely on this invariance is therefore a central task in the goal of being able to deal with sparse/structured polynomials and more general probability distributions. We make another step in this research direction by developing a condition-based complexity framework that relies on the ∞-norm of the cube, and so it does not rely on the above invariance.

In this paper, we develop the above framework for univariate polynomials and hypersurfaces. We hope to extend it for polynomial systems in future work. To illustrate its advantages we apply it to the study of the complexity of the Plantinga-Vegter algorithm [6, 23] and the separation bounds for the roots of a real univariate polynomials.

In the case of the Plantinga-Vegter algorithm, we are able to show that this algorithm is efficient (i.e., takes polynomial time on the average) for a wide class of random sparse polynomials (Theorem 2.10). This significantly extends the results of [7] (cf. [9]). Additionally, we also cover Gaussian polynomials, in which all coefficients have the same variance.

We note that our aim is not to show that the Plantinga-Vegter is the most efficient algorithm for random sparse polynomials, but that it remains efficient when we restrict it to a wide class of random sparse polynomials. A similar approach was employed in [13] for the algorithm for finding real zeros of real polynomial systems from [10]. However, unlike [13], our analysis applies to structured

polynomials that are sparse, but with a combinatorial restriction on the support. We note that our condition is similar to that in [24] and so is the bound we obtain; the latter is polynomial in the degree and the size of the support and exponential in the number of variables.

We also note that our bounds depend polynomially on the degree and not logarithmically. The latter would be ideal in view of the results of Khovanskiĭ [16] and Kushnirenko's hypothesis, which bound the size of the Betti numbers of zero sets of sparse polynomials independently of the degree. However, few progress have been made in this direction beyond the univariate case [15]. Moreover, many computational problems in real algebraic geometry lack algorithms that are polynomial in the degree, so such bounds contribute to the state-of-the-art.

In the case of univariate polynomials, our results imply that the complex roots of a random real univariate sparse polynomial around the unit interval are well-separated with high probability. Given than the logarithm of the separation bound is an important parameter that controls the complexity of many univariate solvers, this will lead to interesting probabilistic complexity bounds for these solvers.

Our framework is based on variational properties of the polynomials and considered condition numbers and probabilistic techniques from geometric functional analysis. The former follows the variational approach to condition numbers of [27, $2^{\S 2}$] and extends [8] to new norms. The latter has been already applied in [13, 14] and [7], but our applications these methods takes them to the maximum development.

The 1-norm on the space of polynomials behaves as the "dual"norm to the ∞-norm on the cube. This norm is naturally suited for subdivision methods on the cube. The analysis of the Plantinga-Vegter subdivision process using our framework serves the purpose to convince the reader of the advantages of the new framework for the analysis of algorithms. It also has the ambition to bring new insights in the study of algorithms in numerical algebraic geometry. Our approach continues the trend started by [7] of bringing further interactions between the communities of numerical algebraic geometry and symbolic computation.

Notation. Let $\mathcal{P}_{n,d}$ be the space of polynomials in n variables of total degree at most d, $I^n := [-1,1]^n \subset \mathbb{R}^n$ the unit cube and $B_{\mathbb{C}}(x,r)$ complex disk centered at x of radius r. A polynomial $f \in \mathcal{P}_{n,d}$, is $f = \sum_{|\alpha|\leq d} f_\alpha X^\alpha$, even though we commonly omit the summation index. For $X \subseteq \mathbb{R}^n$, we denote by $\mathcal{B}(X)$ the set of boxes (i.e., cubes) contained in X. For any $B \in \mathcal{B}(\mathbb{R}^n)$, we denote by $m(B)$ its *midpoint* and by $w(I)$ its *width*, so that $B = m(B) + w(B)/2[-1,1]^n$.

Organization. In the next section, we introduce the randomness model that we will consider, zintzo random polynomials, and how our framework applies to the subdivision routine of the Plantinga-Vegter algorithm. In Section 3, we introduce the norms with which we will be working and their main properties. In Section 4, we introduce a new condition number adapted to the introduced norms and we prove its main properties. In Section 5, we develop a probabilistic analysis of the introduced condition number for zintzo random polynomials. Finally, in Section 6, we perform the complexity analysis of the subdivision routine of the Plantinga-Vegter algorithm; and in Section 7, we introduce the separation bound.

2 MAIN RESULTS

In this paper, the main result is a different condition-based framework that allows to control the probability of numerical algorithms with respect random polynomials that are sparse and don't have any scaling in their coefficients, as it has been usual with the so-called KSS or dobro random polynomials introduced in [7]. We showcase our techniques with the Plantinga-Vegter algorithm.

2.1 Randomness model

We introduce a new class of random polynomials that is similar to the class of dobro random polynomials [7]. The main difference is that we require a scaling in the coefficients of the random polynomials. In this way, the new class is a more natural model of random polynomials. Moreover, we explicitly include sparseness in the model of randomness.

Let us recall some basic definitions.

(SG) We call a random variable X *subgaussian*, if there exist a $K > 0$ such that for all $t \geq K$,

$$\mathbb{P}(|X| > t) \leq 2\exp(-t^2/K^2).$$

The smallest such K is the *subgaussian constant* of X.

(AC) A random variable X has the *anti-concentration property*, if there exists a $\rho > 0$, such that for all $\varepsilon > 0$,

$$\max\{\mathbb{P}\left(|X-u| \leq \varepsilon\right) \mid u \in \mathbb{R}\} \leq 2\rho\varepsilon.$$

The smallest such ρ is the anti-concentration constant of X.

Definition 2.1. Let $M \subseteq \mathbb{N}^n$ be a finite set such that $0, e_1, \ldots, e_n \in M$. A *zintzo random polynomial supported on* M is a random polynomial $\mathfrak{f} = \sum_{\alpha\in M} \mathfrak{f}_\alpha X^\alpha \in \mathcal{P}_{n,d}$ such that the coefficients $\mathfrak{f}_\alpha$ are independent subgaussian random variables with the anti-concentration property.

Remark 2.2. The word "zintzo" is a Basque word that means honest, upright, righteous. We use this word instead of a variation of dobro to emphasize that this class of random polynomials is different from the class of dobro polynomials.

Remark 2.3. The technical condition $0, e_1, \ldots, e_n \in M$ is there because is needed in our proofs. In layman's terms, this technical condition states that all the terms of the first order approximation of $\mathfrak{f}$ at 0, $\mathfrak{f}_0 + \mathfrak{f}_{e_1}X_1 + \cdots + \mathfrak{f}_{e_n}X_n$, appear with probability one. In terms of the Newton polytope, this condition implies that the tangent cone of the Newton polytope at 0 is a simple cone.

Given a zintzo random polynomial, the complexity estimates that we present in the sequel depend on the product of the following two parameters:

(1) the *subgaussian constant* of $\mathfrak{f}$ which is given by

$$K_{\mathfrak{f}} := \sum\nolimits_{\alpha\in M} K_\alpha, \tag{2.1}$$

where K_α is the subgaussian constant of $\mathfrak{f}_\alpha$, and

(2) the *anti-concentration constants* of $\mathfrak{f}$ which is given by

$$\rho_{\mathfrak{f}} := \sqrt[n+1]{\rho_0\rho_{e_1}\cdots\rho_{e_n}}, \tag{2.2}$$

where ρ_0 is the anti-concentration constant of $\mathfrak{f}_0$ and for each i, ρ_{e_i} is the anti-concentration constant of $\mathfrak{f}_{e_i}$.

We note that the product $K_{\mathfrak{f}}\rho_{\mathfrak{f}}$ that will appear in our estimates is invariant under multiplication of $\mathfrak{f}$ by non-zero scalars. It also satisfies the following inequality, which we will prove in Section 5.

Proposition 2.4. *Let $\mathfrak{f}$ be a zintzo random polynomial supported on M. Then $K_{\mathfrak{f}}\rho_{\mathfrak{f}} > (n+1)/4 \geq 1/2$.*

Let $M \subseteq \mathbb{N}^n$ be such that it contains $0, e_1, \ldots, e_n$. The following are the two most important examples of our randomness model.

- G A *Gaussian polynomial supported on M* is a zintzo random polynomial $\mathfrak{f}$ supported on M, the coefficients of which are i.i.d. Gaussian random variables. In this case, it holds that $\rho_{\mathfrak{f}} = 1/\sqrt{2\pi}$ and $K_{\mathfrak{f}} \leq |M|$.
- U A *uniform random polynomial supported on M* is a zintzo random polynomial $\mathfrak{f}$ supported on M, the coefficients of which are i.i.d. uniform random variables on $[-1, 1]$. In this case, $\rho_{\mathfrak{f}} = 1/2$ and $K_{\mathfrak{f}} \leq |M|$.

An important feature of our randomness model is that it includes the smoothed analysis inside the probabilistic analysis. We recall that the smoothed case, as introduced by Spielman and Teng [26], considers a fixed polynomial on which we perform a random perturbation. Recall that $\|f\|_1 := \sum_\alpha |f_\alpha|$. The presence of the norm in the following statement is to make the random perturbation of size proportional to the size of the polynomial.

Proposition 2.5. *Let $\mathfrak{f}$ be a zintzo random polynomial supported on M, $f \in \mathcal{P}_{n,d}$ a polynomial supported on M, and $\sigma > 0$. Then, $\mathfrak{f}_\sigma := f + \sigma\|f\|_1\mathfrak{f}$ is a zintzo random polynomial supported on M such that $K_{\mathfrak{f}_\sigma} \leq \|f\|_1(1 + \sigma K_{\mathfrak{f}})$ and $\rho_{\mathfrak{f}_\sigma} \leq \rho_{\mathfrak{f}}/(\sigma\|f\|_1)$. In particular,*

$$K_{\mathfrak{f}_\sigma}\rho_{\mathfrak{f}_\sigma} = (K_{\mathfrak{f}} + 1/\sigma)\rho_{\mathfrak{f}}.$$

The proof of the proposition appears in Section 5. Note that

$$\lim_{\sigma\to 0} K_{\mathfrak{f}_\sigma}\rho_{\mathfrak{f}_\sigma} = \infty \text{ and } \lim_{\sigma\to\infty} K_{\mathfrak{f}_\sigma}\rho_{\mathfrak{f}_\sigma} = K_{\mathfrak{f}}\rho_{\mathfrak{f}},$$

so that we have that the smoothed case recovers both the worst and the average case. In particular, the worst case emerges as the perturbation becomes zero and the average case as the perturbation becomes of infinite magnitude.

Remark 2.6. We use the term subgaussian constant instead of the ψ_2-norm since our choice may not agree with the usual definition of ψ_2-norm which is

$$\|X\|_{\psi_2} := \inf\{t > 0 \mid \mathbb{E}\exp(-X^2/t^2) \leq 2\},$$

see [28, Definition 2.5.6]. However, one can see that what we call subgaussian constant is always bounded from above by the ψ_2-norm.

Remark 2.7. Our methods also apply if we replace the subgaussian property by the more general subexponential property [28, 2.7] or by probability distributions having stronger tail decays (see [28, Exercise 2.7.3]).

Remark 2.8. Saying that X has the anti-concentration property with anti-concentration constant ρ is the same as saying that X has a density (with respect the Lebesgue measure) bounded almost everywhere by ρ. See [25] for more details on this.

Remark 2.9. By Proposition 2.5, any probabilistic average complexity analysis includes the smoothed complexity analysis. Because of this, we will only provide complexity estimates in the average case.

2.2 Complexity results

Our main complexity result is the following probabilistic complexity analysis for the subdivision routine of the Plantinga-Vegter, PV-Subdivsion, that we prove in Section 6.

Theorem 2.10. *Let $\mathfrak{f} \in \mathcal{P}_{n,d}$ be a zintzo random polynomial supported on M. The average number of boxes of the final subdivision of PV-Subdivsion using the interval approximations* (6.1) *and* (6.2) *on input $\mathfrak{f}$ is at most*

$$n^{\frac{3}{2}} d^{2n} |M| \left(80\sqrt{n(n+1)}K_{\mathfrak{f}}\rho_{\mathfrak{f}}\right)^{n+1}.$$

Let us particularize the result for the two main examples of zintzo random polynomials.

Corollary 2.11. *Let $\mathfrak{f} \in \mathcal{P}_{n,d}$ be a random polynomial supported on M. The average number of boxes of the final subdivision of PV-Subdivsion using the interval approximations* (6.1) *and* (6.2) *on input $\mathfrak{f}$ is at most*

$$n^{\frac{3}{2}} \left(40\sqrt{n(n+1)}\right)^{n+1} d^{2n}|M|^{n+2}$$

if $\mathfrak{f}$ is Gaussian or uniform.

We observe that in all these results the bound is polynomial in the degree, as in [7], providing further theoretical justification of the practical success of the Plantinga-Vegter algorithm. However, unlike the estimates in [7], the above results justify the success of the Plantinga-Vegter algorithm for sparse random polynomials. As mentioned in the introduction, this is one of the first such probabilistic complexity estimates in numerical algebraic geometry.

3 A NORM TO WORK IN THE CUBE

To work in the cube I^n, we will use the ∞-norm which is

$$\|x\|_\infty := \max_i |x_i|,$$

for $x \in \mathbb{R}^n$. Motivated by duality, we will consider the following norm on $\mathcal{P}_{n,d}$, the space of affine polynomials of degree at most d in n variables:

$$\|f\|_1 := \sum_\alpha |f_\alpha|, \tag{3.1}$$

for $f := \sum_{|\alpha|\leq d} f_\alpha X^\alpha \in \mathcal{P}_{n,d}$.

The motivation to choose the 1-norm emanates from the following proposition which shows that we can control the evaluation of f at $x \in I^n$, that is $f(x)$, using 1-norm for f.

Proposition 3.1. *Let $f \in \mathcal{P}_{n,d}$ and $x \in I^n$. Then $|f(x)| \leq \|f\|_1$.*

Proof. It holds $|f(x)| = |\sum_\alpha f_\alpha x^\alpha| \leq \sum_\alpha |f_\alpha|\|x\|_\infty^{|\alpha|} \leq \|f\|_1$; as $x \in I^n$ implies that $\|x\|_\infty \leq 1$. □

Remark 3.2. A reader might wonder why we do not choose another norm. For example, if we choose $\|f\|_2 := \sqrt{\sum_\alpha |f_\alpha|^2}$, then we can prove that for all $x \in I^n$, it holds $|f(x)| \leq \sqrt{N}\|f\|_2$. Unfortunately, the inequality depends on $\sqrt{N}$. This $\sqrt{N}$ factor will spread throughout the analysis and it will take away any gain we obtain from choosing the Euclidean norm. Because of this, we pick the norm that makes our analysis as simple as possible, that is the 1-norm.

An important feature of the 1-norm is that, using the norm of a polynomial, we can control the norm of its derivative. Proposition 3.4 and its Corollary 3.5 quantify this feature.

Remark 3.3. We use the convention of writing ∇f to refer to the formal gradient vector, whose entries are the formal partial derivatives of f. We write $\nabla_x f$ to refer to the gradient vector of f at x, whose entries are the partial derivatives of f evaluated at x. In this way, for $v \in \mathbb{R}^n$, $\langle \nabla f, v\rangle = \sum_i v_i \partial_i f$ is a polynomial, while $\langle \nabla_x f, v\rangle = \sum_i v_i \partial_i f(x)$ is a number.

Proposition 3.4. *Let $f \in \mathcal{P}_{n,d}$, $x \in I^n$, and $v \in \mathbb{R}^n$. Then, it holds $\|\langle \nabla f, v\rangle\|_1 \leq d\|f\|_1\|v\|_\infty$.*

Proof. We have $d\|f\|_1\|v\|_\infty = \sum_\alpha d|f_\alpha|\|v\|_\infty$ and $\|\langle \nabla f, v\rangle\|_1 \leq \sum_\alpha |f_\alpha|\|\langle \nabla(X^\alpha, v\rangle\|_1$. Therefore, it is enough to prove the claim for X^α. But then $\langle \nabla X^\alpha, v\rangle = \sum_{i=1}^n \alpha_i v_i X^\alpha / X_i$ and so $\|\langle \nabla X^\alpha, v\rangle\|_1 \leq \left(\sum_{i=1}^n \alpha_i\right)\|v\|_\infty \leq d\|v\|_\infty$. □

Corollary 3.5. *The map $\hat{f} : I^n \to \mathbb{R}$, given by $x \mapsto \hat{f}(x) = f(x)/\|f\|_1$, is d-Lipschitz with respect the ∞-norm.*

Proof. By the fundamental theorem of calculus, $|f(x) - f(y)| \leq \int_0^1 |\langle \nabla_{x+t(x-y)} f, x - y\rangle|\, dt$. Now, by Proposition 3.1, the integrand is bounded from above by $d\|f\|_1\|x - y\|_\infty$. Hence $|f(x) - f(y)| \leq d\|f\|_1\|x - y\|_\infty$, as desired. □

Recall that, by duality, it is natural to measure the gradient of f with the 1-norm, which, for $y \in \mathbb{R}^n$ is

$$\|y\|_1 := \sum_{i=1}^n |y_i|.$$

This is so, because this norm is the optimal norm satisfying the condition that for all $x, y \in \mathbb{R}^n$,

$$\langle y, x\rangle \leq \|y\|_1\|x\|_\infty.$$

This motivates the choice of norms in corollary below.

Corollary 3.6. *The map $\widehat{\nabla f} : I^n \to \mathbb{R}$, given by $x \mapsto \widehat{\nabla f}(x) := \nabla_x f/(d\|f\|_1)$, is $(d-1)$-Lipschitz with respect the ∞-norm in the domain and the 1-norm on the codomain.*

Proof. By Proposition 3.4 and Corollary 3.5, the map $x \mapsto \langle \nabla_x f, v\rangle/(d\|f\|_1\|v\|_\infty)$ is $(d-1)$-Lipschitz with respect the ∞-norm. Hence for all $v \in \mathbb{R}^n \setminus 0$, it holds

$$\frac{1}{\|v\|_\infty}\left|\left\langle \frac{\nabla_x f}{d\|f\|_1} - \frac{\nabla_y f}{d\|f\|_1}, v\right\rangle\right| \leq (d-1)\|x - y\|_\infty.$$

If we maximize the left hand side, then we obtain the 1-norm (as it is the dual norm of the ∞-norm) and so

$$\left\|\frac{\nabla_x f}{d\|f\|_1} - \frac{\nabla_y f}{d\|f\|_1}\right\|_1 \leq (d-1)\|x - y\|_\infty,$$

which concludes the proof. □

4 CONDITION AND ITS PROPERTIES

The following definition adapts the real local condition number [3, Chapter 19] to our setting.

Definition 4.1. Let $f \in \mathcal{P}_{n,d}$ and $x \in I^n$, the *local condition number of f at x* is the quantity

$$\mathtt{C}(f, x) := \frac{\|f\|_1}{\max\left\{|f(x)|, \frac{1}{d}\|\nabla_x f\|_1\right\}}.$$

Remark 4.2. We note that $\mathtt{C}(f, x)$ is infinity only when f has a singular zero at x. In all the other cases, it is finite and it measures how close is f to having a singularity at x.

Following [27, 2[§2]], a condition number should satisfy the following properties: regularity inequality, the 1st and the 2nd Lipschitz property, and the Higher Derivative Estimate. These properties are the ones that we usually need to bound the various quantities when we analyze algorithms in real numerical algebraic geometry.

Proposition 4.3 (Regularity inequality). *Let $f \in \mathcal{P}_{n,d}$ and $x \in I^n$. Then,*

$$\text{either } |f(x)|/\|f\|_1 \geq 1/\mathtt{C}(f, x) \text{ or } \|\nabla_x f\|_1/(d\|f\|_1) \geq 1/\mathtt{C}(f, x).$$

Proof. This follows from the observation that $1/\mathtt{C}(f, x)$ is the maximum of $|f(x)|/\|f\|_1$ and $\|\nabla_x f\|_1/(d\|f\|_1)$. □

Proposition 4.4 (1st Lipschitz property). *The map $\mathcal{P}_{n,d} \ni f \mapsto \|f\|_1/\mathtt{C}(f, x)$ is 1-Lipschitz.*

Proof. If we apply the reverse triangle inequality several times, we get

$$\begin{aligned}
&|\|f\|_1/\mathtt{C}(f, x) - \|g\|_1/\mathtt{C}(g, x)| \\
&\leq |\max\{|f(x)| - |g(x)|, \|\nabla_x f\|_1/d - \|\nabla_x g\|_1/d\}| \\
&\leq |\max\{|f(x) - g(x)|, \|\nabla_x f - \nabla_x g\|_1/d\}| \\
&\leq |\max\{|(f-g)(x)|, \|\nabla_x (f-g)\|_1/d\}|.
\end{aligned}$$

Finally, Propositions 3.1 and 3.4 conclude the proof. □

Let $\Sigma_x \leq \mathcal{P}_{n,d}$ be the subspace of polynomials that are singular at 0, that is

$$\Sigma_x := \{g \in \mathcal{P}_{n,d} \mid g(x) = 0,\ \nabla_x g = 0\}.$$

We cannot prove a Condition Number Theorem where the condition number is (the inverse of) the distance to the discriminantal variety. However, bound the condition number, in both directions, with this distance.

Corollary 4.5 (Condition Number Theorem). *For all $f \in \mathcal{P}_{n,d}$ and $x \in I^n$,*

$$\|f\|_1/\mathrm{dist}_1(f, \Sigma_x) \leq \mathtt{C}(f, x) \leq 2d\,\|f\|_1/\mathrm{dist}_1(f, \Sigma_x)$$

where dist_1 is the distance induced by the 1-norm.

Proof. The left hand side follows from Proposition 4.4. For the right hand side, consider the polynomial

$$g := f - f(x) - \textstyle\sum_{i=1}^n \partial_i f(x) X_i.$$

It is clear that $g \in \Sigma_x$ and that $\|f - g\|_1 \leq |f(x)| + \|\nabla_x f\|_1$. Hence $\mathrm{dist}_1(f, \Sigma_x) \leq \|f - g\|_1 \leq 2d \max\{|f(x)|, \|\nabla_x f\|_1/d\} = 2d\|f\|_1/\mathtt{C}(f, x)$, as desired. □

Proposition 4.6 (2nd Lipschitz property). *The map $I^n \ni x \mapsto 1/\mathtt{C}(f, x)$ is d-Lipschitz.*

Proof. Without loss of generality, we can assume that $\|f\|_1 = 1$. The proof is analogous, mutatis mutandis, to the proof of Proposition 4.4. By using the reverse triangular inequality, we have

$$\left|\frac{1}{\mathtt{C}(f, x)} - \frac{1}{\mathtt{C}(f, y)}\right| \leq \max\left\{|f(x) - f(y)|, \frac{1}{d}\|\nabla_x f - \nabla_y f\|\right\}.$$

Now, Corollaries 3.5 and 3.6 conclude the proof. □

We recall that Smale's gamma, γ, is the invariant given by

$$\gamma(f,x) := \sup_{k\geq 2}\left\|\frac{1}{k!}\,\mathrm{D}_x f^{\dagger}\,\mathrm{D}_x^k f\right\|^{\frac{1}{k-1}} = \sup_{k\geq 2}\left(\frac{1}{\|\nabla_x f\|_2^2}\left\|\frac{1}{k!}(\nabla_x f)^*\,\mathrm{D}_x^k f\right\|\right)^{\frac{1}{k-1}},$$

where the $\dagger$ is the pseudoinverse, and the norm the operator norm with respect the Euclidean norm. We also notice that the second equality follows from computing the pseudoinverse for a covector. The following proposition serves the purpose of the Higher Derivative Estimate [3, Prop. 16.45] in our setting.

Proposition 4.7 (Higher Derivative Estimate). *Let $x \in I^n$ be such that $\mathsf{C}(f,x)\hat{f}(x) < 1$. Then*

$$\gamma(f,x) \leq \frac{1}{2}(d-1)\sqrt{n}\,\mathsf{C}(f,x).$$

Proof. Let $\mathrm{D}_X^k f(v_1,\dots,v_k)$ stand for the polynomial obtained by evaluating the formal kth derivative of f evaluated at $v_1, \dots, v_k \in \mathbb{R}^n$. Then, by Proposition 3.4 and induction, we have

$$\left\|\frac{1}{k!}\,\mathrm{D}_X f(v_1,\dots,v_k)\right\|_1 \leq \binom{d}{k}\|f\|_1\|v_1\|_\infty\cdots\|v_k\|_\infty.$$

Now, by the above inequality, $\|v\|_\infty \leq \|v\|_2$ and submultiplicativity of operator norms, we have that

$$\frac{1}{\|\nabla_x f\|_2^2}\left\|\frac{1}{k!}(\nabla_x f)^*\,\mathrm{D}_x^k f\right\| \leq \frac{\|f\|_1}{\|\nabla_x f\|_2}\binom{d}{k}.$$

Since $\|\nabla_x f\|_2 \geq \|\nabla_x f\|_1/\sqrt{n}$, we deduce that can bound the previous inequality by

$$\binom{d}{k}\sqrt{n}\frac{\|f\|_1}{\|\nabla_x f\|_1} \leq \frac{1}{d}\binom{d}{k}\sqrt{n}\,\mathsf{C}(f,x),$$

where the inequality follows from the Regularity Inequality (Proposition 4.3). Finally, we observe that $\frac{1}{d}\binom{d}{k} \leq (d-1)^{k-1}/2^{k-1}$; then, the claim follows by taking the $(k-1)$th root and the supremum. □

5 PROBABILITY ESTIMATES

We refine the techniques of [7] to obtain explicit constants in the bounds and to deal with a restricted class of sparse polynomials.

5.1 Probabilistic toolbox

Our probabilistic toolbox should control, on the one hand, the norm and, on the other hand, the size of the projection. For the former we need a variant of the Hoeffding inequality, and for latter we need a bound on small ball probabilities.

Proposition 5.1. *Let $\mathfrak{x} \in \mathbb{R}^M$ be a random vector such that for each $\alpha \in M$, $\mathfrak{x}_\alpha$ is subgaussian with subgaaussian constant K_α. Then for all $t \geq \sum_\alpha K_\alpha$, it have*

$$\mathbb{P}(\|\mathfrak{x}\|_1 \geq t) \leq 2|M|\exp\left(-t^2/\left(\sum_{\alpha\in M} K_\alpha\right)^2\right).$$

Proof. We have that

$$\begin{aligned}\mathbb{P}\left(\textstyle\sum_{\alpha\in M}|\mathfrak{x}_\alpha| \geq t\right) &= \mathbb{P}\left(\textstyle\sum_{\alpha\in M}|\mathfrak{x}_\alpha| \geq \sum_{\alpha\in M} K_\alpha t/(\sum_{\alpha\in M} K_\alpha)\right)\\ &\leq \mathbb{P}\left(\exists\alpha\in M \mid |\mathfrak{x}_\alpha| \geq K_\alpha t/(\textstyle\sum_{\alpha\in M} K_\alpha)\right)\\ &\leq |M|\max_{\alpha\in M}\mathbb{P}\left(|\mathfrak{x}_\alpha| \geq K_\alpha t/(\textstyle\sum_{\alpha\in M} K_\alpha)\right)\\ &\leq 2|M|\exp\left(-t^2/(\textstyle\sum_{\alpha\in M} K_\alpha)^2\right),\end{aligned}$$

where the first inequality follows from the implication bound –note that for $x, y \in \mathbb{R}^n_+$, we have that if $\sum_{i=1}^n x_i \geq \sum_{i=1}^n y_i$, then for some i, $x_i \geq y_i$, as otherwise the first claim would be false– the second one from the union bound, and the third one by hypothesis. □

Proposition 5.2. *Let $A \in \mathbb{R}^{k\times N}$ be a surjective linear map and $\mathfrak{x} \in \mathbb{R}^N$ be a random vector such that the $\mathfrak{x}_i$'s are independent random variables with densities (with respect the Lebesgue measure) bounded almost everywhere by ρ. Then, for all measurable $U \subseteq \mathbb{R}^k$,*

$$\mathbb{P}(A\mathfrak{x} \in U) \leq \mathrm{vol}(U)\left(\sqrt{2}\rho\right)^k/\sqrt{\det AA^*}.$$

Proof. Using SVD, write $A = QSP$ where, $P \in \mathbb{R}^{k\times N}$ is an orthogonal projection, S a diagonal matrix containing the singular values of A, and Q an orthogonal matrix.

By [25, Theorem 1.1], see also [18, Theorem 1.1] for the explicit constant, we have that $P\mathfrak{x} \in \mathbb{R}^k$ is a random vector with density bounded, almost everywhere, by $(\sqrt{2}\rho)^k$. Hence

$$\mathbb{P}(A\mathfrak{x} \in U) = \mathbb{P}(P\mathfrak{x} \in (QS)^{-1}U) \leq \mathrm{vol}\left((QS)^{-1}U\right)(\sqrt{2}\rho)^k.$$

This suffices to conclude the proof, since we have $\mathrm{vol}\left((QS)^{-1}U\right) = \mathrm{vol}(U)/\det(QS)$ and $\det(QS) = \sqrt{\det AA^*}$. □

5.2 Condition of zintzo random polynomials

We apply our probabilistic toolbox to zintzo random polynomials.

Theorem 5.3. *Let $\mathfrak{f} \in \mathcal{P}_{n,d}$ a zintzo random polynomial supported on M. Then for all $t \geq e$,*

$$\mathbb{P}(\mathsf{C}(\mathfrak{f},x) \geq t) \leq \sqrt{n}d^n|M|\left(8K_{\mathfrak{f}}\rho_{\mathfrak{f}}\right)^{n+1}\frac{\ln^{\frac{n+1}{2}} t}{t^{n+1}}.$$

Lemma 5.4. *Let $M \subseteq \mathbb{N}^n$ as in Definition 2.1 and $\mathcal{P}_{n,d}(M)$ the set of polynomials in $\mathcal{P}_{n,d}$ supported on M. Let $R_x : \mathcal{P}_{n,d}(M) \to \mathbb{R}^{n+1}$ be the linear map given by*

$$R_x : f \mapsto \begin{pmatrix} f(x) & \frac{1}{d}\partial_1 f(x) & \cdots & \frac{1}{d}\partial_n f(x)\end{pmatrix}^*,$$

and $S : \mathcal{P}_{n,d}(M) \to \mathcal{P}_{n,d}(M)$ be the linear map given by

$$S : f = \sum_{\alpha\in M} f_\alpha X^\alpha \mapsto \sum_{\alpha\in M}\rho_\alpha f_\alpha X^\alpha,$$

where $\rho \in \mathbb{R}^M_+$. Then for $\tilde{R}_x := R_x S^{-1}$ we have that

$$\sqrt{\det \tilde{R}_x\tilde{R}_x^*} \geq \frac{1}{d^n\rho_0\rho_{e_1}\cdots\rho_{e_n}},$$

with respect to coordinates induced by the standard monomial basis.

Proof of Theorem 5.3. We write $\mathsf{C}(\mathfrak{f},x) = \|f\|_1/\|R_x\mathfrak{f}\|$, where R_x is as in Lemma 5.4 and the norm $\|\cdot\|$ in the denominator is

given by $\|y\| = \max\{|y_1|, |y_2| + \cdots + |y_{n+1}|\}$. By the union bound, we have that for $u, s > 0$, it holds

$$\mathbb{P}(\mathtt{C}(\mathtt{f}, x) \geq t) \leq \mathbb{P}(\|\mathtt{f}\| \geq u) + \mathbb{P}(\|A_x \mathtt{f}\| \leq u/t). \tag{5.1}$$

By Propositions 5.1, we have that for $u \geq K_{\mathtt{f}}$,

$$\mathbb{P}(\|\mathtt{f}\| \geq u) \leq 2|M| \exp\left(-u^2/K_{\mathtt{f}}^2\right). \tag{5.2}$$

Let $S : \mathcal{P}_{n,d}(M) \to \mathcal{P}_{n,d}(M)$ be as in Lemma 5.4 with ρ_α the anti-concentration constant of $\mathtt{f}_\alpha$. Then, we have that $S\mathtt{f}$ has independent random coefficients with densities bounded (almost everywhere) by 1 and so we can apply to it to the Proposition 5.2. We do so with the help of Lemma 5.4, so that we obtain

$$\mathbb{P}(\|R_x \mathtt{f}\| \leq u/t) = \mathbb{P}(\|\tilde{R}_x(S\mathtt{f})\| \leq u/t) \leq \frac{2^{n+1}}{n!} d^n (\sqrt{2}\rho_{\mathtt{f}} u/t)^{n+1}, \tag{5.3}$$

where $\tilde{R}_x$ is as in Lemma 5.4.

Combining (5.1), (5.2), and (5.3) with $u = K_{\mathtt{f}}\sqrt{(n+1)\ln t}$, we get

$$\mathbb{P}(\mathtt{C}(\mathtt{f}, x) \geq t) \leq \frac{2|M|}{t^{n+1}} + \frac{2^{n+1}}{n!} d^n (\sqrt{2} K_{\mathtt{f}} \rho_{\mathtt{f}} (n+1))^{n+1} \frac{\ln^{\frac{n+1}{2}} t}{t^{n+1}}.$$

By Stirling's formula,

$$(n+1)^{n+1}/n! \leq \sqrt{n} e^n (1 + 1/n)^{n+1}/\sqrt{2\pi} \leq \sqrt{n} e^{n+1},$$

and so the desired claim follows for $t \geq e$, by Proposition 2.4. □

Proof of Lemma 5.4. The maximal minor of A_x is given by

$$\begin{pmatrix} 1 & x^* \\ 0 & \frac{1}{d}\mathbb{I} \end{pmatrix}.$$

This is precisely the minor associated to the subset $\{1, X_1, \ldots, X_n\}$ of the standard monomial basis of $\mathcal{P}_{n,d}(M)$. Note that at this point we require the assumption that $0, e_1, \ldots, e_n \in M$.

By the Cauchy-Binet identity, $\sqrt{\det A_x A_x^*}$ is lower-bounded by the absolute value of the determinant of the given maximal minor. Hence the lemma follows. □

Proof of Proposition 2.4. Using the positivity of the subgaussian constants, K_α, of the coefficients of the zintzo polynomial $\mathtt{f}$ and the arithmetic-geometric inequality,

$$K_{\mathtt{f}}\rho_{\mathtt{f}} \geq (n+1)\sqrt[n+1]{(K_0\rho_0)(K_{e_1}\rho_{e_1})\cdots(K_{e_n}\rho_{e_n})}.$$

Hence, it suffices to show that for a random variable with X with subgaussian constant K and anti-concentration constant ρ, $K\rho \geq 1/4$. Now, by definition,

$$3K\rho \geq \mathbb{P}(|X| \leq 1.5K) = 1 - \mathbb{P}(|X| > 1.5K) \geq 1 - \exp(-2.25).$$

Calculating we get $K\rho \geq 1/4$, as desired. □

Corollary 5.5. *Let $\mathtt{f} \in \mathcal{P}_{n,d}$ be a zintzo random polynomial supported on M. Then,*

$$\mathbb{E}_{\mathtt{f}}\mathbb{E}_{\mathtt{f}\in I^n} \mathtt{C}(f,x)^n \leq 2n^2 d^n |M| \left(10\sqrt{n+1}) K_{\mathtt{f}}\rho_{\mathtt{f}}\right)^{n+1}.$$

Proof. By the Fubini-Tonelli theorem, we have

$$\mathbb{E}_{\mathtt{f}}\mathbb{E}_{\mathtt{f}\in I^n} \mathtt{C}(f,x)^n = \mathbb{E}_{\mathtt{f}\in I^n}\mathbb{E}_{\mathtt{f}} \mathtt{C}(f,x)^n,$$

so it is enough to compute $\mathbb{E}_{\mathtt{f}} \mathtt{C}(f,x)^n = \int_1^\infty \mathbb{P}(\mathtt{C}(\mathtt{f},x)^n \geq t)$. The latter, by Theorem 5.3, is bounded from above by

$$e^n \sqrt{n} d^n |M| \left(\frac{8K_{\mathtt{f}}\rho_{\mathtt{f}}}{\sqrt{n}}\right)^{n+1} \int_1^\infty \frac{\ln^{\frac{n+1}{2}} t}{t^{1+\frac{1}{n}}}\, \mathrm{d}t.$$

After straightforward calculations, we obtain

$$\int_1^\infty \frac{\ln^{\frac{n+1}{2}} t}{t^{1+\frac{1}{n}}}\, \mathrm{d}t = n^{\frac{n+3}{2}} \Gamma\left(\frac{n+3}{2}\right) \leq e\sqrt{\pi} n^{\frac{n+4}{2}} \left(\frac{n+1}{2e}\right)^{\frac{n+1}{2}},$$

where Γ is Euler's Gamma function and the second inequality follows from Striling's approximation. Hence, the bound follows. □

We can also bound the *global condition number*, that is

$$\mathtt{C}(f) := \max\{\mathtt{C}(f,x) \mid x \in I^n\}. \tag{5.4}$$

Corollary 5.6. *Let $\mathtt{f} \in \mathcal{P}_{n,d}$ be a zintzo random polynomial supported on M. Then, for all $t > 2e$,*

$$\mathbb{P}(\mathtt{C}(\mathtt{f}) \geq t) \leq \frac{1}{4}\sqrt{n} d^{2n} |M| \left(64 K_{\mathtt{f}}\rho_{\mathtt{f}}\right)^{n+1} \frac{\ln^{\frac{n+1}{2}} t}{t}.$$

Proof. The idea is to use an efficient ε-net of I^n and the 2nd Lipschitz property to turn our local estimates into global ones, as is done in [27, Theorem 1§2 19]. Recall, that an ε-net of I^n (with respect to the ∞-norm) is a finite subset $\mathcal{G} \subseteq I^n$ such that, for all $y \in I^n$, $\mathrm{dist}_\infty(y, \mathcal{G}) \leq \varepsilon$.

Note that for every $\varepsilon > 0$, we have an ε-net $\mathcal{G}_\varepsilon \subseteq I^n$ of size $\leq 2^n \varepsilon^{-n}$. To construct it, we take the uniform grid in the cube.

Now, we notice that if $\mathtt{C}(\mathtt{f}) \geq t$, then

$$\max\{\mathtt{C}(\mathtt{f}, x) \mid x \in \mathcal{G}_{1/(2dt)}\} \geq t/2$$

by the 2nd Lipschitz property (Proposition 4.6). In this way, by the implication and the union bound, we obtain

$$\mathbb{P}(\mathtt{C}(\mathtt{f}) \geq t) \leq \left|\mathcal{G}_{1/(2dt)}\right| \max\{\mathbb{P}(\mathtt{C}(\mathtt{f}, x) \geq t/2) \mid x \in I^n\}.$$

By Theorem 5.3 and the bound on $\left|\mathcal{G}_{1/(2dt)}\right|$, we conclude. □

Now we have all the tools to prove Proposition 2.5 which shows that the smoothed case is included in the above average cases.

Proof of Proposition 2.5. It is enough to show that for $x, s \in \mathbb{R}$ and a random variable $\mathtt{x}$ with subgaussian constant K and anti-concentration constant ρ, $x + s\mathtt{x}$ is a random variable with subgaussian constant $\leq |x| + sK$ and anti-concentration constant $\leq \rho/s$. We note that the latter follows directly from the definition, so we only prove the former.

Now, for all $t \geq |x| + sK$,

$$\mathbb{P}(|x + s\mathtt{x}| \geq t) \leq \mathbb{P}(|\mathtt{x}| \geq (t - |x|)/s) \leq 2\exp(-(t - |x|)^2/(sK)^2).$$

We can easily check that $t \geq |x| + sK$ implies $(t - |x|)/(sK) \geq t/(|x| + sK)$. Hence, the claim follows. □

6 PLANTINGA-VEGTER ALGORITHM

The Plantinga-Vegter algorithm [23] is a subdivision-based algorithm that computes an isotopically correct approximation of the zeros of a univariate polynomial in an interval, of a curve in the plane, or of a surface in 3-dimensional space. Following [6] and [7], we will focus on the subdivision procedure, which is extended for an arbitrary number of variables, and bound the complexity by bounding the number of boxes that the algorithm produces. We

Algorithm 1: PV-Subdivsion

Input : $f \in \mathcal{P}_{n,d}$ which is non-singular in I^n
Output : A subdivision $\mathcal{S}$ of I^n into boxes such that for all $B \in \mathcal{S}$, $C_f(B)$ holds

$\mathcal{S}_0 \leftarrow \{I^n\}, \mathcal{S} \leftarrow \emptyset$;
while $\mathcal{S}_0 \neq \emptyset$ **do**
  Take $B \in \mathcal{S}_0$;
  if $C_f(B)$ *holds* **then**
    $\mathcal{S} \leftarrow \mathcal{S} \cup \{B\}, \mathcal{S}_0 \leftarrow \mathcal{S}_0 \setminus \{B\}$;
  else
    $\mathcal{S}_0 \leftarrow \mathcal{S}_0 \setminus \{B\} \cup$ StandardSubdivsion(B);
return $\mathcal{S}$;

refer to [6], [7] and [27, 5$^{\S 2}$] for further justification of the approach taken here.

Remark 6.1. Even though we present our results for the unit cube I^n, we note that our tools apply for a cube of arbitrary size (up to the technical assumption on the support). To do so, we need to normalize evaluations appropiately by a power of $\max\{1, \|x\|_\infty\}$ for $\|x\|_\infty > 1$. However, this would obfuscate many of the ideas presented in this paper. Hence, for the sake of simplicity, we analyze Algorithm PV-Subdivsion only in the unit cube.

6.1 PV Algorithm and its interval version

The subdivision routine of the PV algorithm, PV-Subdivsion, relies on subdividing the unit cube I^n until each box B in the subdivision satisfies the condition

$$C_f(B) : \text{ either } 0 \notin f(B) \text{ or } 0 \notin \{\langle \nabla_x f, \nabla_y f\rangle \mid x, y \in B\}.$$

To implement this algorithm one uses interval arithmetic. Recall that an *interval approximation* of a map $g : I^n \to \mathbb{R}^q$ is a map $\square[g] : \mathcal{B}(I^n) \to \mathcal{B}(\mathbb{R}^q)$, where $\mathcal{B}(X)$ is the set of (coordinate) boxes contained in X, such that for all $B \in \mathcal{B}(I^n)$, we have

$$g(B) \subseteq \square[g](B).$$

Using the language of Xu and Yap [29], we will consider only the interval level of the algorithm, leaving the effective version to an extended version of this work.

We note that Corollaries 3.5 and 3.6 establish Lipschitz properties for both f and ∇f, with respect the ∞-norm. This is ideal for constructing interval approximations to implement PV-Subdivsion. In our case, our interval approximations will be:

$$\square[f](B) \coloneqq f(m(B)) + d\|f\|_1 w(B)/2[-1, 1] \tag{6.1}$$

and

$$\square[\|\nabla f\|_1](B) \coloneqq \|\nabla_{m(B)} f\|_1 + \sqrt{2n}d(d-1)\|f\|_1 w(B)[-1, 1]. \tag{6.2}$$

For these interval approximations, we can interpret the stopping criterion as follows:

Proposition 6.2. *The condition $C_f(B)$ is implied by the condition*

$$C'_f(B) : \begin{cases} |f(m(B))| & > d\|f\|_1 w(B)/2 \\ \text{or} & \|\nabla_{m(B)} f\|_1 > \sqrt{2n}d(d-1)\|f\|_1 w(B) \end{cases}.$$

Hence, PV-Subdivsion with the interval approximations given in (6.1) *and* (6.2) *is correct if we substitute the condition $C_f(B)$ by*

$$C_f^{\square}(B) : \text{ either } 0 \notin \square[f](B) \text{ or } 0 \notin \square[\|\nabla f\|_1](B).$$

Proof. The statement follows from Corollaries 3.5 and 3.6, [7, Lemma 4.4] and the fact that for $y \in \mathbb{R}^n$, $\|y\|_1/\sqrt{n} \leq \|y\|_2$. □

For now on, the interval version of PV-Subdivsion will be a variant that exploits the interval approximations in (6.1) and (6.2).

6.2 Complexity analysis

As in [6] and [7], our complexity analysis relies on the construction of a local size bound for PV-Subdivsion and the application of the continuous amortization developed by Burr, Krahmer and Yap [4, 5].

We recall the definition of the local size bound and the result that we will exploit in our complexity analysis.

Definition 6.3. A *local size bound* for the interval version of PV-Subdivsion on input f is a function $b_f : I^n \to [0, 1]$ such that for all $x \in \mathbb{R}^n$,

$$b_f(x) \leq \inf\{\text{vol}(B) \mid x \in B \in \mathcal{B}(I^n) \text{ and } C_f^{\square}(B) \text{ false}\}.$$

Theorem 6.4. *[4–6] The number of boxes of the final subdivision of the interval version of PV-Subdivsion on input f is at most*

$$4^n \mathbb{E}_{\mathbf{x} \in I^n}\left(b_f(x)^{-1}\right).$$

Also, the bound is finite if and only if PV-Subdivsion terminates. □

Theorem 6.5. *The function*

$$x \mapsto \left(2d\sqrt{n}\,\mathtt{C}(f, x)\right)^{-n}$$

is a local size bound for PV-Subdivsion on input f.

Proof. Let $x \in B \in \mathcal{B}(I^n)$. Then $\|m(B) - x\|_\infty \leq w(B)/2$ and so, by Corollaries 3.5 and 3.6 and the regularity inequality (Propsoition 4.3), we have that

$$|f(m(B))| > \|f\|_1 \left(\mathtt{C}(f, x)^{-1} - dw(B)/2\right), \tag{6.3}$$

and

$$\|\nabla_{m(B)} f\|_1 > d\|f\|_1 \left(\mathtt{C}(f, x)^{-1} - (d-1)w(B)/2\right). \tag{6.4}$$

Hence, $C_f(B)$ is true as long as, either $\mathtt{C}(f, x)^{-1} \geq dw(B)$, or $\mathtt{C}(f, x)^{-1} > 2\sqrt{n}dw(B)$. The result follows, since $\text{vol}(B) = w(B)^n$. □

Theorem 6.4 and Theorem 6.5 result the following corollary:

Corollary 6.6. *The number of boxes of the final subdivision of the interval version of PV-Subdivsion on input f is at most*

$$8^n n^{\frac{n}{2}} d^n \mathbb{E}_{\mathbf{x} \in I^n} \mathtt{C}(f, \mathbf{x})^n.$$

Theorem 2.10 follows now from Corollaries 5.5 and 6.6.

Remark 6.7. A similar argument as in the proof of [7, Theorem 6.4] shows that we can bound the local size bound of [6] in terms of $1/\mathtt{C}(f, x)^n$. Since the interval approximation of the analyzed version is simpler, requiring a single evaluation, we only analyze the complexity of this.

Remark 6.8. Our tools apply for a cube of arbitrary size (up to the technical assumption on the support). To do so, we need to normalize evaluations by a power of $\max\{1, \|x\|_\infty\}$ for $\|x\|_\infty > 1$. However, this would obfuscate many of the ideas presented. Hence, for the sake of simplicity, we restrict our analysis to the unit cube.

7 CONDITION AND SEPARATION BOUNDS

The following theorem is a variation of a result due to Dedieu [11, Theorem 3.2 and Theorem 5.1]. It relates the condition number with the separation bound, that is the minimum distance between the roots, in the univariate case.

Theorem 7.1. *Let $f \in \mathcal{P}_{1,d}$ be a univariate polynomial and $x \in I$. Then, for any two distinct and non-singular roots, α and $\tilde{\alpha}$, such that $\alpha, \tilde{\alpha} \in B_{\mathbb{C}}(x, 1/(2(d-1)\,\mathtt{C}(f,x)))$,*

$$|\alpha - \tilde{\alpha}| \geq 1/(16(d-1)\,\mathtt{C}(f,x)).$$

Proof. By [12, Théorème 91], the Newton method converges for any point in $B_{\mathbb{C}}(\alpha, 1/(6\gamma(f,\alpha))$, where γ is Smale's gamma. This means that for any two roots α and $\tilde{\alpha}$ of f, we must have

$$|\alpha - \tilde{\alpha}| \geq 1/\max\{3\gamma(f,\alpha), 3\gamma(f,\tilde{\alpha})\}.$$

Now, by [12, Lemme 98], for any $y \in B_{\mathbb{C}}(x, 1/(4\gamma(f,x)))$,

$$\gamma(f,y) \leq 32\gamma(f,x)/3.$$

Hence, for any distinct roots $\alpha, \tilde{\alpha} \in B_{\mathbb{C}}(x, 1/(4\gamma(f,x)))$ that are not singular, and because Smale's gamma is finite at them, we have

$$|\alpha - \tilde{\alpha}| \geq 1/(32\gamma(f,x)).$$

Using the Higher Derivative Estimate (Prop. 4.7) we conclude. □

Recall that the local separation at a root α is given by $\Delta_\alpha := \min_{\beta \in f^{-1}(0)\setminus\{\alpha\}} |\alpha - \beta|$. The following corollary controls the local separation of the roots near an interval I.

Corollary 7.2. *Let $f \in \mathcal{P}_{1,d}$. Then, for every complex $\alpha \in f^{-1}(0)$ such that $\operatorname{dist}(\alpha, I) \leq 1/(3(d-1)\,\mathtt{C}(f))$,*

$$\Delta_\alpha \geq 1/(16(d-1)\,\mathtt{C}(f)).$$

Corollary 7.2 together with Corollary 5.6 allows us to give probabilistic estimates of the separation bound for roots that lie near the unit interval.

Acknowledgements. Both authors are grateful to Alperen Ergür for various discussions and suggestions. The first author is grateful to Evgenia Lagoda for moral support. Both authors are partially supported by ANR JCJC GALOP (ANR-17-CE40-0009), the PGMO grant ALMA, and the PHC GRAPE.

REFERENCES

[1] Diego Armentano and Carlos Beltrán. 2019. The polynomial eigenvalue problem is well conditioned for random inputs. *SIAM J. Matrix Anal. Appl.* 40, 1 (2019), 175–193. https://doi.org/10.1137/17M1139941

[2] Carlos Beltrán and Khazhgali Kozhasov. 2019. The Real Polynomial Eigenvalue Problem is Well Conditioned on the Average. *Foundations of Computational Mathematics* On-line First (2019), 19. https://doi.org/10.1007/s10208-019-09414-2

[3] Peter Bürgisser and Felipe Cucker. 2013. *Condition.* Grundlehren der mathematischen Wissenschaften, Vol. 349. Springer-Verlag, Berlin. https://doi.org/10.1007/978-3-642-38896-5

[4] Michael Burr, Felix Krahmer, and Chee Yap. 2009. Continuous amortization: A non-probabilistic adaptive analysis technique. *Electronic Colloquium on Computational Complexity* 16, Report. No. 136 (Dec. 2009), 22.

[5] Michael A. Burr. 2016. Continuous amortization and extensions: with applications to bisection-based root isolation. *J. Symbolic Comput.* 77 (2016), 78–126. https://doi.org/10.1016/j.jsc.2016.01.007

[6] Michael A. Burr, Shuhong Gao, and Elias P. Tsigaridas. 2017. The complexity of an adaptive subdivision method for approximating real curves. In *ISSAC'17—Proceedings of the 2017 ACM International Symposium on Symbolic and Algebraic Computation.* ACM, New York, 61–68. https://doi.org/10.1145/3087604.3087654

[7] Felipe Cucker, Alperen A. Ergür, and Josué Tonelli-Cueto. 2019. Plantinga-Vegter Algorithm Takes Average Polynomial Time. In *Proceedings of the 2019 on International Symposium on Symbolic and Algebraic Computation (ISSAC '19).* ACM, New York, Beijing, China, 114–121. https://doi.org/10.1145/3326229.3326252

[8] Felipe Cucker, Alperen A. Ergür, and Josué Tonelli-Cueto. 2020. Functional norms, condition numbers and numerical algorithms in algebraic geometry. Manuscript.

[9] Felipe Cucker, Alperen A. Ergür, and Josué Tonelli-Cueto. 2020. On the Complexity of the Plantinga-Vegter Algorithm. arXiv:2004.06879.

[10] Felipe Cucker, Teresa Krick, Gregorio Malajovich, and Mario Wschebor. 2008. A numerical algorithm for zero counting. I: Complexity and accuracy. *J. Complexity* 24 (2008), 582–605. https://doi.org/10.1016/j.jco.2008.03.001

[11] Jean-Pierre Dedieu. 1997. Estimations for the Separation Number of a Polynomial System. *Journal of Symbolic Computation* 24, 6 (Dec. 1997), 683–693.

[12] Jean-Pierre Dedieu. 2006. *Points fixes, zéros et la méthode de Newton.* Mathématiques & Applications (Berlin) [Mathematics & Applications], Vol. 54. Springer, Berlin. xii+196 pages.

[13] Alperen A. Ergür, Grigoris Paouris, and J. Maurice Rojas. 2018. Probabilistic Condition Number Estimates for Real Polynomial Systems II: Structure and Smoothed Analysis. (Sept. 2018), 22 pages. arXiv:1809.03626.

[14] Alperen A. Ergür, Grigoris Paouris, and J. Maurice Rojas. 2019. Probabilistic Condition Number Estimates for Real Polynomial Systems I: A Broader Family of Distributions. *Found. Comput. Math.* 19, 1 (2019), 131–157. https://doi.org/10.1007/s10208-018-9380-5

[15] Gorav Jindal and Michael Sagraloff. 2017. Efficiently computing real roots of sparse polynomials. In *ISSAC'17—Proceedings of the 2017 ACM International Symposium on Symbolic and Algebraic Computation.* ACM, New York, 229–236. https://doi.org/10.1145/3087604.3087652

[16] Askold G. Khovanskiĭ. 1991. *Fewnomials.* Translations of Mathematical Monographs, Vol. 88. American Mathematical Society, Providence, RI. viii+139 pages. Trans. from the Russian by S. Zdravkovska.

[17] Pierre Lairez. 2017. A deterministic algorithm to compute approximate roots of polynomial systems in polynomial average time. *Found. Comput. Math.* 17, 5 (2017), 1265–1292. https://doi.org/10.1007/s10208-016-9319-7

[18] Galyna Livshyts, Grigoris Paouris, and Peter Pivovarov. 2016. On sharp bounds for marginal densities of product measures. *Israel Journal of Mathematics* 216, 2 (2016), 877–889. https://doi.org/10.1007/s11856-016-1431-5

[19] Gregorio Malajovich. 2019. Complexity of sparse polynomial solving: homotopy on toric varieties and the condition metric. *Found. Comput. Math.* 19, 1 (2019), 1–53. https://doi.org/10.1007/s10208-018-9375-2

[20] Gregorio Malajovich. 2020. Complexity of Sparse Polynomial Solving 2: Renormalization. (May 2020), 84 pages. arXiv:2005.01223.

[21] Gregorio Malajovich and J. Maurice Rojas. 2002. Polynomial systems and the momentum map. In *Foundations of computational mathematics (Hong Kong, 2000).* World Sci. Publ., River Edge, NJ, 251–266.

[22] Gregorio Malajovich and J. Maurice Rojas. 2004. High probability analysis of the condition number of sparse polynomial systems. *Theoret. Comput. Sci.* 315, 2-3 (2004), 524–555. https://doi.org/10.1016/j.tcs.2004.01.006

[23] Simon Plantinga and Gert Vegter. 2004. Isotopic Approximation of Implicit Curves and Surfaces. In *Proceedings of the 2004 Eurographics/ACM SIGGRAPH Symposium on Geometry Processing (SGP '04).* ACM, New York, NY, USA, 245–254. https://doi.org/10.1145/1057432.1057465

[24] J. Renegar. 1987. On the efficiency of Newton's method in approximating all zeros of a system of complex polynomials. *Math. Oper. Res.* 12, 1 (1987), 121–148. https://doi.org/10.1287/moor.12.1.121

[25] Mark Rudelson and Roman Vershynin. 2015. Small ball probabilities for linear images of high-dimensional distributions. *Int. Math. Res. Not. IMRN* 19 (2015), 9594–9617. https://doi.org/10.1093/imrn/rnu243

[26] Daniel A. Spielman and Shang-Hua Teng. 2002. Smoothed Analysis of Algorithms. In *Proceedings of the International Congress of Mathematicians, Vol. I (Beijing, 2002).* Higher Ed. Press, Beijing, 597–606.

[27] Josué Tonelli-Cueto. 2019. *Condition and Homology in Semialgebraic Geometry.* Doctoral Thesis. Technische Universität Berlin, DepositOnce Repository. https://doi.org/10.14279/depositonce-9453

[28] Roman Vershynin. 2018. *High-dimensional probability: An introduction with applications in data science.* Cambridge Series in Statistical and Probabilistic Mathematics, Vol. 47. Cambridge University Press, Cambridge. https://doi.org/10.1017/9781108231596

[29] Juan Xu and Chee Yap. 2019. Effective subdivision algorithm for isolating zeros of real systems of equations, with complexity analysis. In *ISSAC'19—Proceedings of the 2019 ACM International Symposium on Symbolic and Algebraic Computation.* ACM, New York, 355–362.

An Extended GCD Algorithm for Parametric Univariate Polynomials and Application to Parametric Smith Normal Form

Dingkang Wang
[1]KLMM, Academy of Mathematics and Systems Science, Chinese Academy of Sciences
Beijing 100190, China
[2]School of Mathematical Sciences, University of Chinese Academy of Sciences
Beijing, China
dwang@mmrc.iss.ac.cn

Hesong Wang
[1]KLMM, Academy of Mathematics and Systems Science, Chinese Academy of Sciences
Beijing 100190, China
[2]School of Mathematical Sciences, University of Chinese Academy of Sciences
Beijing, China
wanghesong2021@gmail.com

Fanghui Xiao
[1]KLMM, Academy of Mathematics and Systems Science, Chinese Academy of Sciences
Beijing 100190, China
[2]School of Mathematical Sciences, University of Chinese Academy of Sciences
Beijing, China
xiaofanghui@amss.ac.cn

ABSTRACT

An extended greatest common divisor (GCD) algorithm for parametric univariate polynomials is presented in this paper. This algorithm computes not only the GCD of parametric univariate polynomials in each constructible set but also the corresponding representation coefficients (or multipliers) for the GCD expressed as a linear combination of these parametric univariate polynomials. The key idea of our algorithm is that for non-parametric case the GCD of arbitrary finite number of univariate polynomials can be obtained by computing the minimal Gröbner basis of the ideal generated by those polynomials. But instead of computing the Gröbner basis of the ideal generated by those polynomials directly, we construct a special module by adding the unit vectors which can record the representation coefficients, then obtain the GCD and representation coefficients by computing a Gröbner basis of the module. This method can be naturally generalized to the parametric case because of the comprehensive Gröbner systems for modules. As a consequence, we obtain an extended GCD algorithm for parametric univariate polynomials. More importantly, we apply the proposed extended GCD algorithm to the computation of Smith normal form, and give the first algorithm for reducing a univariate polynomial matrix with parameters to its Smith normal form.

CCS CONCEPTS

• **Computing methodologies** → **Symbolic and algebraic algorithms**; **Algebraic algorithms**;

KEYWORDS

Extended greatest common divisor, Parametric univariate polynomial, Comprehensive Gröbner system, Smith normal form

ISSAC '20, July 20–23, 2020, Kalamata, Greece

ACM ISBN 978-1-4503-7100-1/20/07…$15.00
https://doi.org/10.1145/3373207.3404019

ACM Reference Format:
Dingkang Wang, Hesong Wang, and Fanghui Xiao. 2020. An Extended GCD Algorithm for Parametric Univariate Polynomials and Application to Parametric Smith Normal Form. In *International Symposium on Symbolic and Algebraic Computation (ISSAC '20), July 20–23, 2020, Kalamata, Greece.* ACM, New York, NY, USA, 8 pages. https://doi.org/10.1145/3373207.3404019

1 INTRODUCTION

The computation of polynomial greatest common divisor (GCD) is one of the most primitive computations in computer algebra with a wide range of applications that include simplifying rational expressions, partial fraction expansions, canonical transformations, mechanical geometry theorem proving, hybrid rational function approximation, and decoder implementation for error-correction; see [7, 10, 15, 17, 38]. It has been extensively studied and a crowd of algorithms have been constructed [8, 16, 23, 37]. Among them Euclidean algorithm which is the oldest algorithm for computing the GCD of two univariate polynomials and its variants are the most common algorithms. As an extension of polynomial GCD, parametric GCDs came into being. That is, the parameters space is decomposed into a finite number of constructible sets such that a GCD of the parametric polynomials is given uniformly in each constructible set. Abramov and Kvashenko [1] proposed an algorithm for computing the parametric GCD of two univariate polynomials with one parameter using sub-resultant chain. Ayad [2] studied the parametric GCD of several univariate polynomials with many parameters and mainly introduced two algorithms to compute the parametric GCD. Also with the idea of the comprehensive Gröbner system (CGS) introduced by Weispfenning [36], Nagasaka [27] extended the theories of Gianni and Trager [16], and Sasaki and Suzuki [31] which compute the GCD by Gröbner bases method to multivariate polynomials with parameters. Kapur et al. [19] proposed another algorithm for computing the parametric GCD of parametric multivariate polynomials. Besides, based on triangular set methods, Chen and Maza [9], and Bächler et al. [3] used subresultant chains and regular chains to compute parametric GCDs.

As for the extended polynomial GCD computation, of course it is also an important problem in symbolic algebraic computation and applications. To our knowledge, for non-parametric univariate polynomials, there are two kinds of algorithms to compute the extended GCD. One is the well-known extended Euclidean algorithm,

and the other is the algorithm for solving the extended GCD problem by means of Hankel matrix techniques which was proposed by Sendra and Llovet [32]. However, there is currently no algorithm for computing the extended parametric polynomial GCD.

In this paper, we present an algorithm for computing the extended GCD of parametric univariate polynomials. We begin to present our key idea from non-parametric case, then extend the method for computing the extended GCD of univariate polynomials to the parametric case.

As we known, the GCD d of univariate polynomials $f_1, \ldots, f_s$ can be obtained by computing the minimal Gröbner basis of the ideal $\langle f_1, \ldots, f_s \rangle$. To get the representation coefficients (or multipliers) $a_1, \ldots, a_s$ for the GCD expressed as a linear combination: $d = a_1 f_1 + \cdots + a_s f_s$, we construct a module generated by s column vectors $(f_1, \epsilon_1)^T, \ldots, (f_s, \epsilon_s)^T$, where $\{\epsilon_1, \ldots, \epsilon_s\}$ is a standard basis for s-dimensional vector space. Under the proper position over term (POT) monomial order, one computes a minimal Gröbner basis of this module in which there exists only one element $(d', a'_1, \ldots, a'_s)$ such that d' is nonzero. These are exactly what we want, i.e. $d = d'$ and $a_i = a'_i$ for $i = 1, \ldots, s$. Most importantly, using comprehensive Gröbner systems for modules which presented by Nabeshima [26] as the generalization of comprehensive Gröbner systems for polynomial rings studied by Weispfenning [36], this method can be naturally extended to the parametric case. Meanwhile, we also get a free basis for the syzygy module of given polynomials $f_1, \ldots, f_s$ as a by-product.

In the rest of this paper, we will apply the proposed extended GCD algorithm to the computation of the Smith normal form together with transforming matrices, which is different from the method presented by Storjohann in [33] for computing the Smith normal form and transforming matrices of an integer matrix using the modulo N extended GCD algorithm. The reduction of univariate polynomial matrices to the Smith normal form is very useful in many areas of system theory, for instance, the analysis and minimal realization of transfer function matrices of time-invariant linear dynamical systems [7, 30], and the existence of a solution to an integer programming problem [4]. A constructive proof of the uniqueness of the Smith form is given by Gantmakher [14]. This construction gives a basic algorithm for Smith form reduction and many other algorithms [6, 29] based on this have been proposed with the view to improving efficiency.

An essential step in the calculation of the Smith normal form is the calculation of the GCD and multipliers for each of its rows and columns. In order to get the GCD of each column (row), the algorithms in [6, 29] have to subtract multiples of the least degree polynomial in the corresponding column (row) of matrices, at any instant, from the others, until only one non-zero polynomial remains. The proposed extended GCD algorithm in this paper, however, can give the GCD and multipliers directly. What's more, our algorithm can be extended to parametric case naturally, which is, to our knowledge, the first algorithm for computing the Smith normal form of polynomial matrices with parameters. Also, it's worth mentioning that Corless et al. [11] presented an algorithm for computing the Jordan canonical form of a matrix in Frobenius (rational) canonical form where entries are polynomials with parameters.

This paper is organized as follows. In Section 2, we introduce some notations and definitions. The main results is presented in Section 3, where we start from the non-parametric case, giving the method for computing the extended GCD of univariate polynomials and extending this result to the parametric case. Consequently the extended GCD algorithm for parametric univariate polynomials is presented. In Section 4, we apply the proposed algorithm to the computation of Smith normal form. We end with some concluding remarks in Section 5.

2 PRELIMINARIES

In this section we will introduce some notations and definitions to prepare for the discussion of this article.

Let k be a field, L be an algebraic closed field containing k, $R = k[x]$ be the polynomial ring in the variable x (or $R = k[U][x]$ be the parametric polynomial ring with the parameters $U = \{u_1, \ldots, u_m\}$ and variable x). Generally, we use the letters f, g, h for single polynomials (or elements of the ring $k[x]$) and boldface letters $\mathbf{e}, \mathbf{f}, \mathbf{g}, \mathbf{h}$ for column vectors (that is, elements of the module $k[x]^s$).

In practice, we frequently consider such a very important class of modules as follows.

Definition 2.1. Let $(f_1, \ldots, f_s)$ be an ordered $s-$tuple with $f_i \in R$. The set of all $(a_1, \ldots, a_s)^T \in R^s$ such that $a_1 f_1 + \cdots + a_s f_s = 0$ is an R-submodule of R^s, called the **syzygy module** of $(f_1, \ldots, f_s)$, and denoted by $\mathrm{Syz}(f_1, \ldots, f_s)$.

Unlike vector spaces, modules need not have any generating set which is linearly independent. If a R-module have a module basis, that is, a generating set that is R-linearly independent, it is given a special name, **free module**.

For example, the R-module R^s is free. Let $\epsilon_1 = (1, 0, \cdots, 0)^T$, $\epsilon_2 = (0, 1, \cdots, 0)^T$, $\cdots$, $\epsilon_s = (0, 0, \cdots, 1)^T$, then $\{\epsilon_1, \cdots, \epsilon_s\}$ is a free basis of R^s.

Next, we introduce Gröbner bases and comprehensive Gröbner systems for modules.

Let $\succ$ be a monomial order on $k[x]$, and $\succ_s$ be a module order by extending $\succ$ in a position over term (POT) fashion to $k[x]^s$, that is, for $\alpha,\ \beta \in \mathbb{N}$, $x^\alpha \epsilon_i \succ_s x^\beta \epsilon_j$ if $i > j$, or $i = j$ and $x^\alpha \succ x^\beta$. For $f \in k[x]$, $\mathbf{g} \in k[x]^s$, the leading term, leading coefficient, and leading monomial of f and $\mathbf{g}$ with respect to $\succ$ and $\succ_s$ respectively are conveniently denoted by $\mathrm{LT}(f)$, $\mathrm{LC}(f)$, $\mathrm{LM}(f)$, $\mathrm{LT}(\mathbf{g})$, $\mathrm{LC}(\mathbf{g})$, and $\mathrm{LM}(\mathbf{g})$.

The definition of Gröbner bases for submodules is as follows.

Definition 2.2. Let $R = k[x]$ and M be a submodule of R^s, and let $\succ_s$ be a monomial order on $k[x]^s$.

(1) We will denote by $\langle \mathrm{LT}(M) \rangle$ the monomial submodule generated by the leading terms of all $\mathbf{g} \in M$ w.r.t. $\succ_s$.
(2) A finite collection $G = \{\mathbf{g}_1, \ldots, \mathbf{g}_t\} \subset M$ is called a **Gröbner basis** for M if $\langle \mathrm{LT}(M) \rangle = \langle \mathrm{LT}(\mathbf{g}_1), \cdots, \mathrm{LT}(\mathbf{g}_t) \rangle$.

The following are about the definitions of minimal and reduced Gröbner bases for modules.

Definition 2.3. Let $G = \{\mathbf{g}_1, \ldots, \mathbf{g}_t\}$ be a Gröbner basis for $M \subset k[x]^s$ with respect to a monomial order $\succ_s$.

(1) G is said to be **minimal**, if $\mathrm{LM}(\mathbf{g}) \notin \langle \mathrm{LM}(G \setminus \{\mathbf{g}\}) \rangle$ for all $\mathbf{g} \in G$.

(2) G is said to be **reduced**, if $\mathrm{LC}(\mathbf{g}) = 1$ and no monomial of $\mathbf{g}$ lies in $\langle \mathrm{LM}(G\backslash\{\mathbf{g}\})\rangle$.

Now we introduce some definitions for parametric univariate polynomials. For $\mathbf{g} \in k[U][x]^s$, $\mathrm{LC}_x(\mathbf{g})$ denotes the leading coefficient of $\mathbf{g}$ with respect to the variable x under the order $\succ_s$.

A **specialization** of $k[U]$ is a homomorphism $\sigma : k[U] \to L$. In this paper, we only consider the specializations induced by the elements in L^m. That is, for $\alpha = (\alpha_1, \ldots, \alpha_m) \in L^m$, the induced specialization σ_α is defined as

$$\sigma_\alpha : f \to f(\alpha),$$

where $f \in k[U]$. Every specialization σ: $k[U] \to L$ extends canonically to a specialization σ: $k[U][x]^s \to L[x]^s$ by applying σ coefficient-wise.

For an ideal $E \subset k[U]$, the variety defined by E in L^m is denoted by $\mathbb{V}(E) = \{\alpha \in L^m \mid f(\alpha) = 0 \text{ for all } f \in E\}$. $A = \mathbb{V}(E) \setminus \mathbb{V}(N)$ is an algebraically constructible set, where E, N are ideals in $k[U]$.

For parametric systems, the definitions of comprehensive Gröbner systems and minimal comprehensive Gröbner systems for modules are given below.

Definition 2.4. Let F be a subset of $k[U][x]^s$, S be a subset of L^m, $G_1, \ldots, G_l$ be subsets of $k[U][x]^s$, and $A_1, \ldots, A_l$ be algebraically constructible subsets of L^m such that $S = \bigcup_{i=1}^{l} A_i$. A finite set $\mathcal{G} = \{(A_1, G_1), \ldots, (A_l, G_l)\}$ is called a **comprehensive Gröbner system** (CGS) on S for F if $\sigma_\alpha(G_i)$ is a Gröbner basis of the submodule $\langle \sigma_\alpha(F)\rangle \subset L[x]^s$ for $\alpha \in A_i$ and $i = 1, \ldots, l$. Each (A_i, G_i) is called a branch of $\mathcal{G}$. In particular, if $S = L^m$, then $\mathcal{G}$ is called a comprehensive Gröbner system for F.

Definition 2.5. A comprehensive Gröbner system $\mathcal{G} = \{(A_1, G_1), \cdots, (A_l, G_l)\}$ on S for $M \subset k[U][x]^s$ is said to be **minimal (reduced)** under some monomial order $\succ_s$, if for each $i = 1, \ldots, l$,

(1) $A_i \neq \emptyset$, and furthermore, for each $i, j = 1, \cdots, l$, $A_i \cap A_j = \emptyset$ whenever $i \neq j$, and
(2) $\sigma_\alpha(G_i)$ is a minimal (reduced) Gröbner basis of $\langle \sigma_\alpha(F)\rangle \subset L[x]^m$ for $\alpha \in A_i$, and
(3) for each $\mathbf{g} \in G_i \neq \{\mathbf{0}\}$, $\sigma_\alpha(\mathrm{LC}_x(\mathbf{g})) \neq 0$ for $\alpha \in A_i$.

Remark 1. *For the computation of CGSs for modules, there exists an algorithm given by Nabeshima[26] which is based on the results proposed by Suzuki and Sato [35]. Moreover, there exist various algorithms to compute the minimal CGS for polynomial rings; see [18, 22, 24, 25, 34, 35] and so on. These algorithms can be extended to the case of modules. In this paper, we extend the KSW algorithm for computing CGSs over polynomial rings presented by Kapur et al. [20, 21] to the case of modules and then compute CGSs for modules since the KSW algorithm generates fewer branches and is the most efficient algorithm so far.*

Finally, we introduce the GCD systems for parametric univariate polynomials.

Definition 2.6. Let $F = \{f_1, \cdots, f_s\}$ be a subset of $k[U][x]$, S be a subset of L^m and $d_1, \ldots, d_l$ be parametric univariate polynomials in $k[U][x]$, and $A_1, \ldots, A_l$ be algebraically constructible subsets of L^m such that $S = \bigcup_{i=1}^{l} A_i$ and $A_i \cap A_j = \emptyset$ for $i \neq j$. A finite set $\mathcal{D} = \{(A_1, d_1), \ldots, (A_l, d_l)\}$ is called a **GCD system** on S for F if $\sigma_\alpha(d_i)$ is a GCD of $\sigma_\alpha(F) \subset L[x]$ for $\alpha \in A_i$ and $i = 1, \ldots, l$. Moreover, for each $d_i \neq 0$, $\sigma_\alpha(\mathrm{LC}_x(d_i)) \neq 0$ for $\alpha \in A_i$. Each (A_i, d_i) is regarded as a branch of $\mathcal{D}$. In particular, $\mathcal{D}$ is simply called a GCD system for F if $S = L^m$.

3 THE PROPOSED ALGORITHM

As stated in the introduction, there is currently no algorithm for computing extended GCD of parametric univariate polynomials.

In this section, we are devoted to giving an extended GCD algorithm for parametric univariate polynomials. Since the algorithm based on Gröbner bases is more suitable to be generalized to the parametric case because of the CGS, by means of structural features of the module and by constructing a special module we compute the GCD and obtain an extended GCD algorithm based on the computation of Gröbner bases for modules, which can be naturally generalized to the parametric case.

Now, let us introduce what is to be stated in this section. We first present the key idea for computing the extended GCD of any finite number of non-parametric univariate polynomials, and then generalize it to the parametric case. As a consequence, we propose an algorithm based on CGSs for modules to compute the extended GCD system for a set of parametric univariate polynomials.

3.1 Extended GCD for univariate polynomials

Let $R = k[x]$ and $f_1, \cdots, f_s \in R$. Assume $d = \mathrm{GCD}(f_1, \cdots, f_s)$. Since R is a principal ideal domain (PID), then there are $a_1, \ldots, a_s \in R$ such that $a_1 f_1 + \cdots + a_s f_s = d$, and we call $a_1, \ldots, a_s$ **representation coefficients** for the GCD (not unique).

As we all know, one can obtain a GCD d by computing a Gröbner basis of the ideal generated by $f_1, \cdots, f_s$. Nevertheless, in many case we have to solve the problem: how can we get $a_1, \ldots, a_s$ and d simultaneously? Next, we share our approach.

Before presenting the main theorem, there are several lemmas to be rendered. For the first lemma below, we can refer to [13].

Lemma 3.1. *Let $R = k[x]$ and suppose that $f_1, \ldots, f_s \in R$ are polynomials that are not all zero. Then $\mathrm{Syz}(f_1, \ldots, f_s)$ is a free module with $s - 1$ generators.*

Therefore, the syzygy module M over $R = k[x]$ as a free module has two sets of bases: the free basis and the Gröbner basis under some monomial order, denoted by F and G respectively. Generally speaking, $|G| \geq |F|$, where "$|\cdot|$" represents the number of elements in the set. The proof is as follows.

Lemma 3.2. *Let $M \subset R^s$ be a free R-module, F and G be a free basis and a minimal Gröbner basis under some monomial order $\succ_s$ for M. Then $|G| \geq |F|$.*

Here we construct a module M and let's take a look at some of the properties of this module, which is from Exercise 15 of Chapter 5, Section 3 in [12].

Proposition 3.3. *Let $R' = k[x_1, \ldots, x_n]$ be a polynomial ring with a monomial order $\succ$, and for any integer $s \geq 1$, we denote the standard basis of R'^{s+1} by $\mathbf{e}_1, \mathbf{e}_2, \ldots, \mathbf{e}_{s+1}$. Let $\succ_{s+1}$ denote the POT extension of $\succ$ to R'^{s+1} with $\mathbf{e}_1 \succ \mathbf{e}_i$ for $2 \leq i \leq s+1$. Given $f_1, \ldots, f_s \in R'$, without loss of generality, assume that $f_1, \ldots, f_s$ are not all zero. Then consider the submodule $M \subset R'^{s+1}$ generated by*

$$\mathbf{m}_i = f_i \mathbf{e}_1 + \mathbf{e}_{i+1} = (f_i, 0, \cdots, 0, 1, 0, \cdots, 0)^T, \quad i = 1, \cdots, s.$$

Let G be a minimal Gröbner basis of M with respect to $\succ_{s+1}$, then the following conclusions hold:

1. *If $(g, h_1, \ldots, h_s)^T \in M$, then $g = h_1 f_1 + \cdots + h_s f_s$.*
2. *$M \cap (\{0\} \times R'^s) = \{0\} \times Syz(f_1, \ldots, f_s)$.*
3. *The set $G' = \{g \in R' | \ g \neq 0 \ \wedge \exists\, h_1, \ldots, h_s \in R' \ s.t. \ (g, h_1, \ldots, h_s)^T \in G\}$ is a minimal Gröbner basis with respect to $\succ$ for the ideal $\langle f_1, \ldots, f_s \rangle$.*
4. *The set G'' defined by $\{0\} \times G'' = G \cap (\{0\} \times R'^s)$ is a minimal Gröbner basis with respect to $\succ_s$ being the restriction of $\succ_{s+1}$ to R'^s for the syzygy module $Syz(f_1, \ldots, f_s)$.*

Proof. According to the construction of M, (1) and (2) are obvious. Besides, (3) and (4) are also obtained by the definition of G', G'', and Gröbner bases for modules w.r.t. $\succ_{s+1}$. □

In particular, for the case of univariate, there are better results.

Theorem 3.4. *With the above notations. If $R' = R = k[x]$ is a univariate polynomial ring, then $|G'| = 1$ and $|G''| = s-1$. Therefore, $|G| = s$. Note that s is the number of these given polynomials.*

Proof. First, it follows from (3) of Proposition 3.3 and the univariate polynomial ring R' that $|G'| = 1$.

Now we prove that $|G''| = s-1$. By Lemma 3.1 and Lemma 3.2, we have $|G''| \geq s-1$. In the following all we need to do is to prove that $|G''| > s-1$ is impossible. Let $|G''| = t$ and $G'' = \{\mathbf{g}_1'', \ldots, \mathbf{g}_t''\}$ where $\mathbf{g}_1'' \succ_s \cdots \succ_s \mathbf{g}_t''$. Suppose that $t > s-1$, i.e. $t \geq s$. By Proposition 3.3 we know that G'' is the minimal Gröbner basis for $\mathrm{Syz}(f_1, \ldots, f_s)$, hence $\mathbf{g}_t''$ must be in the form: $\mathbf{g}_t'' = (0, \cdots, 0, g)^T$ where $g \in k[x]$ and $g \neq 0$ because R' is a univariate polynomial ring. This contradicts $\mathbf{g}_t'' \in \mathrm{Syz}(f_1, \ldots, f_s)$, so $|G''| = s-1$. □

Combining Lemma 3.1 and Theorem 3.4, it is easy to know that G'' is a free basis for the syzygy module $\mathrm{Syz}(f_1, \ldots, f_s)$ where $f_1, \ldots, f_s \in k[x]$.

Theorem 3.5. *As above, assume $G = \{\mathbf{g}_1, \ldots, \mathbf{g}_s\}$ is a minimal Gröbner basis for $M \subset k[x]^{s+1}$ under the order $\succ_{s+1}$ with $\mathbf{e}_1 \succ \mathbf{e}_i$ for $2 \leq i \leq s+1$, and $\mathbf{g}_1 = (d, u_{11}, \ldots, u_{1s})^T$, $\mathbf{g}_j = (0, u_{j1}, \ldots, u_{js})^T$, $2 \leq j \leq s$. Then d is a GCD of $f_1, \ldots, f_s$ and $u_{11}, \ldots, u_{1s}$ are the corresponding representation coefficients for d as a linear combination of $f_1, \ldots, f_s$. Further, the matrix $\mathbf{U} = (u_{ij})_{s\times s} \in k[x]^{s\times s}$ is unimodular, that is, $\det(\mathbf{U}) \in k \setminus \{0\}$, and $\mathbf{Uf} = \mathbf{d}$, where*

$$\mathbf{U} = \begin{pmatrix} u_{11} & \ldots & u_{1s} \\ u_{21} & \ldots & u_{2s} \\ \vdots & & \vdots \\ u_{s1} & \ldots & u_{ss} \end{pmatrix}, \quad \mathbf{f} = \begin{pmatrix} f_1 \\ f_2 \\ \vdots \\ f_s \end{pmatrix}, \quad \mathbf{d} = \begin{pmatrix} d \\ 0 \\ \vdots \\ 0 \end{pmatrix}.$$

Proof. According to Proposition 3.3 and Theorem 3.4, $G' = \{d\}$ is a Gröbner basis of the ideal $\langle f_1, \ldots, f_s \rangle$, then d is a GCD of $f_1, \ldots, f_s$ and $u_{11}, \ldots, u_{1s}$ are the corresponding representation coefficients. Moreover, by the construction of the matrix $\mathbf{U}$, it's obvious that $\mathbf{Uf} = \mathbf{d}$. Now let's prove that $\mathbf{U}$ is a unimodular matrix. Since $G = \{\mathbf{g}_1, \ldots, \mathbf{g}_s\}$ is the minimal Gröbner basis for M, hence these generators $\mathbf{m}_1, \ldots, \mathbf{m}_s$ of M can be represented by $\mathbf{g}_1, \ldots, \mathbf{g}_s$. In other words, there exists matrix $\mathbf{V} \in k[x]^{s\times s}$ such that $(\mathbf{m}_1, \ldots, \mathbf{m}_s)^T = \mathbf{V}(\mathbf{g}_1, \ldots, \mathbf{g}_s)^T$. To make things clearer, let's write out $(\mathbf{m}_1, \ldots, \mathbf{m}_s)^T$ and $(\mathbf{g}_1, \ldots, \mathbf{g}_s)^T$ concretely.

$$\begin{pmatrix} \mathbf{m}_1^T \\ \mathbf{m}_2^T \\ \vdots \\ \mathbf{m}_s^T \end{pmatrix} = \begin{pmatrix} f_1 & 1 & 0 & \ldots & 0 \\ f_2 & 0 & 1 & \ldots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ f_s & 0 & 0 & \ldots & 1 \end{pmatrix} \begin{pmatrix} \mathbf{g}_1^T \\ \mathbf{g}_2^T \\ \vdots \\ \mathbf{g}_s^T \end{pmatrix} = \begin{pmatrix} d & u_{11} & \ldots & u_{1s} \\ 0 & u_{21} & \ldots & u_{2s} \\ \vdots & \vdots & \vdots & \vdots \\ 0 & u_{s1} & \ldots & u_{ss} \end{pmatrix}.$$

By $(\mathbf{m}_1, \ldots, \mathbf{m}_s)^T = \mathbf{V}(\mathbf{g}_1, \ldots, \mathbf{g}_s)^T$, we have $\mathbf{E}_s = \mathbf{VU}$, where $\mathbf{E}_s$ is the $s \times s$ unit matrix. So $\mathbf{U}$ is unimodular. □

Based on the results of Theorem 3.4 and 3.5, we can design an algorithm to compute the GCD of $f_1, \ldots, f_s$ and unimodular matrix $\mathbf{U}$, where the first row $u_{11}, \ldots, u_{1s}$ of $\mathbf{U}$ are the representation coefficients. That is, we only need to construct the module M by inputting polynomials $f_1, \ldots, f_s$ and then compute a minimal Gröbner basis for M with respect to $\succ_{s+1}$.

3.2 Extended GCD systems for parametric univariate polynomials

Now we are ready to generalize the above method to the parametric case by means of the CGS for modules, and get the following result.

Theorem 3.6. *Given $f_1, \ldots, f_s \in k[U][x]$ and a subset $S \subset L^m$. Let $\mathcal{G} = \{(A_i, G_i)\}_{i=1}^l$ be a minimal comprehensive Gröbner system of the module $M = \langle f_1 \mathbf{e}_1 + \mathbf{e}_2, \ldots, f_s \mathbf{e}_1 + \mathbf{e}_{s+1} \rangle \subset k[U][x]^{s+1}$ on S with respect to an order $\succ_{s+1}$ extended from $\succ$ in a position over term fashion with $\mathbf{e}_1 \succ \mathbf{e}_i$ for $2 \leq i \leq s+1$. For each branch (A_i, G_i) where $G_i \neq \{\mathbf{0}\}$ we have the following results.*

1. *Let $G_i' = \{g \in k[U][x] | \ g \neq 0 \ \wedge \exists\, h_1, \ldots, h_s \in k[U][x] \ s.t. \ (g, h_1, \ldots, h_s)^T \in G_i\}$, then $\sigma_\alpha(G_i')$ is a minimal Gröbner basis of the ideal $\langle \sigma_\alpha(f_1), \ldots, \sigma_\alpha(f_s) \rangle$ with respect to $\succ$ for any $\alpha \in A_i$, and $|G_i'| = 1$.*
2. *Let G_i'' be a set defined by $\{0\} \times G_i'' = G_i \cap (\{0\} \times k[U][x]^s)$, then $\sigma_\alpha(G_i'')$ is a minimal Gröbner basis of the syzygy module $\mathrm{Syz}(\sigma_\alpha(\mathrm{f}_1), \ldots, \sigma_\alpha(\mathrm{f}_s))$ with respect to $\succ_s$ for any $\alpha \in A_i$, and $|G_i''| = s-1$. Thus, $\sigma_\alpha(G_i'')$ is a free basis of the syzygy module $\mathrm{Syz}(\sigma_\alpha(\mathrm{f}_1), \ldots, \sigma_\alpha(\mathrm{f}_s))$.*
3. *Assume $G_i = \{\mathbf{g}_1, \cdots, \mathbf{g}_s\}$ and $\mathbf{g}_1 = (d_i, u_{11}, \cdots, u_{1s})^T$, $\mathbf{g}_j = (0, u_{j1}, \cdots, u_{js})^T$ for $2 \leq j \leq s$. Then $\sigma_\alpha(d_i)$ is a GCD of $\sigma_\alpha(f_1), \ldots, \sigma_\alpha(f_s)$ and $\sigma_\alpha(u_{11}), \ldots, \sigma_\alpha(u_{1s})$ are the representation coefficients for $\sigma_\alpha(d_i)$ as a linear combination of $\sigma_\alpha(f_1), \ldots, \sigma_\alpha(f_s)$. Moreover, assume the matrix $\mathbf{U}_i = (u_{kj})_{s\times s}$, then $\sigma_\alpha(\mathbf{U}_i)\sigma_\alpha(\mathbf{f}) = \sigma_\alpha(\mathbf{d}_i)$ and $\sigma_\alpha(\mathbf{U}_i)$ is unimodular for any $\alpha \in A_i$, where*

$$\mathbf{U}_i = \begin{pmatrix} u_{11} & \ldots & u_{1s} \\ u_{21} & \ldots & u_{2s} \\ \vdots & & \vdots \\ u_{s1} & \ldots & u_{ss} \end{pmatrix}, \quad \mathbf{f} = \begin{pmatrix} f_1 \\ f_2 \\ \vdots \\ f_s \end{pmatrix}, \quad \mathbf{d}_i = \begin{pmatrix} d_i \\ 0 \\ \vdots \\ 0 \end{pmatrix}.$$

Particularly, for the branch (A_i, G_i) where $G_i = \{\mathbf{0}\}$, $\sigma_\alpha(d_i) = 0$ and $\sigma_\alpha(\mathbf{U}_i) = \mathbf{E}_s$ for $\alpha \in A_i$. In this case, the corresponding syzygy module $\mathrm{Syz}(\sigma_\alpha(\mathrm{f}_1), \ldots, \sigma_\alpha(\mathrm{f}_s))$ is $k[x]^s$.

Proof. Since $\mathcal{G}$ is a minimal comprehensive Gröbner system, in each branch (A_i, G_i) where $G_i \neq \{\mathbf{0}\}$, the set $\sigma_\alpha(G_i)$ is a minimal Gröbner basis of $\sigma_\alpha(M)$ for any $\alpha \in A_i$. Besides, there is no element in G_i specializing to $\mathbf{0}$ because the leading coefficients of all elements in G_i are non-zero under specialization. Thus, it is easy to derive the results from Proposition 3.3, Theorem 3.4 and 3.5. □

3.3 Parametric extended GCD algorithm

Based on Theorem 3.6, we are ready to give an algorithm to compute the extended GCD system for parametric univariate polynomials.

Theorem 3.7. *Algorithm 1 works correctly and terminates.*

Algorithm 1: Parametric extended GCD algorithm

Input : $f_1, \ldots, f_s \in k[U][x]$, a constructible set $A \subset L^m$, and a POT order $\succ_{s+1}$ with $\mathbf{e}_1 \succ \mathbf{e}_i, i \geq 2$.

Output: an extended GCD system $\{(A_i, \mathbf{U}_i, d_i)_{i=1}^{l}\}$, where $\mathrm{GCD}(\sigma_\alpha(f_1), \ldots, \sigma_\alpha(f_s)) = \sigma_\alpha(d_i)$ and $\sigma_\alpha(\mathbf{U}_i)$ is unimodular for any $\alpha \in A_i$.

begin
  compute a minimal CGS $\{(A_i, G_i)_{i=1}^{l}\}$ for the module $M = \langle f_1\mathbf{e}_1 + \mathbf{e}_2, \ldots, f_s\mathbf{e}_1 + \mathbf{e}_{s+1}\rangle$ w.r.t. $\succ_{s+1}$;
  for *i from 1 to l* **do**
    $G_i := \{u_0\mathbf{e}_1 + \sum_{j=1}^{s} u_{1j}\mathbf{e}_{j+1}, \sum_{j=1}^{s} u_{2j}\mathbf{e}_{j+1}, \ldots, \sum_{j=1}^{s} u_{sj}\mathbf{e}_{j+1}\}$;
    $\mathbf{U}_i := (u_{kj})_{s\times s},\ 1 \leq k, j \leq s$;
    $d_i := u_0$;
  return $\{(A_i, \mathbf{U}_i, d_i)\}_{i=1}^{l}$;

Proof. The correctness of Algorithm 1 directly follows from Theorem 3.6, and the termination of Algorithm 1 fully depends on that of the algorithm for computing CGSs of the module M which is obviously derived from the termination of KSW algorithm as mentioned in Remark 1. □

Remark 2. *For each $(A_i, \mathbf{U}_i, d_i)$, the components of the first row vector in $\mathbf{U}_i$ are the representation coefficients of d_i.*

We use the following simple example to illustrate the steps in the above proposed algorithm.

Example 3.8. Let $f_1, f_2, f_3 \in \mathbb{C}[U][x]$ be as follows:

$$f_1 = (x-a)^2, \quad f_2 = (x-b)^2, \quad f_3 = x(x-b),$$

where $U = \{a, b\}$ and $\succ$ is a lexicographic order.

Step 1: we compute a minimal CGS $\mathcal{G}$ for the module $M = \langle f_1\mathbf{e}_1 + \mathbf{e}_2, f_2\mathbf{e}_1 + \mathbf{e}_3, f_3\mathbf{e}_1 + \mathbf{e}_4\rangle \subset \mathbb{C}[a, b][x]^4$ with respect to $\succ_4$ where $\mathbf{e}_1 \succ \mathbf{e}_2 \succ \mathbf{e}_3 \succ \mathbf{e}_4$, and the result is shown in Table 1 where

Table 1: a minimal CGS $\mathcal{G}$ for the module M

No.	A_i	G_i
1	$\mathbb{C}^2\backslash\mathbb{V}(b(b-a))$	G_1
2	$\mathbb{V}(b)\backslash\mathbb{V}(a^2)$	G_2
3	$\mathbb{V}(a-b)\backslash\mathbb{V}(b)$	G_3
4	$\mathbb{V}(a, b)$	G_4

$$\begin{aligned}
G_1 =\{ & b(a-b)^2\mathbf{e}_1 + b\mathbf{e}_2 + (-2a+b)\mathbf{e}_3 + (2a-2b)\mathbf{e}_4,\\
& (bx-b^2)\mathbf{e}_2 + a^2\mathbf{e}_3 + (-bx-a^2+2ab)\mathbf{e}_4,\ x\mathbf{e}_3 + (b-x)\mathbf{e}_4\};\\
G_2 =\{ & a^3\mathbf{e}_1 + (a+2x)\mathbf{e}_2 + (3a-2x)\mathbf{e}_4,\ x^2\mathbf{e}_2 - (a^2-2ax+x^2)\mathbf{e}_4,\\
& \mathbf{e}_3 - \mathbf{e}_4\};\\
G_3 =\{ & (-b^2+bx)\mathbf{e}_1 - \mathbf{e}_3 + \mathbf{e}_4,\ \mathbf{e}_2 - \mathbf{e}_3,\ x\mathbf{e}_3 + (b-x)\mathbf{e}_4\};\\
G_4 =\{ & x^2\mathbf{e}_1 + \mathbf{e}_4,\ \mathbf{e}_2 - \mathbf{e}_4,\ \mathbf{e}_3 - \mathbf{e}_4\}.
\end{aligned}$$

Step 2: according to G_i in the minimal CGS for module M, we construct $\mathbf{U}_i$ and d_i, where

$$d_1 = b(a-b)^2, \quad d_2 = a^3, \quad d_3 = -b^2 + bx, \quad d_4 = x^2.$$

$$\mathbf{U}_1 = \begin{pmatrix} b & -2a+b & 2a-2b \\ bx-b^2 & a^2 & -bx-a^2+2ab \\ 0 & x & b-x \end{pmatrix}, \ \mathbf{U}_2 = \begin{pmatrix} a+2x & 0 & 3a-2x \\ x^2 & 0 & -(a-x)^2 \\ 0 & 1 & -1 \end{pmatrix},$$

$$\mathbf{U}_3 = \begin{pmatrix} 0 & -1 & 1 \\ 1 & -1 & 0 \\ 0 & x & b-x \end{pmatrix}, \qquad \mathbf{U}_4 = \begin{pmatrix} 0 & 0 & 1 \\ 1 & 0 & -1 \\ 0 & 1 & -1 \end{pmatrix}.$$

In summary, parametric GCDs are expressed as the linear representations of f_1, f_2, f_3 as follows.

$$\begin{cases} if\ a \neq b\ and\ b \neq 0,\ bf_1 + (-2a+b)f_2 + (2a-2b)f_3 = b(a-b)^2; \\ if\ a \neq b\ and\ b = 0,\ (a+2x)f_1 + 0\cdot f_2 + (3a-2x)f_3 = a^3; \\ if\ a = b\ and\ b \neq 0,\ 0\cdot f_1 - 1\cdot f_2 + 1\cdot f_3 = -b^2 + bx; \\ if\ a = b\ and\ b = 0,\ 0\cdot f_1 + 0\cdot f_2 + 1\cdot f_3 = x^2. \end{cases}$$

4 APPLICATION TO SMITH NORMAL FORM

4.1 Notations and definitions

In this subsection, we give some definitions and notations related to the Smith normal form. A matrix is called non-parametric (parametric) univariate polynomial matrix if its entries belong to $k[x]$ ($k[U][x]$).

Definition 4.1. Let $\mathbf{D}$ be an $s \times t$ matrix over $k[x]$ such that

(1) all (i, j)-entries in $\mathbf{D}$ are zero for $i \neq j$, that is, $\mathbf{D}$ is a diagonal matrix;
(2) each (i, i)–entry d_i in $\mathbf{D}$ is either monic or zero;
(3) $d_i \mid d_{i+1}$ for $1 \leq i < min\{s, t\}$.

Then $\mathbf{D} = diag(d_1, \ldots, d_{min\{s,t\}})$ is said to be in Smith normal form, where "diag" stands for the diagonal matrix.

In addition, we give the following theorem appearing in [28] which ensures the existence of the Smith normal form for any univariate polynomial matrix $\mathbf{B}$ over $k[x]$.

Theorem 4.2. *Let $\mathbf{B}$ be an $s \times t$ matrix over $k[x]$, then there is a sequence of elementary operations over $k[x]$ which changes $\mathbf{B}$ into $S(\mathbf{B})$ that is in Smith normal form, called the Smith normal form of $\mathbf{B}$.*

That is, there exist unimodular matrices $\mathbf{U} \in k[x]^{s\times s}$, $\mathbf{V} \in k[x]^{t\times t}$ such that $\mathbf{UBV} = S(\mathbf{B})$.

4.2 The Smith normal form of parametric univariate polynomial matrix

For the non-parametric case, as stated in Theorem 4.2 any univariate polynomial matrix can be reduced to its Smith normal form under the elementary operations. As for the the parametric case, corresponding to each algebraically constructible subset $A_i \subset L^m$, the parametric univariate polynomials matrix under the specialization σ_α can be reduced to its Smith normal form by elementary operations, i.e. there exist parametric unimodular matrices $\mathbf{U} \in k[U][x]^{s\times s}$, $\mathbf{V} \in k[U][x]^{t\times t}$ such that $\sigma_\alpha(\mathbf{U})\sigma_\alpha(\mathbf{B})\sigma_\alpha(\mathbf{V}) = S(\sigma_\alpha(\mathbf{B}))$ for $\alpha \in A_i$. Now we discuss how to reduce a univariate polynomials matrix to its Smith normal form.

In the above section, we have proposed an extended GCD algorithm which not only can output the GCD, but also gives a unimodular matrix $\mathbf{U}$. In particular, $\mathbf{U}(\mathbf{f}_1, \ldots, \mathbf{f}_s)^T = (d, 0, \ldots, 0)$, where $\mathbf{f}_1, \ldots, \mathbf{f}_s$ are given polynomials and d is the GCD of these polynomials. Then, we can apply the extended GCD algorithm to the calculation of the Smith normal form, and the actual practice is as follows.

Given $\mathbf{B} \in k[x]^{s\times t}$ (without loss of generality, assume $s \le t$), we first call the extended GCD algorithm on the first column of $\mathbf{B}$ and obtain the unimodular matrix $\mathbf{U} \in k[x]^{s\times s}$. Then $\mathbf{U}$ acts on $\mathbf{B}$, and the first column of $\mathbf{UB}$ are zeros except for the first element. Next, do the same operation for the first row of the $\mathbf{UB}$, we still get a unimodular matrix $\mathbf{V} \in k[x]^{t\times t}$ such that the first row in $\mathbf{UBV}$ are zeros except for the first element, but note that the first column are not necessarily zeros. So we repeatedly perform the above operation in order to get a matrix in which the first column and row are zeros except for the (1,1)-component. This is the first step. If all other elements in the new obtained matrix can be divisible by the (1,1)-element, then we only need to conduct the same step as the first step on the lower right submatrix of this matrix. Otherwise, we need an extra step to ensure the divisibility relation. Finally we will get the Smith normal form of $\mathbf{B}$. Most importantly, these can be naturally extended to the parametric case.

Here we will give the algorithm for the parametric case. Before discussing the algorithm, we would like to introduce some useful propositions which are related to the termination of the algorithm.

As known to all, currently the algorithms are all computing the minimal CGS, and the minimal CGS for modules over parametric multivariate polynomial rings can't always be reduced to the reduced CGS. Here we show that for univariate polynomial rings it can be done.

Proposition 4.3. *A minimal CGS $\mathcal{G} = \{(A_1, G_1), \ldots, (A_l, G_l)\}$ for module $M \subset k[U][x]^s$ with respect to the POT order $\succ_s$ can be reduced to a reduced CGS.*

Proof. By Definition 2.5, we only need to prove that for each branch (A_k, G_k) of $\mathcal{G}$ where $k = 1, \ldots, l$, the parametric minimal Gröbner basis G_k for M can be reduced to the parametric reduced Gröbner basis on A_k. For any $\mathbf{g_i}, \mathbf{g_j} \in G_k$, suppose that $\mathrm{LM}(\mathbf{g_i}) = g_1\epsilon_i$ and $\mathrm{LM}(\mathbf{g_j}) = g\epsilon_j$. Without loss of generality, one can assume $\epsilon_i \succ \epsilon_j$ and the j-th component of $\mathbf{g_i}$ is f, then the $i-$th component of $\mathbf{g_j}$ must be zero. If f is reduced w.r.t. g (i.e. no monomial of f is divisible by $\mathrm{LM}(g)$), there is nothing to do. Otherwise do pseudo division to f by g, then one get $hf = qg + r$ where h is the power of the leading coefficient of g w.r.t. the main variable x and $\sigma_\alpha(h) \neq 0$ for any $\alpha \in A_k$. Thus, $h\mathbf{g_i} - q\mathbf{g_j} = \mathbf{g_i'}$ where $\mathbf{g_i'}$ is reduced w.r.t $\mathbf{g_j}$. Replacing $\mathbf{g_i}$ with $\mathbf{g_i'}$ and repeating the above process. Moreover, according to the definition of minimal CGS, $\sigma_\alpha(\mathrm{LC}_x(\mathbf{g})) \neq 0$ for any $\mathbf{g} \in G_k$ and $\alpha \in A_k$, then we can divide the coefficient such that $\sigma_\alpha(\mathrm{LC}_x(\mathbf{g})) = 1$, Thus, $\sigma_\alpha(G_k)$ is reduced. This proves the proposition. □

By the above proposition, we can get a new version of Algorithm 1 by computing a reduced CGS instead of a minimal CGS for M, denoted by Algorithm 1*.

Proposition 4.4. *Given $f_1, \cdots, f_s \in k[U][x]$, a constructible set $A \subset L^m$ and a POT order $\succ_{s+1}$ with $\mathbf{e}_1 \succ \mathbf{e}_{s+1} \succ \cdots \succ \mathbf{e}_2$. By Algorithm 1* we will get a reduced CGS $\{(A_i, G_i)\}_{i=1}^l$ and a GCD system $\{(A_i, \mathbf{U}_i, d_i)\}_{i=1}^l$, where $G_i = \{\mathbf{g}_1, \ldots, \mathbf{g}_s\}$, $\mathbf{g}_1 = (d_i, u_{11}, \cdots, u_{1s})^T$, $\mathbf{g}_j = (0, u_{j1}, \cdots, u_{js})^T$ for $2 \le j \le s$. Then for any $\alpha \in A_i$, under the specialization σ_α, $\mathbf{u}_i = (u_{11}, \ldots, u_{1s})^T$ is the minimal element in $M_i = \{(h_1, \ldots, h_s)^T | h_1 f_1 + \cdots + h_s f_s = d_i\}$ under $\succ_s$ being the restriction of $\succ_{s+1}$ on $k[x]^s$.*

Proof. Assume that under σ_α, $\mathbf{u}_i$ is not minimal, then there exists $\mathbf{u}_i' \in M_i$ and $\sigma_\alpha(\mathbf{u}_i) \succ_s \sigma_\alpha(\mathbf{u}_i')$. By the definition of M_i, we have $\sigma_\alpha(\mathbf{u}_i - \mathbf{u}_i') \in \mathrm{Syz}(\sigma_\alpha(f_1), \ldots, \sigma_\alpha(f_s))$. Thus $\mathrm{LM}(\sigma_\alpha(\mathbf{u}_i)) = \mathrm{LM}(\sigma_\alpha(\mathbf{u}_i - \mathbf{u}_i')) \in \mathrm{LM}(\mathrm{Syz}(\sigma_\alpha(f_1), \ldots, \sigma_\alpha(f_s)))$. By Theorem 3.6, it implies that some term of $\sigma_\alpha(\mathbf{g}_1)$ is divisible by one of $\mathrm{LM}(\sigma_\alpha(\mathbf{g}_2))$, $\ldots$, $\mathrm{LM}(\sigma_\alpha(\mathbf{g}_s))$, which contradicts that $\sigma_\alpha(G_i)$ is reduced. □

Now we give the algorithm for computing the Smith normal form of univariate polynomial matrices with parameters, and prove the termination of the algorithm.

Algorithm 2: Parametric Smith normal form algorithm

Input : $\mathbf{B} \in k[U][x]^{s\times t}$, a constructible set $A \subset L^m$, and a POT order $\succ_{s+1}$ with $\mathbf{e}_1 \succ \mathbf{e}_{s+1} \succ \cdots \succ \mathbf{e}_2$.

Output: $\{[A_i, \mathbf{B}_i, \mathbf{U}_i, \mathbf{V}_i]\}_{i=1}^l$, where $\sigma_\alpha(\mathbf{U}_i)\sigma_\alpha(\mathbf{B})\sigma_\alpha(\mathbf{V}_i) = \sigma_\alpha(\mathbf{B}_i)$ and $\sigma_\alpha(\mathbf{B}_i)$ is in Smith normal form for any $\alpha \in A_i$.

```
begin
    G := {}; G_1 := {[A, B, E_s, E_t, B]}; d := 0;
    while G_1 is not empty do
        [A_0, B_0, U_0, V_0, S_0] := G_1[1]; G_1 := G_1 \ {G_1[1]};
        H_1 := Reduce2Zero(A_0, S_0);
        for [A_i, B_i, U_i, V_i] in H_1 do
            H_2 := Divisible(A_i, B_i);
            for [A_j, B_j, U_j, V_j] in H_2 do
                U_1 := diag(E_d, U_j U_i);
                V_1 := diag(E_d, V_i V_j);
                B_1 := U_1 B_0 V_1; U := U_1 U_0; V := V_0 V_1;
                if d = s − 1 then
                    G := G ∪ {[A_j, B_1, U, V]};
                else
                    d := d + 1;
                    G_1 := G_1 ∪ {[A_j, B_1, U, V, SubMatrix(B_1, d)]};
    return G;
```

In Algorithm 2, Reduce2Zero$(A_0, \mathbf{S}_0)$ stands for repeatedly calling Algorithm 1* on the first column and row of the matrix (matrices) for each algebraically constructible subset and the details is as follows. Divisible$(A_i, \mathbf{B}_i)$ is used to check whether all other elements in $\mathbf{B}_i$ can be divisible by (1,1)-element on A_i, if not, we need the extra step: adding the corresponding column in which the element which isn't divisible by (1,1)-element of $\mathbf{B}_i$ is to the first column of $\mathbf{B}_i$ and getting $\mathbf{B}_i'$, then performing Reduce2Zero$(A_i, \mathbf{B}_i')$. SubMatrix$(\mathbf{B}_1, \mathrm{d})$ denotes the lower right submatrix of $\mathbf{B}_1$ which consists of the last $s - d$ rows and $t - d$ columns.

In Algorithm 3, CEGCD$(\mathrm{A}, \mathbf{B})$ and REGCD$(A, \mathbf{B})$ stand for calling Algorithm 1* on the first column and row of matrix $\mathbf{B}$ on the constructible set A, respectively. IsZero$(A_{i_j}, \mathbf{B}_{i_j})$ is a subroutine to determine if the first column and row of $\mathbf{B}_{i_j}$ are zeros except for the (1,1)-element on algebraically constructible subset A_{i_j}.

Proposition 4.5. *Algorithm 2 terminates within finite steps.*

Proof. According to the design of the algorithm and above explain, we only need to prove that Algorithm 3 (Reduce2Zero$(A, \mathbf{B})$) terminates within finite steps. Since the original (1,1)-element of univariate polynomial matrix $\mathbf{B}$ has a definite degree and since

Algorithm 3: Reduce2Zero

Input : $\mathbf{B} \in k[U][x]^{s\times t}$, a constructible set $A \subset L^m$, and a POT order $\succ_{s+1}$ with $\mathbf{e}_1 \succ \mathbf{e}_{s+1} \succ \cdots \succ \mathbf{e}_2$.

Output: $\{[A_i, \mathbf{B}_i, \mathbf{U}_i, \mathbf{V}_i]\}_{i=1}^{l}$, where $\sigma_\alpha(\mathbf{U}_i)\sigma_\alpha(\mathbf{B})\sigma_\alpha(\mathbf{V}_i) = \sigma_\alpha(\mathbf{B}_i)$ for any $\alpha \in A_i$ and the first column and row of $\mathbf{B}_i$ are zeros except for the (1,1)-element on A_i.

```
begin
    G := {}; G_1 := {[A, B, E_s, E_t]};
    while G_1 is not empty do
        [A_0, B_0, U_0, V_0] := G_1[1]; G_1 := G_1 \ {G_1[1]};
        H_1 := CEGCD(A_0, B_0);
        for [A_i, U_i, d_i] in H_1 do
            B_i := U_i B_0; U_i := U_i U_0;
            H_2 := REGCD(A_i, B_i);
            for [A_ij, V_ij, d_ij] in H_2 do
                B_ij := B_i V_ij^T; V_ij := V_0 V_ij^T;
                if IsZero(A_ij, B_ij) then
                    G := G ∪ {[A_ij, B_ij, U_i, V_ij]};
                else
                    G_1 := G_1 ∪ {[A_ij, B_ij, U_i, V_ij]};
    return G;
```

the process of reducing the degree for the (1,1)-element cannot be continued indefinitely, after a finite times of loops the degree of (1,1)-element w.r.t. main variable x is stable and assume at the moment we get $\mathbf{B}_i$ of which the first column of are zeros except for the (1,1)-element on A_i. Then $H_2 := \mathrm{REGCD}(\mathrm{A_i}, \mathrm{B_i})$, and we get a unimodular matrix $\mathbf{V}_{i_j}^T$ which can reduce the first row of $\mathbf{B}_i$ to be zeros on new algebraically constructible subset A_{i_j}. Since under the specialization, the degree of (b_{11}) is stable, b_{11} is the GCD of the first row elements of $\mathbf{B}_i$. We claim that $\mathbf{V}_{i_j}^T$ has the following form:

$$\mathbf{V}_{i_j}^T = \begin{bmatrix} v_{11} & v_{12} & \dots & v_{1t} \\ 0 & v_{11} & \dots & v_{2t} \\ \vdots & \vdots & \vdots & \vdots \\ 0 & v_{t2} & \dots & v_{tt} \end{bmatrix}.$$

Otherwise, assume that for some $\alpha \in A_{i_j}$, there exists at least one $\sigma_\alpha(v_{l1}) \neq 0, 2 \le l \le t$. Obviously, $\sigma_\alpha(\mathbf{v_1}) = (\sigma_\alpha(v_{11}), \dots, \sigma_\alpha(v_{t1}))^T \succ_t (\sigma_\alpha(v_{11}), 0, \dots, 0)^T$ under the POT order $\succ_t$ being the restriction of $\succ_{t+1}$ with $\mathbf{e}_1 \succ \mathbf{e}_{t+1} \succ \cdots \succ \mathbf{e}_2$ on $k[x]^t$, which contradicts that $\sigma_\alpha(\mathbf{v_1})$ should be minimal by Proposition 4.4.

Thus, $\mathbf{B}_{i_j} = \mathbf{B}_i \mathbf{V}_{i_j}^T$ satisfies that the first column and row are zeros except for the (1,1)-element on A_{i_j}. Consequently, Algorithm 3 terminates. □

We use a simple example to illustrate Algorithm 2.

Example 4.6. Given a matrix $B \in \mathbb{C}[a][x]^{3\times 3}$ and a constructible set $A = \mathbb{C}$ as follows:

$$\mathbf{B} = \begin{bmatrix} a-x & 2x & 0 \\ 0 & 0 & x \\ x^2+1 & x^3+a+x & -x^2 \end{bmatrix}.$$

Step 1: perform the routine Reduce2Zero($A, \mathbf{B}$), that is, repeatedly call Algorithm 1* on the first column and row of the matrix, then we get the matrices in which the first column and row are zeros except for the (1,1)-component.

Table 2: Output of Reduce2Zero($A, \mathbf{B}$)

No.	A_i	$\mathbf{B}_i$	$\mathbf{U}_i$	$\mathbf{V}_i$
1	$\mathbb{C}\backslash\mathbb{V}(a^2+1)$	$\mathbf{B}_1$	$\mathbf{U}_1$	$\mathbf{V}_1$
2	$\mathbb{V}(a^2+1)$	$\mathbf{B}_2$	$\mathbf{U}_2$	$\mathbf{V}_2$

where ($\mathbf{U}_i\mathbf{B}\mathbf{V}_i = \mathbf{B}_i$, $i = 1, 2$.)

$$\mathbf{B}_1 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & x(a^2+1) & 0 \\ 0 & (a^2+1)(a-x)x^2 & b_{133} \end{bmatrix}, \quad \mathbf{B}_2 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & x & 0 \\ 0 & -2x^2 & b_{233} \end{bmatrix},$$

$$\mathbf{U}_1 = \begin{bmatrix} a+x & 0 & 1 \\ u_{121} & 1 & u_{123} \\ u_{131} & 0 & u_{133} \end{bmatrix}, \quad \mathbf{U}_2 = \begin{bmatrix} -1 & 0 & 0 \\ 0 & 1 & 0 \\ ax^3+2ax^2+ax+2a-1 & 0 & 2 \end{bmatrix},$$

$$\mathbf{V}_1 = \begin{bmatrix} -4x^2+1 & x^2 & v_{113} \\ 2ax-a+x & 0 & a^2+1 \\ v_{131} & a^2+1 & 0 \end{bmatrix}, \quad \mathbf{V}_2 = \begin{bmatrix} a & 0 & 2x \\ a/2 & 0 & -a+x \\ 0 & 1 & 0 \end{bmatrix},$$

$$b_{133} = -(a^2+1)(ax^3 - x^4 - 2x^3 + a^2 - x^2 - 2x),$$
$$b_{233} = -2(a-x)(x^3 + 2ax + 2x^2 + a + x),$$
$$u_{121} = -x(a+x)(2ax^2 + 3ax + x^2 + 2a + 2x - 3),$$
$$u_{123} = -2ax^3 - 3ax^2 - x^3 - 2ax - 2x^2 + 3x,$$
$$u_{131} = (a^2+1)(-4ax^3 - 4x^4 + 2a^2x - 2x^3 - a^2 + x^2 + 1),$$
$$u_{133} = (a^2+1)(-4x^3 + 2ax - 2x^2 - a + x),$$
$$v_{113} = -x^3 - 2ax - 2x^2 - a - x,$$
$$v_{131} = 2ax^2 + 3ax + x^2 + 2a + 2x - 3.$$

Step 2: perform the subroutine Divisible($A_i, \mathbf{B}_i$) to check if all elements in $\mathbf{B}_i$ are divisible by the (1,1)-element.

Obviously, $\mathbf{B}_1$ and $\mathbf{B}_2$ satisfy the divisibility relation between the (1,1)-element and other elements.

Step 3: repeat the Step 1 and Step 2 on the lower right submatrices of $\mathbf{B}_1$ and $\mathbf{B}_2$. We obtain the following, where $A_1' \cup A_2' = A_1$, $\mathbf{B}_1'$ and $\mathbf{B}_2'$ come from SubMatrix($\mathbf{B}_1$, 1).

Table 3: Output of SubMatrix($\mathbf{B}_1$, 1) **and** SubMatrix($\mathbf{B}_2$, 1)

No.	A_i'	$\mathbf{B}_i'$	$\mathbf{U}_i'$	$\mathbf{V}_i'$
1	$\mathbb{C}\backslash\mathbb{V}(a(a^2+1))$	$\mathbf{B}_1'$	$\mathbf{U}_1'$	$\mathbf{V}_1'$
2	$\mathbb{V}(a)\backslash\mathbb{V}(a^2+1)$	$\mathbf{B}_2'$	$\mathbf{U}_2'$	$\mathbf{V}_2'$
3	$\mathbb{V}(a^2+1)$	$\mathbf{B}_3'$	$\mathbf{U}_3'$	$\mathbf{V}_3'$

$$\mathbf{B}_1' = \begin{bmatrix} 1 & 0 \\ 0 & b'_{122} \end{bmatrix}, \quad \mathbf{B}_2' = \begin{bmatrix} x & 0 \\ 0 & b'_{222} \end{bmatrix}, \quad \mathbf{B}_3' = \begin{bmatrix} x & 0 \\ 0 & b'_{322} \end{bmatrix},$$

$$\mathbf{U}_1' = \begin{bmatrix} u'_{111} & -1/(a^4+a^2) \\ u'_{121} & x/(a^4+a^2) \end{bmatrix}, \quad \mathbf{U}_2' = \begin{bmatrix} 1 & 0 \\ u'_{221} & 1/(a^4+a^2) \end{bmatrix}, \quad \mathbf{U}_3' = \begin{bmatrix} 1 & 0 \\ x & 1/2 \end{bmatrix},$$

$$\mathbf{V}_1' = \begin{bmatrix} 1 & v'_{112} \\ 1 & v'_{122} \end{bmatrix}, \quad \mathbf{V}_2' = \begin{bmatrix} 1 & 0 \\ 1 & 1 \end{bmatrix}, \quad \mathbf{V}_3' = \begin{bmatrix} 1 & 0 \\ x & 1/2 \end{bmatrix},$$

$$b'_{122} = -x(ax^3 - x^4 - 2x^3 + a^2 - x^2 - 2x),$$
$$b'_{222} = -x(ax^2 - x^3 - 2x^2 - x - 2),$$
$$b'_{322} = (x-a)(x^3 + 2ax + 2x^2 + a + x),$$
$$u'_{111} = (-ax^2 + x^3 + ax + x^2 + x + 2)/(a^4 + a^2),$$
$$u'_{121} = (ax^3 - x^4 - ax^2 - x^3 + a^2 - x^2 - 2x)/(a^4 + a^2),$$
$$u'_{221} = (ax^2 - x^3 - ax - x^2 - x - 2)/(a^4 + a^2),$$
$$v'_{112} = -ax^3 + x^4 + 2x^3 - a^2 + x^2 + 2x,$$
$$v'_{122} = -ax^3 + x^4 + 2x^3 + x^2 + 2x.$$

Step 4: recover the Smith normal forms. Where

Table 4: recover Smith normal forms

No.	A''_i	$\mathbf{B}''_i$	$\mathbf{U}''_i$	$\mathbf{V}''_i$
1	$\mathbb{C}\backslash\mathbb{V}(a(a^2+1))$	$\mathbf{B}''_1$	$\mathbf{U}''_1$	$\mathbf{V}''_1$
2	$\mathbb{V}(a)\backslash\mathbb{V}(a^2+1)$	$\mathbf{B}''_2$	$\mathbf{U}''_2$	$\mathbf{V}''_2$
3	$\mathbb{V}(a^2+1)$	$\mathbf{B}''_3$	$\mathbf{U}''_3$	$\mathbf{V}''_3$

$$\mathbf{U}''_1 = \begin{bmatrix} 1 & 0 \\ 0 & \mathbf{U}'_1 \end{bmatrix}\mathbf{U}_1, \quad \mathbf{U}''_2 = \begin{bmatrix} 1 & 0 \\ 0 & \mathbf{U}'_2 \end{bmatrix}\mathbf{U}_1, \quad \mathbf{U}''_3 = \begin{bmatrix} 1 & 0 \\ 0 & \mathbf{U}'_3 \end{bmatrix}\mathbf{U}_2,$$

$$\mathbf{V}''_1 = \mathbf{V}_1\begin{bmatrix} 1 & 0 \\ 0 & \mathbf{V}'_1 \end{bmatrix}, \quad \mathbf{V}''_2 = \mathbf{V}_1\begin{bmatrix} 1 & 0 \\ 0 & \mathbf{V}'_2 \end{bmatrix}, \quad \mathbf{V}''_3 = \mathbf{V}_2\begin{bmatrix} 1 & 0 \\ 0 & \mathbf{V}'_3 \end{bmatrix},$$

$$\mathbf{B}''_1 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & b''_{133} \end{bmatrix}, \quad \mathbf{B}''_2 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & x & 0 \\ 0 & 0 & b''_{233} \end{bmatrix}, \quad \mathbf{B}''_3 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & b''_{333} \end{bmatrix},$$

$$b''_{133} = x^5 + (-a+2)x^4 + x^3 + 2x^2 - a^2x,$$
$$b''_{233} = x^4 + 2x^3 + x^2 + 2x,$$
$$b''_{333} = -2a^2x^2 - a^2x + x^3 + (-a+2)x^4 + x^5.$$

5 CONCLUDING REMARKS

An algorithm for computing extended GCD systems of parametric univariate polynomials has been proposed. We can see that this algorithm simultaneously give the GCD and the representation coefficients by computing the CGS of a constructed module, which adds the unit vectors to record the representation coefficients (as mentioned in [5]). Meanwhile, this CGS for M also gives a set of free bases for the parametric syzygy module of input polynomials. It is worth noting that we get a stronger result: the unimodular matrix $\mathbf{U}$. Therefore, we can apply the proposed extended GCD algorithm to the computation of the Smith normal form and present the first algorithm for computing the Smith normal form of univariate polynomial matrices with parameters. In addition, the proposed algorithms have been implemented on the computer algebra system *Maple*, and the codes and examples are available on the web: http://www.mmrc.iss.ac.cn/~dwang/software.html.

ACKNOWLEDGMENTS

This research was supported in part by CAS Project QYZDJ-SSW-SYS022.

REFERENCES

[1] S.A. Abramov and K.Y. Kvashenko. 1993. On the Greatest Common Divisor of Polynomials which Depend on a Parameter. In *Proceedings of the 1993 ACM International Symposium on Symbolic and Algebraic Computation*. 152–156.
[2] A. Ayad. 2010. Complexity of algorithms for computing greatest common divisors of parametric univariate polynomials. *International Journal of Algebra* 4 (2010), 173–188.
[3] T. Bächler, V. Gerdt, M. Lange-Hegermann, and D. Robertz. 2012. Algorithmic Thomas decomposition of algebraic and differential systems. *Journal of Symbolic Computation* 47, 10 (2012), 1233–1266.
[4] S. Bamett. 1971. Matrices in control theory. *Van Norstrand Reinhold* (1971).
[5] B. Beckermann, G. Labahn, and G. Villard. 1999. Shifted normal forms of polynomial matrices. In *Proceedings of ISSAC' 1999*. 189–196.
[6] G. Bradley. 1971. Algorithms for Hermite and Smith normal matrices and linear diophantine equations. *Math. Comp.* 25, 116 (1971), 897–907.
[7] R. P. Brent and H. T. Kung. 1984. Systolic VLSI Arrays for Polynomial GCD Computation. *IEEE Trans. Comput.* 100, 8 (1984), 731–736.
[8] W.S. Brown. 1971. On Euclid's Algorithm And The Computation Of Polynomial Greatest Common Divisors. *J. ACM* 18, 4 (1971), 478–504.
[9] C. Chen and M. Maza. 2012. Algorithms for computing triangular decomposition of polynomial systems. *Journal of Symbolic Computation* 47, 6 (2012), 610–642.
[10] S.C. Chou. 1988. *Mechanical geometry theorem proving*. Vol. 41. Springer Science and Business Media.
[11] R.M. Corless, M.M. Maza, and S.E. Thornton. 2017. Jordan Canonical Form with Parameters from Frobenius Form with Parameters. In *International Conference on Mathematical Aspects of Computer and Information Sciences*. 179–194.
[12] D. Cox, J. Little, and D. O'shea. 2006. *Using algebraic geometry*. Vol. 185. Springer Science & Business Media.
[13] D. Cox, T. Sederberg, and F.L. Chen. 1998. The moving line ideal basis of planar rational curves. *Computer Aided Geometric Design* 15, 8 (1998), 803–827.
[14] F. R. Gantmakher. 1959. *The theory of matrices*. American Mathematical Soc.
[15] K. Geddes, S. Czapor, and G. Labahn. 1992. *Algorithms for computer algebra*. Springer Science and Business Media.
[16] P. Gianni and B. Trager. 1985. Gcd's and factoring multivariate polynomials using Grobner bases. In *European Conference on Computer Algebra*. Springer, 409–410.
[17] H. Kai and M.-T. Noda. 2000. Hybrid rational approximation and its applications. *Reliable Computing* 6 (2000), 429–438.
[18] M. Kalkbrener. 1997. On the Stability of Gröbner Bases Under Specializations. *Journal of Symbolic Computation* 24, 1 (1997), 51–58.
[19] D. Kapur, D. Lu, M. Monagan, Y. Sun, and D.K. Wang. 2018. An Efficient Algorithm for Computing Parametric Multivariate Polynomial GCD. In *Proceedings of the 2018 International Symposium on Symbolic and Algebraic Computation*. 239–246.
[20] D. Kapur, Y. Sun, and D.K. Wang. 2010. A new algorithm for computing comprehensive Gröbner systems. In *Proceedings of ISSAC' 2010*. 29–36.
[21] D. Kapur, Y. Sun, and D.K. Wang. 2013. An efficient algorithm for computing a comprehensive Gröbner system of a parametric polynomial system. *Journal of Symbolic Computation* 49 (2013), 27–44.
[22] A. Montes. 2002. A new algorithm for discussing Gröbner bases with parameters. *Journal of Symbolic Computation* 33, 2 (2002), 183–208.
[23] J. Moses and D. Yun. 1973. The ez gcd algorithm. In *Proceedings of the ACM annual conference*. ACM, 159–166.
[24] K. Nabeshima. 2007. PGB: a package for computing parametric Gröbner and related objects. *ACM Communications in Computer Algebra* 41, 3 (2007), 104–105.
[25] K. Nabeshima. 2007. A speed-up of the algorithm for computing comprehensive Gröbner systems. In *Proceedings of ISSAC' 2007*. 299–306.
[26] K. Nabeshima. 2010. On the computation of parametric gröbner bases for modules and syzygies. *Japan Journal of Industrial and Applied Mathematics* 27, 2 (2010), 217–238.
[27] K. Nagasaka. 2017. Parametric Greatest Common Divisors using Comprehensive Gröbner Systems. In *Proceedings of ISSAC' 2017*. 341–348.
[28] C. Norman. 2012. Finitely Generated Abelian Groups and Similarity of Matrices over a Field. *Springer Undergraduate Mathematics* (2012).
[29] I.S. Pace and S. Barnett. 1974. Efficient algorithms for linear system calculations. I: Smith form and common divisor of polynomial matrices. *Internat.j.systems Sci* (1974), 403–411.
[30] H.H. Rosenbrock. 1970. State-space and multivariable theory. (1970).
[31] T. Sasaki and M. Suzuki. 1992. Three new algorithms for multivariate polynomial GCD. *Journal of Symbolic Computation* 13, 4 (1992), 395–411.
[32] J. Sendra and J. Llovet. 1992. An extended polynomial GCD algorithm using Hankel matrices. *Journal of symbolic computation* 13, 1 (1992), 25–39.
[33] A. Storjohann. 1997. A solution to the extended GCD problem with applications. In *Proceedings of the 1997 international symposium on Symbolic and algebraic computation*. 109–116.
[34] A. Suzuki and Y. Sato. 2002. An alternative approach to comprehensive Gröbner bases. *Journal of Symbolic Computation* 36, 3 (2002), 649–667.
[35] A. Suzuki and Y. Sato. 2006. A simple algorithm to compute comprehensive Gröbner bases using Gröbner bases. In *Proceedings of the 2006 ACM International Symposium on Symbolic and Algebraic Computation*. 326–331.
[36] V. Weispfenning. 1992. Comprehensive Gröbner bases. *Journal of Symbolic Computation* 14, 1 (1992), 1–29.
[37] R. Zippel. 1979. Probabilistic algorithms for sparse polynomials. In *Proceedings of the EUROSAM'79*. Springer-Verlag, 216–226.
[38] R. Zippel. 1993. *Effective Polynomial Computation*. Vol. 241. Springer Science and Business Media.

A Second Order Cone Characterization for Sums of Nonnegative Circuits

Jie Wang and Victor Magron
jwang,vmagron@laas.fr
Laboratoire d'Analyse et d'Architecture des Systèmes (LAAS)
Toulouse, France

ABSTRACT

The second-order cone (SOC) is a class of simple convex cones and optimizing over them can be done more efficiently than with semidefinite programming. It is interesting both in theory and in practice to investigate which convex cones admit a representation using SOCs, given that they have a strong expressive ability. In this paper, we prove constructively that the cone of sums of nonnegative circuits (SONC) admits an SOC representation. Based on this, we give a new algorithm to compute SONC decompositions for certain classes of nonnegative polynomials via SOC programming. Numerical experiments demonstrate the efficiency of our algorithm for polynomials with a fairly large size (both size of degree and number of variables).

CCS CONCEPTS

• **Mathematics of computing** → **Semidefinite programming**; • **Computing methodologies** → **Algebraic algorithms**; **Optimization algorithms**.

KEYWORDS

sum of nonnegative circuit polynomials, second-order cone representation, second-order cone programming, polynomial optimization, sum of binomial squares

ACM Reference Format:
Jie Wang and Victor Magron. 2020. A Second Order Cone Characterization for Sums of Nonnegative Circuits. In *International Symposium on Symbolic and Algebraic Computation (ISSAC '20), July 20–23, 2020, Kalamata, Greece.* ACM, New York, NY, USA, 8 pages. https://doi.org/10.1145/3373207.3404033

1 INTRODUCTION

A *circuit polynomial* is of the form $\sum_{\boldsymbol{\alpha}\in\mathscr{A}} c_{\boldsymbol{\alpha}}\mathbf{x}^{\boldsymbol{\alpha}} - d\mathbf{x}^{\boldsymbol{\beta}} \in \mathbb{R}[\mathbf{x}] = \mathbb{R}[x_1,\ldots,x_n]$, where $c_{\boldsymbol{\alpha}} > 0$ for all $\boldsymbol{\alpha} \in \mathscr{A}$, $\mathscr{A} \subseteq (2\mathbb{N})^n$ comprises the vertices of a simplex and $\boldsymbol{\beta}$ lies in the interior of this simplex. The set of *sums of nonnegative circuit polynomials (SONC)* was introduced by Iliman and Wolff in [10] as a new certificate of nonnegativity for sparse polynomials, which is independent of the well-known set of sums of squares (SOS). Another recently introduced alternative certificates [6] are sums of arithmetic-geometric-exponentials (SAGE), which can be obtained via relative entropy programming. The connection between SONC and SAGE polynomials have been recently studied in [13, 20, 27]. It happens that SONC polynomials and SAGE polynomials are actually equivalent [20], and that both have a cancellation-free representation in terms of generators [20, 27].

One of the significant differences between SONC and SOS is that SONC decompositions preserve sparsity of polynomials while SOS decompositions do not in general [27]. The set of SONC polynomials with a given support forms a convex cone, called a *SONC cone*. Optimization problems over SONC cones can be formulated as geometric programs or more generally relative entropy programs (see [11] for the unconstrained case and [7] for the constrained case). Numerical experiments for unconstrained POPs (polynomial optimization problems) in [25] have demonstrated the advantage of the SONC-based methods compared to the SOS-based methods, especially in the high-degree but fairly sparse case.

In the SOS case, there have been several attempts to exploit sparsity occurring in (un-)constrained POPs. The sparse variant [26] of the moment-SOS hierarchy exploits the correlative sparsity pattern among the input variables to reduce the support of the resulting SOS decompositions. Such sparse representation results have been successfully applied in many fields, such as optimal power-flow [12], roundoff error bounds [15] and recently extended to the noncommutative case [14]. Another way to exploit sparsity is to consider patterns based on terms (rather than variables), yielding an alternative sparse variant of Lasserre's hierarchy [28].

One of the similar features shared by SOS/SONC-based frameworks is their intrinsic connections with conic programming: SOS decompositions are computed via semidefinite programming and SONC decompositions via geometric programming. In both cases, the resulting optimization problems are solved with interior-point algorithms, thus output approximate nonnegativity certificates. However, one can still obtain an exact certificate from such output via hybrid numerical-symbolic algorithms when the input polynomial lies in the interior of the SOS/SONC cone. One way is to rely on rounding-projection algorithms adapted to the SOS cone [22] and the SONC cone [19], or alternatively on perturbation-compensation schemes [16, 18] available within the `RealCertify` [17] library.

In this paper, we study the second-order cone representation of SONC cones. An n-dimensional *(rotated) second-order cone (SOC)* is defined as $\mathbf{K}^n := \{\boldsymbol{a} \in \mathbb{R}^n \mid 2a_1a_2 \geq \sum_{i=3}^n a_i^2, a_1 \geq 0, a_2 \geq 0\}$. The SOC is well-studied and has mature solvers. Optimizing via second-order cone programming (SOCP) can be handled more efficiently than with semidefinite programming [1, 2]. On the other hand, despite the simplicity of SOCs, they have a strong ability to express other convex cones (many such examples can be found in [5, Section 3.3]). Therefore, it is interesting in theory and also important from

ISSAC '20, July 20–23, 2020, Kalamata, Greece

ACM ISBN 978-1-4503-7100-1/20/07…$15.00
https://doi.org/10.1145/3373207.3404033

the view of applications to investigate which convex cones can be expressed by SOCs.

Given sets of lattice points $\mathscr{A} \subseteq (2\mathbb{N})^n$, $\mathscr{B}_1 \subseteq \mathrm{conv}(\mathscr{A}) \cap (2\mathbb{N})^n$ and $\mathscr{B}_2 \subseteq \mathrm{conv}(\mathscr{A}) \cap (\mathbb{N}^n \backslash (2\mathbb{N})^n)$ ($\mathrm{conv}(\mathscr{A})$ is the convex hull of $\mathscr{A}$) with $\mathscr{A} \cap \mathscr{B}_1 = \varnothing$, let $\mathrm{SONC}_{\mathscr{A},\mathscr{B}_1,\mathscr{B}_2}$ be the SONC cone supported on $\mathscr{A}, \mathscr{B}_1, \mathscr{B}_2$ (see Definition 4.2). The first main result of this paper is the following theorem.

Theorem 1.1. *For $\mathscr{A} \subseteq (2\mathbb{N})^n$, $\mathscr{B}_1 \subseteq \mathrm{conv}(\mathscr{A}) \cap (2\mathbb{N})^n$ and $\mathscr{B}_2 \subseteq \mathrm{conv}(\mathscr{A}) \cap (\mathbb{N}^n \backslash (2\mathbb{N})^n)$ with $\mathscr{A} \cap \mathscr{B}_1 = \varnothing$, the convex cone $\mathrm{SONC}_{\mathscr{A},\mathscr{B}_1,\mathscr{B}_2}$ admits an SOC representation.*

The fact that SONC cones admit an SOC characterization was firstly proven by Averkov [4, Theorem 17]. However, Averkov's result is more theoretical. Even though Averkov's proof theoretically allows one to construct an SOC representation for a SONC cone, the construction is complicated and wasn't explicitly given in Averkov's paper. Our proof of Theorem 1.1, which involves writing a SONC polynomial as a sum of binomial squares with rational exponents (Theorem 3.9), is totally different from Averkov's and leads to a more concise (hence more efficient) SOC representation for SONC cones. This enables us to propose a new algorithm, based on SOCP, providing SONC decompositions for a certain class of nonnegative polynomials, which in turn yields lower bounds for unconstrained POPs. We test the algorithm on various randomly generated polynomials up to a fairly large size, involving $n \sim 40$ variables and of degree $d \sim 60$. The numerical results demonstrate the efficiency of our algorithm.

2 PRELIMINARIES

Let $\mathbb{R}[\mathbf{x}] = \mathbb{R}[x_1, \ldots, x_n]$ be the ring of real n-variate polynomials, and let $\mathbb{R}_+$ be the set of positive real numbers. For a finite set $\mathscr{A} \subseteq \mathbb{N}^n$, we denote by $\mathrm{conv}(\mathscr{A})$ the convex hull of $\mathscr{A}$. Given a finite set $\mathscr{A} \subseteq \mathbb{N}^n$, we consider polynomials $f \in \mathbb{R}[\mathbf{x}]$ supported on $\mathscr{A} \subseteq \mathbb{N}^n$, i.e., f is of the form $f(\mathbf{x}) = \sum_{\boldsymbol{\alpha} \in \mathscr{A}} c_{\boldsymbol{\alpha}} \mathbf{x}^{\boldsymbol{\alpha}}$ with $c_{\boldsymbol{\alpha}} \in \mathbb{R}, \mathbf{x}^{\boldsymbol{\alpha}} = x_1^{\alpha_1} \cdots x_n^{\alpha_n}$. The support of f is $\mathrm{supp}(f) := \{\boldsymbol{\alpha} \in \mathscr{A} \mid c_{\boldsymbol{\alpha}} \neq 0\}$ and the Newton polytope of f is defined as $\mathrm{New}(f) := \mathrm{conv}(\mathrm{supp}(f))$. For a polytope P, we use $V(P)$ to denote the vertex set of P and use P° to denote the interior of P. For a set A, we use $\#A$ to denote the cardinality of A. A polynomial $f \in \mathbb{R}[\mathbf{x}]$ which is nonnegative over $\mathbb{R}^n$ is called a *nonnegative polynomial*, or a *positive semi-definite (PSD) polynomial*. The following definition of circuit polynomials was proposed by Iliman and De Wolff in [10].

Definition 2.1. A polynomial $f \in \mathbb{R}[\mathbf{x}]$ is called a *circuit polynomial* if it is of the form $f(\mathbf{x}) = \sum_{\boldsymbol{\alpha} \in \mathscr{A}} c_{\boldsymbol{\alpha}} \mathbf{x}^{\boldsymbol{\alpha}} - d\mathbf{x}^{\boldsymbol{\beta}}$ and satisfies the following conditions: (i) $\mathscr{A} \subseteq (2\mathbb{N})^n$ comprises the vertices of a simplex, (ii) $c_{\boldsymbol{\alpha}} > 0$ for each $\boldsymbol{\alpha} \in \mathscr{A}$, (iii) $\boldsymbol{\beta} \in \mathrm{conv}(\mathscr{A})^\circ \cap \mathbb{N}^n$.

If $f = \sum_{\boldsymbol{\alpha} \in \mathscr{A}} c_{\boldsymbol{\alpha}} \mathbf{x}^{\boldsymbol{\alpha}} - d\mathbf{x}^{\boldsymbol{\beta}}$ is a circuit polynomial, then from the definition we can uniquely write $\boldsymbol{\beta} = \sum_{\boldsymbol{\alpha} \in \mathscr{A}} \lambda_{\boldsymbol{\alpha}} \boldsymbol{\alpha}$ with $\lambda_{\boldsymbol{\alpha}} > 0$ and $\sum_{\boldsymbol{\alpha} \in \mathscr{A}} \lambda_{\boldsymbol{\alpha}} = 1$. We define the corresponding *circuit number* as $\Theta_f := \prod_{\boldsymbol{\alpha} \in \mathscr{A}} (c_{\boldsymbol{\alpha}} / \lambda_{\boldsymbol{\alpha}})^{\lambda_{\boldsymbol{\alpha}}}$. The nonnegativity of the circuit polynomial f is decided by its circuit number alone, that is, f is nonnegative if and only if either $\boldsymbol{\beta} \notin (2\mathbb{N})^n$ and $|d| \leq \Theta_f$, or $\boldsymbol{\beta} \in (2\mathbb{N})^n$ and $d \leq \Theta_f$ ([10, Theorem 3.8]). To provide a concise narrative, we refer to a nonnegative circuit polynomial by a nonnegative circuit and also view a monomial square as a nonnegative circuit. An explicit representation of a polynomial being a *sum of nonnegative circuits*, or *SONC* for short, provides a certificate for its nonnegativity. Such a certificate is called a *SONC decomposition*. For simplicity, we denote the set of SONC polynomials by SONC.

For a polynomial $f \in \mathbb{R}[\mathbf{x}]$, let $\Lambda(f) := \{\boldsymbol{\alpha} \in \mathrm{supp}(f) \mid \boldsymbol{\alpha} \in (2\mathbb{N})^n \text{ and } c_{\boldsymbol{\alpha}} > 0\}$ and $\Gamma(f) := \mathrm{supp}(f) \backslash \Lambda(f)$. Then we can write f as $f = \sum_{\boldsymbol{\alpha} \in \Lambda(f)} c_{\boldsymbol{\alpha}} \mathbf{x}^{\boldsymbol{\alpha}} - \sum_{\boldsymbol{\beta} \in \Gamma(f)} d_{\boldsymbol{\beta}} \mathbf{x}^{\boldsymbol{\beta}}$. For each $\boldsymbol{\beta} \in \Gamma(f)$, let

$$\mathscr{F}(\boldsymbol{\beta}) := \{\Delta \mid \Delta \text{ is a simplex}, \boldsymbol{\beta} \in \Delta^\circ, V(\Delta) \subseteq \Lambda(f)\}. \quad (1)$$

By [27, Theorem 5.5], if $f \in \mathrm{SONC}$, then it has a decomposition

$$f = \sum_{\boldsymbol{\beta} \in \Gamma(f)} \sum_{\Delta \in \mathscr{F}(\boldsymbol{\beta})} f_{\boldsymbol{\beta}\Delta} + \sum_{\boldsymbol{\alpha} \in \tilde{\mathscr{A}}} c_{\boldsymbol{\alpha}} \mathbf{x}^{\boldsymbol{\alpha}}, \quad (2)$$

where $f_{\boldsymbol{\beta}\Delta}$ is a nonnegative circuit supported on $V(\Delta) \cup \{\boldsymbol{\beta}\}$ for each Δ and $\tilde{\mathscr{A}} = \{\boldsymbol{\alpha} \in \Lambda(f) \mid \boldsymbol{\alpha} \notin \cup_{\boldsymbol{\beta} \in \Gamma(f)} \cup_{\Delta \in \mathscr{F}(\boldsymbol{\beta})} V(\Delta)\}$.

3 SONC AND SUMS OF BINOMIAL SQUARES

In this section, we give a characterization of SONC polynomials in terms of sums of binomial squares with rational exponents.

3.1 Rational mediated sets

A lattice point $\boldsymbol{\alpha} \in \mathbb{N}^n$ is *even* if it is in $(2\mathbb{N})^n$. For a subset $M \subseteq \mathbb{N}^n$, define $\overline{A}(M) := \{\frac{1}{2}(\boldsymbol{v} + \boldsymbol{w}) \mid \boldsymbol{v} \neq \boldsymbol{w}, \boldsymbol{v}, \boldsymbol{w} \in M \cap (2\mathbb{N})^n\}$ as the set of averages of distinct even points in M. A subset $\mathscr{A} \subseteq (2\mathbb{N})^n$ is called a *trellis* if $\mathscr{A}$ comprises the vertices of a simplex. For a trellis $\mathscr{A}$, we call M an *$\mathscr{A}$-mediated set* if $\mathscr{A} \subseteq M \subseteq \overline{A}(M) \cup \mathscr{A}$ ([9, 23, 24]).

Theorem 3.1. *Let $f = \sum_{\boldsymbol{\alpha} \in \mathscr{A}} c_{\boldsymbol{\alpha}} \mathbf{x}^{\boldsymbol{\alpha}} - d\mathbf{x}^{\boldsymbol{\beta}} \in \mathbb{R}[\mathbf{x}]$ with $d \neq 0$ be a nonnegative circuit. Then f is a sum of binomial squares iff there exists an $\mathscr{A}$-mediated set containing $\boldsymbol{\beta}$. Moreover, suppose that $\boldsymbol{\beta}$ belongs to an $\mathscr{A}$-mediated set M and for each $\boldsymbol{u} \in M \backslash \mathscr{A}$, let us write $\boldsymbol{u} = \frac{1}{2}(\boldsymbol{v_u} + \boldsymbol{w_u})$ for some $\boldsymbol{v_u} \neq \boldsymbol{w_u} \in M \cap (2\mathbb{N})^n$. Then one has the decomposition $f = \sum_{\boldsymbol{u} \in M \backslash \mathscr{A}} (a_{\boldsymbol{u}} \mathbf{x}^{\frac{1}{2}\boldsymbol{v_u}} - b_{\boldsymbol{u}} \mathbf{x}^{\frac{1}{2}\boldsymbol{w_u}})^2$, with $a_{\boldsymbol{u}}, b_{\boldsymbol{u}} \in \mathbb{R}$.*

Proof. It follows from Theorem 5.2 in [10]. □

By Theorem 3.1, if we want to represent a nonnegative circuit polynomial as a sum of binomial squares, we need to first decide if there exists an $\mathscr{A}$-mediated set containing a given lattice point and then to compute one if there exists. However, there are obstacles for each of these two steps: (1) there may not exist such an $\mathscr{A}$-mediated set containing a given lattice point; (2) even if such a set exists, there is no efficient algorithm to compute it. In order to overcome these two difficulties, we introduce the concept of $\mathscr{A}$-rational mediated sets as a replacement of $\mathscr{A}$-mediated sets by admitting rational numbers in coordinates.

Concretely, for a subset $M \subseteq \mathbb{Q}^n$, let us define $\widetilde{A}(M) := \{\frac{1}{2}(\boldsymbol{v} + \boldsymbol{w}) \mid \boldsymbol{v} \neq \boldsymbol{w}, \boldsymbol{v}, \boldsymbol{w} \in M\}$ as the set of averages of distinct rational points in M. Let us assume that $\mathscr{A} \subseteq \mathbb{Q}^n$ comprises the vertices of a simplex. We say that M is an *$\mathscr{A}$-rational mediated set* if $\mathscr{A} \subseteq M \subseteq \widetilde{A}(M) \cup \mathscr{A}$. We shall see that for a trellis $\mathscr{A}$ and a lattice point $\boldsymbol{\beta} \in \mathrm{conv}(\mathscr{A})^\circ$, an $\mathscr{A}$-rational mediated set containing $\boldsymbol{\beta}$ always exists and moreover, there is an effective algorithm to compute it.

First, let us consider the one dimensional case. For a sequence of integer numbers $A = \{s, q_1, \ldots, q_m, p\}$ (arranged from small to large), if every q_i is an average of two distinct numbers in A, then we say A is an *(s,p)-mediated sequence*. Note that the property of (s,p)-mediated sequences is preserved under translations, that

is, there is a one-to-one correspondence between (s,p)-mediated sequences and $(s+r,p+r)$-mediated sequences for any integer number r. So it suffices to consider the case of $s=0$.

For a fixed p and an integer $0<q<p$, a *minimal* $(0,p)$*-mediated sequence* containing q is a $(0,p)$-mediated sequence containing q with the least number of elements. Denote the number of elements in a minimal $(0,p)$-mediated sequence containing q by $N(\frac{q}{p})$. One can then easily show that $N(\frac{1}{p})=\lceil\log_2(p)\rceil+2$ by induction on p. We conjecture that this formula holds for general q, i.e.,

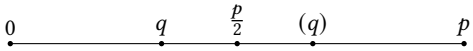
Conjecture 3.2. *If* $gcd(p,q)=1$, *then* $N(\frac{q}{p})=\lceil\log_2(p)\rceil+2$.

Generally we do not know how to compute a minimal $(0,p)$-mediated sequence containing a given q. However, we have an algorithm to compute an approximately minimal $(0,p)$-mediated sequence containing a given q as the following lemma shows.

Lemma 3.3. *For* $0<q<p\in\mathbb{N}$, *there exists a* $(0,p)$*-mediated sequence containing* q *with the cardinality less than* $\frac{1}{2}(\log_2(p)+\frac{3}{2})^2$.

Proof. We can assume $\gcd(p,q)=1$ (otherwise one can consider $p/\gcd(p,q), q/\gcd(p,q)$ instead). Let us do induction on p. Assume that for any $p',q'\in\mathbb{N}, 0<q'<p'<p$, there exists a $(0,p')$-mediated sequence containing q' with the number of elements less than $\frac{1}{2}(\log_2(p')+\frac{3}{2})^2$.

Case 1: Suppose that p is an even number. If $q=\frac{p}{2}$, then by $\gcd(p,q)=1$, we have $q=1$ and $A=\{0,1,2\}$ is a $(0,p)$-mediated sequence containing q. Otherwise, we have either $0<q<\frac{p}{2}$ or $\frac{p}{2}<q<p$. For $0<q<\frac{p}{2}$, by the induction hypothesis, there exists a $(0,\frac{p}{2})$-mediated sequence A' containing q. For $\frac{p}{2}<q<p$, since the property of mediated sequences is preserved under translations, one can first subtract $\frac{p}{2}$ and obtain a $(0,\frac{p}{2})$-mediated sequence containing $q-\frac{p}{2}$ by the induction hypothesis. Then by adding $\frac{p}{2}$, one obtains a $(\frac{p}{2},p)$-mediated sequence A' containing q. It follows that $A=A'\cup\{p\}$ or $A=\{0\}\cup A'$ is a $(0,p)$-mediated sequence containing q.

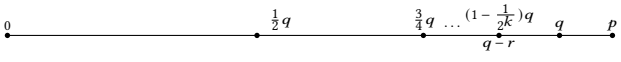


Moreover, we have

$$\#A=1+\#A'<1+\frac{1}{2}(\log_2(\frac{p}{2})+\frac{3}{2})^2<\frac{1}{2}(\log_2(p)+\frac{3}{2})^2.$$

Case 2: Suppose that p is an odd number. Without loss of generality, assume that q is an even number (otherwise one can consider $p-q$ instead and then obtain a $(0,p)$-mediated sequence containing q through the map $x\mapsto p-x$ which clearly preserves the property of mediated sequences).

Let $q=2^k r$ for some $k,r\in\mathbb{N}\backslash\{0\}$ and $2\nmid r$. If $q=p-r$, then $q=\frac{q-r+p}{2}$. Since $\gcd(p,q)=1$, we have $r=1$. Let $A=\{0,\frac{1}{2}q,\frac{3}{4}q,\ldots,(1-\frac{1}{2^k})q,q,p\}$. For $1\le i\le k$, we have $(1-\frac{1}{2^i})q=\frac{1}{2}(1-\frac{1}{2^{i-1}})q+\frac{1}{2}q$. Therefore, A is a $(0,p)$-mediated sequence containing q.

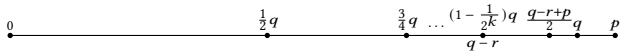


$$\#A=k+3<\frac{1}{2}(\log_2(2^k+1)+\frac{3}{2})^2=\frac{1}{2}(\log_2(p)+\frac{3}{2})^2.$$

If $q<p-r$, then q lies on the line segment between $q-r$ and $\frac{q-r+p}{2}$. Since $\frac{q-r+p}{2}-(q-r)=\frac{p+r-q}{2}<p$, then by the induction hypothesis, there exists a $(q-r,\frac{q-r+p}{2})$-mediated sequence A' containing q (using translations). It follows that $A=\{0,\frac{1}{2}q,\frac{3}{4}q,\ldots,(1-\frac{1}{2^{k-1}})q,p\}\cup A'$ is a $(0,p)$-mediated sequence containing q.

$$\begin{aligned}\#A=k+1+\#A'&<\log_2(\frac{q}{r})+1+\frac{1}{2}(\log_2(\frac{p+r-q}{2})+\frac{3}{2})^2\\&<\log_2(p)+1+\frac{1}{2}(\log_2(\frac{p}{2})+\frac{3}{2})^2\\&=\frac{1}{2}(\log_2(p)+\frac{3}{2})^2.\end{aligned}$$

If $q>p-r$, then q lies on the line segment between $\frac{q-r+p}{2}$ and p. Since $p-\frac{q-r+p}{2}=\frac{p+r-q}{2}<p$, then by the induction hypothesis, there exists a $(\frac{q-r+p}{2},p)$-mediated sequence A' containing q (using translations). It follows that the set $A=\{0,\frac{1}{2}q,\frac{3}{4}q,\ldots,(1-\frac{1}{2^k})q\}\cup A'$ is a $(0,p)$-mediated sequence containing q.



As previously, we have $\#A=k+1+\#A'<\frac{1}{2}(\log_2(p)+\frac{3}{2})^2$. □

Lemma 3.4. *Suppose that* $\boldsymbol{\alpha}_1$ *and* $\boldsymbol{\alpha}_2$ *are two rational points, and* $\boldsymbol{\beta}$ *is any rational point on the line segment between* $\boldsymbol{\alpha}_1$ *and* $\boldsymbol{\alpha}_2$. *Then there exists an* $\{\boldsymbol{\alpha}_1,\boldsymbol{\alpha}_2\}$*-rational mediated set* M *containing* $\boldsymbol{\beta}$. *Furthermore, if the denominators of coordinates of* $\boldsymbol{\alpha}_1,\boldsymbol{\alpha}_2,\boldsymbol{\beta}$ *are odd numbers, and the numerators of coordinates of* $\boldsymbol{\alpha}_1,\boldsymbol{\alpha}_2$ *are even numbers, then we can ensure that the denominators of coordinates of points in* M *are odd numbers and the numerators of coordinates of points in* $M\backslash\{\boldsymbol{\beta}\}$ *are even numbers.*

Proof. Suppose $\boldsymbol{\beta}=(1-\frac{q}{p})\boldsymbol{\alpha}_1+\frac{q}{p}\boldsymbol{\alpha}_2$, $p,q\in\mathbb{N}, 0<q<p$,$\gcd(p,q)=1$. We then construct a one-to-one correspondence between the points on the one-dimensional number axis and the points on the line across $\boldsymbol{\alpha}_1$ and $\boldsymbol{\alpha}_2$ via the map: $s\mapsto(1-\frac{s}{p})\boldsymbol{\alpha}_1+\frac{s}{p}\boldsymbol{\alpha}_2$, such that $\boldsymbol{\alpha}_1$ corresponds to the origin, $\boldsymbol{\alpha}_2$ corresponds to p and $\boldsymbol{\beta}$ corresponds to q. Then it is clear that a $(0,p)$-mediated sequence containing q corresponds to a $\{\boldsymbol{\alpha}_1,\boldsymbol{\alpha}_2\}$-rational mediated set containing $\boldsymbol{\beta}$. Hence by Lemma 3.3, there exists a $\{\boldsymbol{\alpha}_1,\boldsymbol{\alpha}_2\}$-rational mediated set M containing $\boldsymbol{\beta}$ with the number of elements less than $\frac{1}{2}(\log_2(p)+\frac{3}{2})^2$. Moreover, we can see that if $\boldsymbol{\alpha}_1,\boldsymbol{\alpha}_2,\boldsymbol{\beta}$ are lattice points, then the elements in M are also lattice points.

If the denominators of coordinates of $\boldsymbol{\alpha}_1,\boldsymbol{\alpha}_2,\boldsymbol{\beta}$ are odd numbers, and the numerators of coordinates of $\boldsymbol{\alpha}_1,\boldsymbol{\alpha}_2$ are even numbers, assume that the least common multiple of denominators appearing in the coordinates of $\boldsymbol{\alpha}_1,\boldsymbol{\alpha}_2,\boldsymbol{\beta}$ is r and then remove the denominators by multiplying the coordinates of $\boldsymbol{\alpha}_1,\boldsymbol{\alpha}_2,\boldsymbol{\beta}$ by r such that $r\boldsymbol{\alpha}_1, r\boldsymbol{\alpha}_2$ are even lattice points. If $r\boldsymbol{\beta}$ is even, let M' be the $\{\frac{r}{2}\boldsymbol{\alpha}_1,\frac{r}{2}\boldsymbol{\alpha}_2\}$-rational mediated set containing $\frac{r}{2}\boldsymbol{\beta}$ obtained as above (the elements in M' are lattice points). Then $M=\frac{2}{r}M':=\{\frac{2}{r}\boldsymbol{u}\mid\boldsymbol{u}\in M'\}$

is an $\{\boldsymbol{\alpha}_1, \boldsymbol{\alpha}_2\}$-rational mediated set containing $\boldsymbol{\beta}$ such that the denominators of coordinates of points in M are odd numbers and the numerators of coordinates of points in $M\backslash\{\boldsymbol{\beta}\}$ are even numbers.

If $r\boldsymbol{\beta}$ is not even, assume without loss of generality that $\boldsymbol{\beta}$ lies on the line segment between $\boldsymbol{\alpha}_1$ and $\frac{\boldsymbol{\alpha}_1+\boldsymbol{\alpha}_2}{2}$. Let $\boldsymbol{\beta}' = 2\boldsymbol{\beta}-\boldsymbol{\alpha}_1$ with $r\boldsymbol{\beta}'$ an even lattice point. Let M' be the $\{\frac{r}{2}\boldsymbol{\alpha}_1, \frac{r}{2}\boldsymbol{\alpha}_2\}$-rational mediated set containing $\frac{r}{2}\boldsymbol{\beta}'$ obtained as above (note that the elements in M' are lattice points). Then $M = \frac{2}{r}M'\cup\{\boldsymbol{\beta}\}$ is an $\{\boldsymbol{\alpha}_1, \boldsymbol{\alpha}_2\}$-rational mediated set containing $\boldsymbol{\beta}$ such that the denominators of coordinates of points in M are odd numbers and the numerators of coordinates of points in $M\backslash\{\boldsymbol{\beta}\}$ are even numbers as desired. □

Lemma 3.5. *For a trellis $\mathscr{A} = \{\boldsymbol{\alpha}_1, \ldots, \boldsymbol{\alpha}_m\}$ and a lattice point $\boldsymbol{\beta} \in \mathrm{conv}(\mathscr{A})^\circ$, there exists an $\mathscr{A}$-rational mediated set $M_{\mathscr{A}\boldsymbol{\beta}}$ containing $\boldsymbol{\beta}$ such that the denominators of coordinates of points in $M_{\mathscr{A}\boldsymbol{\beta}}$ are odd numbers and the numerators of coordinates of points in $M_{\mathscr{A}\boldsymbol{\beta}}\backslash\{\boldsymbol{\beta}\}$ are even numbers.*

Proof. Suppose $\boldsymbol{\beta} = \sum_{i=1}^m \frac{q_i}{p}\boldsymbol{\alpha}_i$, where $p = \sum_{i=1}^m q_i$, $p, q_i \in \mathbb{N}\backslash\{0\}$, $(p, q_1, \ldots, q_m) = 1$. If p is an even number, then because $(p, q_1, \ldots, q_m) = 1$, there must exist an odd number among the q_i's. Without loss of generality assume q_1 is an odd number. If p is an odd number and there exists an even number among the q_i's, then without loss of generality assume q_1 is an even number. In any of these two cases, we have

$$\boldsymbol{\beta} = \frac{q_1}{p}\boldsymbol{\alpha}_1 + \frac{p-q_1}{p}\left(\frac{q_2}{p-q_1}\boldsymbol{\alpha}_2 + \cdots + \frac{q_m}{p-q_1}\boldsymbol{\alpha}_m\right).$$

Let $\boldsymbol{\beta}_1 = \frac{q_2}{p-q_1}\boldsymbol{\alpha}_2 + \cdots + \frac{q_m}{p-q_1}\boldsymbol{\alpha}_m$. Then $\boldsymbol{\beta} = \frac{q_1}{p}\boldsymbol{\alpha}_1 + \frac{p-q_1}{p}\boldsymbol{\beta}_1$.

If p is an odd number and all q_i's are odd numbers, then we have

$$\begin{aligned}\boldsymbol{\beta} = {} & \frac{q_1}{q_1+q_2}\left(\frac{q_1+q_2}{p}\boldsymbol{\alpha}_1 + \frac{q_3}{p}\boldsymbol{\alpha}_3 + \cdots + \frac{q_m}{p}\boldsymbol{\alpha}_m\right) \\ & + \frac{q_2}{q_1+q_2}\left(\frac{q_1+q_2}{p}\boldsymbol{\alpha}_2 + \frac{q_3}{p}\boldsymbol{\alpha}_3 + \cdots + \frac{q_m}{p}\boldsymbol{\alpha}_m\right).\end{aligned}$$

Let $\boldsymbol{\beta}_1 = \frac{q_1+q_2}{p}\boldsymbol{\alpha}_1 + \frac{q_3}{p}\boldsymbol{\alpha}_3 + \cdots + \frac{q_m}{p}\boldsymbol{\alpha}_m$ and $\boldsymbol{\beta}_2 = \frac{q_1+q_2}{p}\boldsymbol{\alpha}_2 + \frac{q_3}{p}\boldsymbol{\alpha}_3 + \cdots + \frac{q_m}{p}\boldsymbol{\alpha}_m$. Then $\boldsymbol{\beta} = \frac{q_1}{q_1+q_2}\boldsymbol{\beta}_1 + \frac{q_2}{q_1+q_2}\boldsymbol{\beta}_2$.

Apply the same procedure for $\boldsymbol{\beta}_1$ (and $\boldsymbol{\beta}_2$), and continue iteratively. Eventually we obtain a set of points $\{\boldsymbol{\beta}_i\}_{i=1}^l$ such that for each i, $\boldsymbol{\beta}_i = \lambda_i\boldsymbol{\beta}_j + \mu_i\boldsymbol{\beta}_k$ or $\boldsymbol{\beta}_i = \lambda_i\boldsymbol{\beta}_j + \mu_i\boldsymbol{\alpha}_k$ or $\boldsymbol{\beta}_i = \lambda_i\boldsymbol{\alpha}_j + \mu_i\boldsymbol{\alpha}_k$, where $\lambda_i + \mu_i = 1, \lambda_i, \mu_i > 0$. We claim the denominators of coordinates of $\boldsymbol{\beta}_i$ are odd numbers, and the numerators of coordinates of $\boldsymbol{\beta}_i$ are even numbers. This is because for each $\boldsymbol{\beta}_i$, we have the expression $\boldsymbol{\beta}_i = \sum_j \frac{s_j}{r}\boldsymbol{\alpha}_j$, where r is an odd number and all $\boldsymbol{\alpha}_j$'s are even lattice points. For $\boldsymbol{\beta}_i = \lambda\boldsymbol{\beta}_j + \mu\boldsymbol{\beta}_k$ (or $\boldsymbol{\beta}_i = \lambda\boldsymbol{\beta}_j + \mu\boldsymbol{\alpha}_k$, $\boldsymbol{\beta}_i = \lambda\boldsymbol{\alpha}_j + \mu\boldsymbol{\alpha}_k$ respectively), let M_i be the $\{\boldsymbol{\beta}_j, \boldsymbol{\beta}_k\}$- (or $\{\boldsymbol{\beta}_j, \boldsymbol{\alpha}_k\}$-, $\{\boldsymbol{\alpha}_j, \boldsymbol{\alpha}_k\}$- respectively) rational mediated set containing $\boldsymbol{\beta}_i$ obtained by Lemma 3.4 such that the denominators of coordinates of points in M_i are odd numbers and the numerators of coordinates of points in $M_i\backslash\{\boldsymbol{\beta}\}$ are even numbers for $i = 0, \ldots, l$ (set $\boldsymbol{\beta}_0 = \boldsymbol{\beta}$). Let $M_{\mathscr{A}\boldsymbol{\beta}} = \cup_{i=0}^l M_i$. Then $M_{\mathscr{A}\boldsymbol{\beta}}$ is clearly an $\mathscr{A}$-rational mediated set containing $\boldsymbol{\beta}$ with the desired property. □

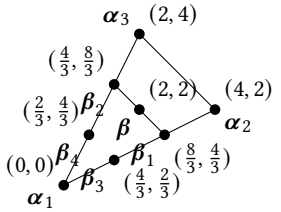
3.2 Decomposing SONC with binomial squares

For $r \in \mathbb{N}$ and $f(\mathbf{x}) \in \mathbb{R}[\mathbf{x}]$, let $f(\mathbf{x}^r) := f(x_1^r, \ldots, x_n^r)$. For any odd $r \in \mathbb{N}$, $f(\mathbf{x}) = \sum_{\boldsymbol{\alpha}\in\mathscr{A}} c_{\boldsymbol{\alpha}}\mathbf{x}^{\boldsymbol{\alpha}} - d\mathbf{x}^{\boldsymbol{\beta}}$ is a nonnegative circuit iff $f(\mathbf{x}^r) = \sum_{\boldsymbol{\alpha}\in\mathscr{A}} c_{\boldsymbol{\alpha}}\mathbf{x}^{r\boldsymbol{\alpha}} - d\mathbf{x}^{r\boldsymbol{\beta}}$ is a nonnegative circuit.

Theorem 3.6. *Let $f = \sum_{\boldsymbol{\alpha}\in\mathscr{A}} c_{\boldsymbol{\alpha}}\mathbf{x}^{\boldsymbol{\alpha}} - d\mathbf{x}^{\boldsymbol{\beta}} \in \mathbb{R}[\mathbf{x}]$ with $d \neq 0$ be a circuit polynomial. Assume that $M_{\mathscr{A}\boldsymbol{\beta}}$ is the $\mathscr{A}$-rational mediated set containing $\boldsymbol{\beta}$ provided by Lemma 3.5. and for each $\boldsymbol{u} \in M_{\mathscr{A}\boldsymbol{\beta}}\backslash\mathscr{A}$, let $\boldsymbol{u} = \frac{1}{2}(\boldsymbol{v_u}+\boldsymbol{w_u}), \boldsymbol{v_u} \neq \boldsymbol{w_u} \in M_{\mathscr{A}\boldsymbol{\beta}}$. Then f is nonnegative iff f can be written as $f = \sum_{\boldsymbol{u}\in M_{\mathscr{A}\boldsymbol{\beta}}\backslash\mathscr{A}}(a_{\boldsymbol{u}}\mathbf{x}^{\frac{1}{2}\boldsymbol{v_u}} - b_{\boldsymbol{u}}\mathbf{x}^{\frac{1}{2}\boldsymbol{w_u}})^2$, $a_{\boldsymbol{u}}, b_{\boldsymbol{u}} \in \mathbb{R}$.*

Proof. Assume that the least common multiple of denominators appearing in the coordinates of points in $M_{\mathscr{A}\boldsymbol{\beta}}$ is r, which is odd. Then $f(\mathbf{x})$ is nonnegative if and only if $f(\mathbf{x}^r)$ is nonnegative. Multiply all coordinates of points in $M_{\mathscr{A}\boldsymbol{\beta}}$ by r to remove the denominators, and the obtained $rM_{\mathscr{A}\boldsymbol{\beta}} := \{r\boldsymbol{u} \mid \boldsymbol{u} \in M_{\mathscr{A}\boldsymbol{\beta}}\}$ is an $r\mathscr{A}$-mediated set containing $r\boldsymbol{\beta}$. Hence by Theorem 3.1, $f(\mathbf{x}^r)$ is nonnegative if and only if $f(\mathbf{x}^r)$ can be written as $f(\mathbf{x}^r) = \sum_{\boldsymbol{u}\in M_{\mathscr{A}\boldsymbol{\beta}}\backslash\mathscr{A}}(a_{\boldsymbol{u}}\mathbf{x}^{\frac{r}{2}\boldsymbol{v_u}} - b_{\boldsymbol{u}}\mathbf{x}^{\frac{r}{2}\boldsymbol{w_u}})^2$, $a_{\boldsymbol{u}}, b_{\boldsymbol{u}} \in \mathbb{R}$, which is equivalent to $f(\mathbf{x}) = \sum_{\boldsymbol{u}\in M_{\mathscr{A}\boldsymbol{\beta}}\backslash\mathscr{A}}(a_{\boldsymbol{u}}\mathbf{x}^{\frac{1}{2}\boldsymbol{v_u}} - b_{\boldsymbol{u}}\mathbf{x}^{\frac{1}{2}\boldsymbol{w_u}})^2$. □

Example 3.7. Let $f = x^4y^2 + x^2y^4 + 1 - 3x^2y^2$ and $\mathscr{A} = \{\boldsymbol{\alpha}_1 = (0,0), \boldsymbol{\alpha}_2 = (4,2), \boldsymbol{\alpha}_3 = (2,4)\}$, $\boldsymbol{\beta} = (2,2)$. Let $\boldsymbol{\beta}_1 = \frac{1}{3}\boldsymbol{\alpha}_1 + \frac{2}{3}\boldsymbol{\alpha}_2$ and $\boldsymbol{\beta}_2 = \frac{1}{3}\boldsymbol{\alpha}_1 + \frac{2}{3}\boldsymbol{\alpha}_3$ such that $\boldsymbol{\beta} = \frac{1}{2}\boldsymbol{\beta}_1 + \frac{1}{2}\boldsymbol{\beta}_2$. Let $\boldsymbol{\beta}_3 = \frac{2}{3}\boldsymbol{\alpha}_1 + \frac{1}{3}\boldsymbol{\alpha}_2$ and $\boldsymbol{\beta}_4 = \frac{2}{3}\boldsymbol{\alpha}_1 + \frac{1}{3}\boldsymbol{\alpha}_3$. Then $M = \{\boldsymbol{\alpha}_1, \boldsymbol{\alpha}_2, \boldsymbol{\alpha}_3, \boldsymbol{\beta}, \boldsymbol{\beta}_1, \boldsymbol{\beta}_2, \boldsymbol{\beta}_3, \boldsymbol{\beta}_4\}$ is an $\mathscr{A}$-rational mediated set containing $\boldsymbol{\beta}$.



By Theorem 3.6, one has $f = x^4y^2+x^2y^4+1-3x^2y^2 = (a_1x^{\frac{2}{3}}y^{\frac{4}{3}} - b_1x^{\frac{4}{3}}y^{\frac{2}{3}})^2+(a_2xy^2-b_2x^{\frac{1}{3}}y^{\frac{2}{3}})^2+(a_3x^{\frac{2}{3}}y^{\frac{4}{3}}-b_3)^2+(a_4x^2y-b_4x^{\frac{2}{3}}y^{\frac{1}{3}})^2+(a_5x^{\frac{4}{3}}y^{\frac{2}{3}} - b_5)^2$. Comparing coefficients yields $f = \frac{3}{2}(x^{\frac{2}{3}}y^{\frac{4}{3}} - x^{\frac{4}{3}}y^{\frac{2}{3}})^2+(xy^2-x^{\frac{1}{3}}y^{\frac{2}{3}})^2+\frac{1}{2}(x^{\frac{2}{3}}y^{\frac{4}{3}}-1)^2+(x^2y-x^{\frac{2}{3}}y^{\frac{1}{3}})^2+\frac{1}{2}(x^{\frac{4}{3}}y^{\frac{2}{3}} - 1)^2$, a sum of five binomial squares with rational exponents.

Lemma 3.8. *Let $f(\mathbf{x}) \in \mathbb{R}[\mathbf{x}]$. For an odd number r, $f(\mathbf{x}) \in$ SONC if and only if $f(\mathbf{x}^r) \in$ SONC.*

Proof. It comes from the fact that $f(\mathbf{x})$ is a nonnegative circuit iff $f(\mathbf{x}^r)$ is a nonnegative circuit for an odd number r. □

Theorem 3.9. *Let $f = \sum_{\boldsymbol{\alpha}\in\Lambda(f)} c_{\boldsymbol{\alpha}}\mathbf{x}^{\boldsymbol{\alpha}} - \sum_{\boldsymbol{\beta}\in\Gamma(f)} d_{\boldsymbol{\beta}}\mathbf{x}^{\boldsymbol{\beta}} \in \mathbb{R}[\mathbf{x}]$. Let $\mathscr{F}(\boldsymbol{\beta})$ be as in (1). For every $\boldsymbol{\beta} \in \Gamma(f)$ and every $\Delta \in \mathscr{F}(\boldsymbol{\beta})$, let $M_{\boldsymbol{\beta}\Delta}$ be the $V(\Delta)$-rational mediated set containing $\boldsymbol{\beta}$ provided by Lemma 3.5. Let $M = \cup_{\boldsymbol{\beta}\in\Gamma(f)}\cup_{\Delta\in\mathscr{F}(\boldsymbol{\beta})} M_{\boldsymbol{\beta}\Delta}$. For each $\boldsymbol{u} \in M\backslash\Lambda(f)$, let $\boldsymbol{u} = \frac{1}{2}(\boldsymbol{v_u}+\boldsymbol{w_u}), \boldsymbol{v_u} \neq \boldsymbol{w_u} \in M$. Let $\tilde{\mathscr{A}} = \{\boldsymbol{\alpha} \in \Lambda(f) \mid \boldsymbol{\alpha} \notin \cup_{\boldsymbol{\beta}\in\Gamma(f)}\cup_{\Delta\in\mathscr{F}(\boldsymbol{\beta})} V(\Delta)\}$. Then $f \in$ SONC iff f can be written as $f = \sum_{\boldsymbol{u}\in M\backslash\Lambda(f)}(a_{\boldsymbol{u}}\mathbf{x}^{\frac{1}{2}\boldsymbol{v_u}} - b_{\boldsymbol{u}}\mathbf{x}^{\frac{1}{2}\boldsymbol{w_u}})^2 + \sum_{\boldsymbol{\alpha}\in\tilde{\mathscr{A}}} c_{\boldsymbol{\alpha}}\mathbf{x}^{\boldsymbol{\alpha}}$, $a_{\boldsymbol{u}}, b_{\boldsymbol{u}} \in \mathbb{R}$.*

Proof. Suppose $f \in$ SONC. By Theorem 5.5 in [27], we can write f as $f = \sum_{\boldsymbol{\beta}\in\Gamma(f)}\sum_{\Delta\in\mathscr{F}(\boldsymbol{\beta})} f_{\boldsymbol{\beta}\Delta} + \sum_{\boldsymbol{\alpha}\in\tilde{\mathscr{A}}} c_{\boldsymbol{\alpha}}\mathbf{x}^{\boldsymbol{\alpha}}$ such that every $f_{\boldsymbol{\beta}\Delta} = \sum_{\boldsymbol{\alpha}\in V(\Delta)} c_{\boldsymbol{\beta}\Delta\boldsymbol{\alpha}}\mathbf{x}^{\boldsymbol{\alpha}} - d_{\boldsymbol{\beta}\Delta}\mathbf{x}^{\boldsymbol{\beta}}$ is a nonnegative circuit polynomial. We have $f_{\boldsymbol{\beta}\Delta} = \sum_{\boldsymbol{u}\in M_{\mathscr{A}\boldsymbol{\beta}}\backslash\mathscr{A}}(a_{\boldsymbol{u}}\mathbf{x}^{\frac{1}{2}\boldsymbol{v_u}} - b_{\boldsymbol{u}}\mathbf{x}^{\frac{1}{2}\boldsymbol{w_u}})^2$, $a_{\boldsymbol{u}}, b_{\boldsymbol{u}} \in \mathbb{R}$ by Theorem 3.6. Thus $f = \sum_{\boldsymbol{u}\in M\backslash\Lambda(f)}(a_{\boldsymbol{u}}\mathbf{x}^{\frac{1}{2}\boldsymbol{v_u}} - b_{\boldsymbol{u}}\mathbf{x}^{\frac{1}{2}\boldsymbol{w_u}})^2 + \sum_{\boldsymbol{\alpha}\in\tilde{\mathscr{A}}} c_{\boldsymbol{\alpha}}\mathbf{x}^{\boldsymbol{\alpha}}$, $a_{\boldsymbol{u}}, b_{\boldsymbol{u}} \in \mathbb{R}$. Suppose $f = \sum_{\boldsymbol{u}\in M\backslash\Lambda(f)}(a_{\boldsymbol{u}}\mathbf{x}^{\frac{1}{2}\boldsymbol{v_u}} -$

$b_{\boldsymbol{u}}\mathbf{x}^{\frac{1}{2}\boldsymbol{w_u}})^2 + \sum_{\boldsymbol{\alpha}\in\tilde{\mathscr{A}}} c_{\boldsymbol{\alpha}}\mathbf{x}^{\boldsymbol{\alpha}}$, $a_{\boldsymbol{u}}, b_{\boldsymbol{u}} \in \mathbb{R}$. Assume that the least common multiple of denominators appearing in the coordinates of points in M is r, which is odd. Then $f(\mathbf{x}^r) = \sum_{\boldsymbol{u}\in M\backslash\Lambda(f)}(a_{\boldsymbol{u}}\mathbf{x}^{\frac{r}{2}\boldsymbol{v_u}} - b_{\boldsymbol{u}}\mathbf{x}^{\frac{r}{2}\boldsymbol{w_u}})^2 + \sum_{\boldsymbol{\alpha}\in\tilde{\mathscr{A}}} c_{\boldsymbol{\alpha}}\mathbf{x}^{r\boldsymbol{\alpha}}$, $a_{\boldsymbol{u}}, b_{\boldsymbol{u}} \in \mathbb{R}$, which is a SONC since every binomial square (and monomial square) is a nonnegative circuit. Hence by Lemma 3.8, $f(\mathbf{x}) \in$ SONC. □

4 SOC REPRESENTATIONS OF SONC CONES

SOCP plays an important role in convex optimization and can be handled via very efficient algorithms. If an SOC representation exists for a given convex cone, then it is possible to design efficient algorithms for optimization problems over the convex cone. In [8], Fawzi proved that PSD cones do not admit any SOC representations in general, which implies that SOS cones do not admit any SOC representations in general. In this section, we prove that dramatically unlike the SOS cones, SONC cones always admit SOC representations. Let $Q^k := Q \times \cdots \times Q$ be the Cartesian product of k copies of an SOC Q. A *linear slice* of Q^k is an intersection of Q^k with a linear subspace.

Definition 4.1. A convex cone $C \subseteq \mathbb{R}^m$ has a *SOC lift of size* k (or simply a Q^k*-lift*) if it can be written as the projection of a slice of Q^k, that is, there is a subspace L of Q^k and a linear map $\pi\colon Q^k \to \mathbb{R}^m$ such that $C = \pi(Q^k \cap L)$.

Definition 4.2. Given sets of lattice points $\mathscr{A} \subseteq (2\mathbb{N})^n$, $\mathscr{B}_1 \subseteq \mathrm{conv}(\mathscr{A}) \cap (2\mathbb{N})^n$ and $\mathscr{B}_2 \subseteq \mathrm{conv}(\mathscr{A}) \cap (\mathbb{N}^n\backslash(2\mathbb{N})^n)$ such that $\mathscr{A} \cap \mathscr{B}_1 = \varnothing$, define the SONC cone supported on $\mathscr{A}, \mathscr{B}_1, \mathscr{B}_2$ as

$$\mathrm{SONC}_{\mathscr{A},\mathscr{B}_1,\mathscr{B}_2} := \{(\mathbf{c}_{\mathscr{A}}, \mathbf{d}_{\mathscr{B}_1}, \mathbf{d}_{\mathscr{B}_2}) \in \mathbb{R}_+^{|\mathscr{A}|} \times \mathbb{R}_+^{|\mathscr{B}_1|} \times \mathbb{R}^{|\mathscr{B}_2|} \mid \sum_{\boldsymbol{\alpha}\in\mathscr{A}} c_{\boldsymbol{\alpha}}\mathbf{x}^{\boldsymbol{\alpha}} - \sum_{\boldsymbol{\beta}\in\mathscr{B}_1\cup\mathscr{B}_2} d_{\boldsymbol{\beta}}\mathbf{x}^{\boldsymbol{\beta}} \in \mathrm{SONC}\},$$

where $\mathbf{c}_{\mathscr{A}} = (c_{\boldsymbol{\alpha}})_{\boldsymbol{\alpha}\in\mathscr{A}}$, $\mathbf{d}_{\mathscr{B}_1} = (d_{\boldsymbol{\beta}})_{\boldsymbol{\beta}\in\mathscr{B}_1}$ and $\mathbf{d}_{\mathscr{B}_2} = (d_{\boldsymbol{\beta}})_{\boldsymbol{\beta}\in\mathscr{B}_2}$. It is easy to check that $\mathrm{SONC}_{\mathscr{A},\mathscr{B}_1,\mathscr{B}_2}$ is indeed a convex cone.

Let $\mathbb{S}_+^2$ be the convex cone of 2×2 positive semidefinite matrices

$$\mathbb{S}_+^2 := \left\{ \begin{bmatrix} a & b \\ b & c \end{bmatrix} \in \mathbb{R}^{2\times 2} \mid \begin{bmatrix} a & b \\ b & c \end{bmatrix} \text{ is positive semidefinite} \right\}.$$

Lemma 4.3. $\mathbb{S}_+^2$ *is a* 3*-dimensional rotated SOC.*

Proof. It is immediate from the definition. □

Theorem 4.4. *For* $\mathscr{A} \subseteq (2\mathbb{N})^n$, $\mathscr{B}_1 \subseteq \mathrm{conv}(\mathscr{A}) \cap (2\mathbb{N})^n$ *and* $\mathscr{B}_2 \subseteq \mathrm{conv}(\mathscr{A}) \cap (\mathbb{N}^n\backslash(2\mathbb{N})^n)$ *such that* $\mathscr{A} \cap \mathscr{B}_1 = \varnothing$*, the convex cone* $\mathrm{SONC}_{\mathscr{A},\mathscr{B}_1,\mathscr{B}_2}$ *has an* $(\mathbb{S}_+^2)^k$*-lift for some* $k \in \mathbb{N}$.

Proof. For every $\boldsymbol{\beta} \in \mathscr{B}_1 \cup \mathscr{B}_2$, let $\mathscr{F}(\boldsymbol{\beta})$ be as in (1). Then for every $\boldsymbol{\beta} \in \mathscr{B}_1 \cup \mathscr{B}_2$ and every $\Delta \in \mathscr{F}(\boldsymbol{\beta})$, let $M_{\boldsymbol{\beta}\Delta}$ be the $V(\Delta)$-rational mediated set containing $\boldsymbol{\beta}$ provided by Lemma 3.5. Let $M = \cup_{\boldsymbol{\beta}\in\mathscr{B}_1\cup\mathscr{B}_2} \cup_{\Delta\in\mathscr{F}(\boldsymbol{\beta})} M_{\boldsymbol{\beta}\Delta}$. For each $\boldsymbol{u}_i \in M\backslash\mathscr{A}$, let us write $\boldsymbol{u}_i = \frac{1}{2}(\boldsymbol{v}_i + \boldsymbol{w}_i)$. Let $B = \cup_{\boldsymbol{u}_i\in M\backslash\mathscr{A}}\{\frac{1}{2}\boldsymbol{v}_i, \frac{1}{2}\boldsymbol{w}_i\}$, $\tilde{\mathscr{A}} = \{\boldsymbol{\alpha} \in \Lambda(f) \mid \boldsymbol{\alpha} \notin \cup_{\boldsymbol{\beta}\in\Gamma(f)} \cup_{\Delta\in\mathscr{F}(\boldsymbol{\beta})} V(\Delta)\}$ and $k = \#M\backslash\mathscr{A} + \#\tilde{\mathscr{A}}$.

Then by Theorem 3.9, a polynomial f is in $\mathrm{SONC}_{\mathscr{A},\mathscr{B}_1,\mathscr{B}_2}$ if and only if f can be written as $f = \sum_{\boldsymbol{u}_i\in M\backslash\mathscr{A}}(a_i\mathbf{x}^{\frac{1}{2}\boldsymbol{v}_i} - b_i\mathbf{x}^{\frac{1}{2}\boldsymbol{w}_i})^2 + \sum_{\boldsymbol{\alpha}\in\tilde{\mathscr{A}}} c_{\boldsymbol{\alpha}}\mathbf{x}^{\boldsymbol{\alpha}}$, $a_i, b_i \in \mathbb{R}$, which is equivalent to the existence of a symmetric matrix $Q = \sum_{i=1}^k Q_i$ such that $f = (\mathbf{x}^B)^T Q \mathbf{x}^B$ with $\mathbf{x}^B := (\mathbf{x}^{\boldsymbol{\beta}})_{\boldsymbol{\beta}\in B}$, where Q_i is a symmetric matrix with zeros everywhere except either at the four positions corresponding to the monomials $\mathbf{x}^{\frac{1}{2}\boldsymbol{v}_i}, \mathbf{x}^{\frac{1}{2}\boldsymbol{w}_i}$ or at the position corresponding to a monomial $\mathbf{x}^{\frac{1}{2}\boldsymbol{\alpha}}$ for some $\boldsymbol{\alpha} \in \tilde{\mathscr{A}}$. This leads respectively to either four entries forming a 2×2 positive semidefinite submatrix or one single positive entry.

Let $\pi : (\mathbb{S}_+^2)^k \to \mathrm{SONC}_{\mathscr{A},\mathscr{B}_1,\mathscr{B}_2}$ be the linear map that maps an element in $Q_1 \times \cdots \times Q_k$ to the coefficient vector of f which is in $\mathrm{SONC}_{\mathscr{A},\mathscr{B}_1,\mathscr{B}_2}$ via the equality $f = (\mathbf{x}^B)^T Q \mathbf{x}^B$ with $Q = \sum_{i=1}^k Q_i$. So we obtain an $(\mathbb{S}_+^2)^k$-lift for $\mathrm{SONC}_{\mathscr{A},\mathscr{B}_1,\mathscr{B}_2}$.

□

5 SONC OPTIMIZATION VIA SOCP

In this section, we tackle the following unconstrained polynomial optimization problem via SOCP, based on the representation of SONC cones derived in the previous section:

$$\text{(P)}: \quad \sup\{\xi : f(\mathbf{x}) - \xi \geq 0, \quad \mathbf{x} \in \mathbb{R}^n\}. \tag{3}$$

Let us denote by ξ^* the optimal value of (3). Replace the nonnegativity constraint in (3) by the following one to obtain a SONC relaxation with optimal value ξ_{sonc}:

$$\text{(SONC)}: \quad \sup\{\xi : f(\mathbf{x}) - \xi \in \mathrm{SONC}\}. \tag{4}$$

5.1 Conversion to PN-polynomials

Suppose $f = \sum_{\boldsymbol{\alpha}\in\Lambda(f)} c_{\boldsymbol{\alpha}}\mathbf{x}^{\boldsymbol{\alpha}} - \sum_{\boldsymbol{\beta}\in\Gamma(f)} d_{\boldsymbol{\beta}}\mathbf{x}^{\boldsymbol{\beta}} \in \mathbb{R}[\mathbf{x}]$. If $d_{\boldsymbol{\beta}} > 0$ for all $\boldsymbol{\beta} \in \Gamma(f)$, then we call f a *PN-polynomial*. The "PN" in PN-polynomial is short for "positive part plus negative part". For a PN-polynomial $f(\mathbf{x})$, it is clear that $f(\mathbf{x}) \geq 0$ for all $\mathbf{x} \in \mathbb{R}^n$ iff $f(\mathbf{x}) \geq 0$ for all $\mathbf{x} \in \mathbb{R}_+^n$.

Lemma 5.1. *Let* $f(\mathbf{x}) \in \mathbb{R}[\mathbf{x}]$ *be a PN-polynomial. Then for any positive integer* k*,* $f(\mathbf{x}) \in$ SONC *if and only if* $f(\mathbf{x}^k) \in$ SONC.

Proof. It comes from the fact that a polynomial $f(\mathbf{x})$ with exactly one negative term is a nonnegative circuit iff $f(\mathbf{x}^k)$ is a nonnegative circuit for any positive integer $k \in \mathbb{N}$. □

Theorem 5.2. *Let* $f = \sum_{\boldsymbol{\alpha}\in\Lambda(f)} c_{\boldsymbol{\alpha}}\mathbf{x}^{\boldsymbol{\alpha}} - \sum_{\boldsymbol{\beta}\in\Gamma(f)} d_{\boldsymbol{\beta}}\mathbf{x}^{\boldsymbol{\beta}} \in \mathbb{R}[\mathbf{x}]$ *be a PN-polynomial. Let* $\mathscr{F}(\boldsymbol{\beta})$ *be as in* (1)*. For every* $\boldsymbol{\beta} \in \Gamma(f)$ *and every* $\Delta \in \mathscr{F}(\boldsymbol{\beta})$*, let* $M_{\boldsymbol{\beta}\Delta}$ *be a* $V(\Delta)$*-rational mediated set containing* $\boldsymbol{\beta}$*. Let* $M = \cup_{\boldsymbol{\beta}\in\Gamma(f)} \cup_{\Delta\in\mathscr{F}(\boldsymbol{\beta})} M_{\boldsymbol{\beta}\Delta}$ *and* $\tilde{\mathscr{A}} = \{\boldsymbol{\alpha} \in \Lambda(f) \mid \boldsymbol{\alpha} \notin \cup_{\boldsymbol{\beta}\in\Gamma(f)} \cup_{\Delta\in\mathscr{F}(\boldsymbol{\beta})} V(\Delta)\}$*. For each* $\boldsymbol{u} \in M\backslash\Lambda(f)$*, let* $\boldsymbol{u} = \frac{1}{2}(\boldsymbol{v} + \boldsymbol{w})$*. Then* $f \in$ SONC *if and only if* f *can be written as* $f = \sum_{\boldsymbol{u}\in M\backslash\Lambda(f)}(a_{\boldsymbol{u}}\mathbf{x}^{\frac{1}{2}\boldsymbol{v}} - b_{\boldsymbol{u}}\mathbf{x}^{\frac{1}{2}\boldsymbol{w}})^2 + \sum_{\boldsymbol{\alpha}\in\tilde{\mathscr{A}}} c_{\boldsymbol{\alpha}}\mathbf{x}^{\boldsymbol{\alpha}}$, $a_{\boldsymbol{u}}, b_{\boldsymbol{u}} \in \mathbb{R}$.

Proof. It follows easily from Lemma 5.1 and Theorem 3.1. □

The significant difference between Theorem 3.9 and Theorem 5.2 is that to represent a SONC PN-polynomial as a sum of binomial squares, we do not require the denominators of coordinates of points in $\mathscr{A}$-rational mediated sets to be odd. By virtue of this fact, for given trellis $\mathscr{A} = \{\boldsymbol{\alpha}_1, \ldots, \boldsymbol{\alpha}_m\}$ and lattice point $\boldsymbol{\beta} \in \mathrm{conv}(\mathscr{A})^\circ$, we can then construct an $\mathscr{A}$-rational mediated set $M_{\mathscr{A}\boldsymbol{\beta}}$ containing $\boldsymbol{\beta}$ which is smaller than that the one from Lemma 3.5.

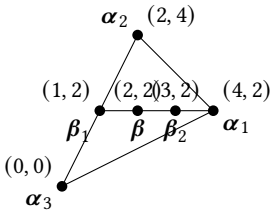
Lemma 5.3. *For a trellis* $\mathscr{A}$ *and a lattice point* $\boldsymbol{\beta} \in \mathrm{conv}(\mathscr{A})^\circ$*, there is an* $\mathscr{A}$*-rational mediated set* $M_{\mathscr{A}\boldsymbol{\beta}}$ *containing* $\boldsymbol{\beta}$*.*

Proof. Suppose that $\boldsymbol{\beta} = \sum_{i=1}^m \frac{q_i}{p}\boldsymbol{\alpha}_i$, where $p = \sum_{i=1}^m q_i$, $p, q_i \in \mathbb{N}^*$, $(p, q_1, \ldots, q_m) = 1$. We can write

$$\boldsymbol{\beta} = \frac{q_1}{p}\boldsymbol{\alpha}_1 + \frac{p-q_1}{p}(\frac{q_2}{p-q_1}\boldsymbol{\alpha}_2 + \cdots + \frac{q_m}{p-q_1}\boldsymbol{\alpha}_m).$$

Let $\boldsymbol{\beta}_1 = \frac{q_2}{p-q_1}\boldsymbol{\alpha}_2 + \cdots + \frac{q_m}{p-q_1}\boldsymbol{\alpha}_m$. Then $\boldsymbol{\beta} = \frac{q_1}{p}\boldsymbol{\alpha}_1 + \frac{p-q_1}{p}\boldsymbol{\beta}_1$. Apply the same procedure for $\boldsymbol{\beta}_1$, and continue like this. Eventually we obtain a set of points $\{\boldsymbol{\beta}_i\}_{i=0}^{m-2}$ (set $\boldsymbol{\beta}_0 = \boldsymbol{\beta}$) such that $\boldsymbol{\beta}_i = \lambda_i\boldsymbol{\alpha}_{i+1} + \mu_i\boldsymbol{\beta}_{i+1}$, $i = 0, \ldots, m-3$ and $\boldsymbol{\beta}_{m-2} = \lambda_{m-2}\boldsymbol{\alpha}_{m-1} + \mu_{m-2}\boldsymbol{\alpha}_m$, where $\lambda_i + \mu_i = 1$, $\lambda_i, \mu_i > 0$, $i = 0, \ldots, m-2$. For $\boldsymbol{\beta}_i = \lambda_i\boldsymbol{\alpha}_{i+1} + \mu_i\boldsymbol{\beta}_{i+1}$ (resp. $\boldsymbol{\beta}_{m-2} = \lambda_{m-2}\boldsymbol{\alpha}_{m-1} + \mu_{m-2}\boldsymbol{\alpha}_m$), let M_i be the $\{\boldsymbol{\alpha}_{i+1}, \boldsymbol{\beta}_{i+1}\}$- (resp. $\{\boldsymbol{\alpha}_{m-1}, \boldsymbol{\alpha}_m\}$-) rational mediated set containing $\boldsymbol{\beta}_i$ obtained by Lemma 3.4, $i = 0, \ldots, m-2$. Let $M_{\mathscr{A}\boldsymbol{\beta}} = \cup_{i=0}^{m-2} M_i$. Then clearly $M_{\mathscr{A}\boldsymbol{\beta}}$ is an $\mathscr{A}$-rational mediated set containing $\boldsymbol{\beta}$. □

Example 5.4. Let $f = x^4y^2 + x^2y^4 + 1 - 3x^2y^2$ be the Motzkin's polynomial and $\mathscr{A} = \{\boldsymbol{\alpha}_1 = (4,2), \boldsymbol{\alpha}_2 = (2,4), \boldsymbol{\alpha}_3 = (0,0)\}$, $\boldsymbol{\beta} = (2,2)$. Then $\boldsymbol{\beta} = \frac{1}{3}\boldsymbol{\alpha}_1 + \frac{1}{3}\boldsymbol{\alpha}_2 + \frac{1}{3}\boldsymbol{\alpha}_3 = \frac{1}{3}\boldsymbol{\alpha}_1 + \frac{2}{3}(\frac{1}{2}\boldsymbol{\alpha}_2 + \frac{1}{2}\boldsymbol{\alpha}_3)$. Let $\boldsymbol{\beta}_1 = \frac{1}{2}\boldsymbol{\alpha}_2 + \frac{1}{2}\boldsymbol{\alpha}_3$ such that $\boldsymbol{\beta} = \frac{1}{3}\boldsymbol{\alpha}_1 + \frac{2}{3}\boldsymbol{\beta}_1$. Let $\boldsymbol{\beta}_2 = \frac{2}{3}\boldsymbol{\alpha}_1 + \frac{1}{3}\boldsymbol{\beta}_1$. Then it is easy to check that $M = \{\boldsymbol{\alpha}_1, \boldsymbol{\alpha}_2, \boldsymbol{\alpha}_3, \boldsymbol{\beta}, \boldsymbol{\beta}_1, \boldsymbol{\beta}_2\}$ is an $\mathscr{A}$-rational mediated set containing $\boldsymbol{\beta}$.



By a simple computation, we have $f = (1-xy^2)^2 + 2(x^{\frac{1}{2}}y - x^{\frac{3}{2}}y)^2 + (xy - x^2y)^2$. Here we represent f as a sum of three binomial squares with rational exponents.

We associate to a polynomial $f = \sum_{\boldsymbol{\alpha}\in\Lambda(f)} c_{\boldsymbol{\alpha}}\mathbf{x}^{\boldsymbol{\alpha}} - \sum_{\boldsymbol{\beta}\in\Gamma(f)} d_{\boldsymbol{\beta}}\mathbf{x}^{\boldsymbol{\beta}}$, the PN-polynomial $\tilde{f} = \sum_{\boldsymbol{\alpha}\in\Lambda(f)} c_{\boldsymbol{\alpha}}\mathbf{x}^{\boldsymbol{\alpha}} - \sum_{\boldsymbol{\beta}\in\Gamma(f)} |d_{\boldsymbol{\beta}}|\mathbf{x}^{\boldsymbol{\beta}}$.

Lemma 5.5. *Suppose $f = \sum_{\boldsymbol{\alpha}\in\Lambda(f)} c_{\boldsymbol{\alpha}}\mathbf{x}^{\boldsymbol{\alpha}} - \sum_{\boldsymbol{\beta}\in\Gamma(f)} d_{\boldsymbol{\beta}}\mathbf{x}^{\boldsymbol{\beta}} \in \mathbb{R}[\mathbf{x}]$. If $\tilde{f}$ is nonnegative, then f is nonnegative. Moreover, $\tilde{f} \in$ SONC if and only if $f \in$ SONC.*

Proof. For any $\mathbf{x} \in \mathbb{R}^n$, we have

$$\begin{aligned} f(\mathbf{x}) &= \sum_{\boldsymbol{\alpha}\in\Lambda(f)} c_{\boldsymbol{\alpha}}\mathbf{x}^{\boldsymbol{\alpha}} - \sum_{\boldsymbol{\beta}\in\Gamma(f)} d_{\boldsymbol{\beta}}\mathbf{x}^{\boldsymbol{\beta}} \\ &\geq \sum_{\boldsymbol{\alpha}\in\Lambda(f)} c_{\boldsymbol{\alpha}}|\mathbf{x}|^{\boldsymbol{\alpha}} - \sum_{\boldsymbol{\beta}\in\Gamma(f)} |d_{\boldsymbol{\beta}}||\mathbf{x}|^{\boldsymbol{\beta}} = \tilde{f}(|\mathbf{x}|), \end{aligned}$$

where $|\mathbf{x}| = (|x_1|, \ldots, |x_n|)$. It follows that the nonnegativity of $\tilde{f}$ implies the nonnegativity of f.

For every $\boldsymbol{\beta} \in \Gamma(f)$, let $\mathscr{F}(\boldsymbol{\beta})$ be as in (1). Let $\mathscr{B} = \{\boldsymbol{\beta} \in \Gamma(f) \mid \boldsymbol{\beta} \notin (2\mathbb{N})^n \text{ and } d_{\boldsymbol{\beta}} < 0\}$ and $\tilde{\mathscr{A}} = \{\boldsymbol{\alpha} \in \Lambda(f) \mid \boldsymbol{\alpha} \notin \cup_{\boldsymbol{\beta}\in\Gamma(f)} \cup_{\Delta\in\mathscr{F}(\boldsymbol{\beta})} V(\Delta)\}$. Assume $\tilde{f} \in$ SONC. Then we can write

$$\begin{aligned} \tilde{f} = &\sum_{\boldsymbol{\beta}\in\Gamma(f)\setminus\mathscr{B}} \sum_{\Delta\in\mathscr{F}(\boldsymbol{\beta})} (\sum_{\boldsymbol{\alpha}\in V(\Delta)} c_{\boldsymbol{\beta}\Delta\boldsymbol{\alpha}}\mathbf{x}^{\boldsymbol{\alpha}} - d_{\boldsymbol{\beta}\Delta}\mathbf{x}^{\boldsymbol{\beta}}) \\ &+ \sum_{\boldsymbol{\beta}\in\mathscr{B}} \sum_{\Delta\in\mathscr{F}(\boldsymbol{\beta})} (\sum_{\boldsymbol{\alpha}\in V(\Delta)} c_{\boldsymbol{\beta}\Delta\boldsymbol{\alpha}}\mathbf{x}^{\boldsymbol{\alpha}} - \tilde{d}_{\boldsymbol{\beta}\Delta}\mathbf{x}^{\boldsymbol{\beta}}) + \sum_{\boldsymbol{\alpha}\in\tilde{\mathscr{A}}} c_{\boldsymbol{\alpha}}\mathbf{x}^{\boldsymbol{\alpha}} \end{aligned}$$

s.t. each $\sum_{\boldsymbol{\alpha}\in V(\Delta)} c_{\boldsymbol{\beta}\Delta\boldsymbol{\alpha}}\mathbf{x}^{\boldsymbol{\alpha}} - d_{\boldsymbol{\beta}\Delta}\mathbf{x}^{\boldsymbol{\beta}}$ and each $\sum_{\boldsymbol{\alpha}\in V(\Delta)} c_{\boldsymbol{\beta}\Delta\boldsymbol{\alpha}}\mathbf{x}^{\boldsymbol{\alpha}} - \tilde{d}_{\boldsymbol{\beta}\Delta}\mathbf{x}^{\boldsymbol{\beta}}$ are nonnegative circuit polynomials. Note that $\sum_{\boldsymbol{\alpha}\in V(\Delta)} c_{\boldsymbol{\beta}\Delta\boldsymbol{\alpha}}\mathbf{x}^{\boldsymbol{\alpha}} + \tilde{d}_{\boldsymbol{\beta}\Delta}\mathbf{x}^{\boldsymbol{\beta}}$ is also a nonnegative circuit polynomial and $\sum_{\Delta\in\Delta(\boldsymbol{\beta})} \tilde{d}_{\boldsymbol{\beta}\Delta} = |d_{\boldsymbol{\beta}}| = -d_{\boldsymbol{\beta}}$ for any $\boldsymbol{\beta} \in \mathscr{B}$. Hence,

$$\begin{aligned} f = &\sum_{\boldsymbol{\beta}\in\Gamma(f)\setminus\mathscr{B}} \sum_{\Delta\in\mathscr{F}(\boldsymbol{\beta})} (\sum_{\boldsymbol{\alpha}\in V(\Delta)} c_{\boldsymbol{\beta}\Delta\boldsymbol{\alpha}}\mathbf{x}^{\boldsymbol{\alpha}} - d_{\boldsymbol{\beta}\Delta}\mathbf{x}^{\boldsymbol{\beta}}) \\ &+ \sum_{\boldsymbol{\beta}\in\mathscr{B}} \sum_{\Delta\in\mathscr{F}(\boldsymbol{\beta})} (\sum_{\boldsymbol{\alpha}\in V(\Delta)} c_{\boldsymbol{\beta}\Delta\boldsymbol{\alpha}}\mathbf{x}^{\boldsymbol{\alpha}} + \tilde{d}_{\boldsymbol{\beta}\Delta}\mathbf{x}^{\boldsymbol{\beta}}) + \sum_{\boldsymbol{\alpha}\in\tilde{\mathscr{A}}} c_{\boldsymbol{\alpha}}\mathbf{x}^{\boldsymbol{\alpha}} \in \text{SONC}. \end{aligned}$$

The inverse follows similarly. □

Hence by Lemma 5.5, if we replace the polynomial f in (4) by its associated PN-polynomial $\tilde{f}$, then this does not affect the optimal value of (4):

$$\text{(SONC-PN)}: \quad \sup\{\xi : \tilde{f}(\mathbf{x}) - \xi \in \text{SONC}\}. \tag{5}$$

5.2 Compute a simplex cover

Given a polynomial $f = \sum_{\boldsymbol{\alpha}\in\Lambda(f)} c_{\boldsymbol{\alpha}}\mathbf{x}^{\boldsymbol{\alpha}} - \sum_{\boldsymbol{\beta}\in\Gamma(f)} d_{\boldsymbol{\beta}}\mathbf{x}^{\boldsymbol{\beta}} \in \mathbb{R}[\mathbf{x}]$, in order to obtain a SONC decomposition of f, we use all simplices containing $\boldsymbol{\beta}$ for each $\boldsymbol{\beta} \in \Gamma(f)$ in Theorem 3.9. In practice, we do not need that many simplices. A recent study [21] proposes a systematic method to compute an optimal simplex cover. It would be worth trying to combine this framework with our SOC characterization for SONC cones to achieve a more accurate algorithm. Here we rely on a heuristics to compute a set of simplices with vertices coming from $\Lambda(f)$ and that covers $\Gamma(f)$. For $\boldsymbol{\beta} \in \Gamma(f)$ and $\boldsymbol{\alpha}_0 \in \Lambda(f)$, define an auxiliary linear program:

$$\begin{aligned} &\text{SimSel}(\boldsymbol{\beta}, \Lambda(f), \boldsymbol{\alpha}_0) = \text{Argmax} \quad \lambda_{\boldsymbol{\alpha}_0} \\ &\text{s.t.}\{\sum_{\boldsymbol{\alpha}\in\Lambda(f)} \lambda_{\boldsymbol{\alpha}} \cdot \boldsymbol{\alpha} = \boldsymbol{\beta}, \sum_{\boldsymbol{\alpha}\in\Lambda(f)} \lambda_{\boldsymbol{\alpha}} = 1, \lambda_{\boldsymbol{\alpha}} \geq 0, \forall \boldsymbol{\alpha} \in \Lambda(f)\}. \end{aligned}$$

Following [25], we can ensure the output of $\text{SimSel}(\boldsymbol{\beta}, \Lambda(f), \boldsymbol{\alpha}_0)$ corresponds to a trellis which contains $\boldsymbol{\alpha}_0$ and covers $\boldsymbol{\beta}$. The so-called `SimplexCover`[1] procedure computes such a simplex cover.

Let $\mathbf{K}$ be the 3-dimensional rotated SOC, i.e.,

$$\mathbf{K} := \{(a,b,c) \in \mathbb{R}^3 \mid 2ab \geq c^2, a \geq 0, b \geq 0\}. \tag{6}$$

Suppose $\tilde{f} = \sum_{\boldsymbol{\alpha}\in\Lambda(f)} c_{\boldsymbol{\alpha}}\mathbf{x}^{\boldsymbol{\alpha}} - \sum_{\boldsymbol{\beta}\in\Gamma(f)} d_{\boldsymbol{\beta}}\mathbf{x}^{\boldsymbol{\beta}} \in \mathbb{R}[\mathbf{x}]$. By algorithm `SimplexCover`, we compute a simplex cover $\{(\mathscr{A}_k, \boldsymbol{\beta}_k)\}_{k=1}^l$. For each k, let M_k be an $\mathscr{A}_k$-rational mediated set containing $\boldsymbol{\beta}_k$ and $s_k = \#M_k \setminus \mathscr{A}_k$. For each $\boldsymbol{u}_i^k \in M_k \setminus \mathscr{A}_k$, let us write $\boldsymbol{u}_i^k = \frac{1}{2}(\boldsymbol{v}_i^k + \boldsymbol{w}_i^k)$. Let $\tilde{\mathscr{A}} = \{\boldsymbol{\alpha} \in \Lambda(f) \mid \boldsymbol{\alpha} \notin \cup_{\boldsymbol{\beta}\in\Gamma(f)} \cup_{\Delta\in\mathscr{F}(\boldsymbol{\beta})} V(\Delta)\}$. Then we can relax (SONC-PN) to an SOCP problem (SONC-SOCP) as follows:

$$\begin{cases} \sup & \xi \\ \text{s.t.} & \tilde{f}(\mathbf{x}) - \xi = \sum_{k=1}^l \sum_{i=1}^{s_k} (2a_i^k\mathbf{x}^{\boldsymbol{v}_i^k} + b_i^k\mathbf{x}^{\boldsymbol{w}_i^k} - 2c_i^k\mathbf{x}^{\boldsymbol{u}_i^k}) + \sum_{\boldsymbol{\alpha}\in\tilde{\mathscr{A}}} c_{\boldsymbol{\alpha}}\mathbf{x}^{\boldsymbol{\alpha}}, \\ & (a_i^k, b_i^k, c_i^k) \in \mathbf{K}, \quad \forall i, k. \end{cases} \tag{7}$$

Let us denote by ξ_{socp} the optimal value of (7). Then, we have $\xi_{socp} \leq \xi_{sonc} \leq \xi^*$.

Remark 5.6. *The quality of obtained SONC lower bounds depends on two successive steps: the relaxation to the corresponding PN-polynomial (from ξ^* to ξ_{sonc}) and the relaxation to a specific simplex cover (from ξ_{sonc} to ξ_{socp}). The loss of bound-quality at the*

[1] Algorithm 4 in https://arxiv.org/abs/1906.06179

second step can be improved by choosing a more optimal simplex cover. Nevertheless, it may happen that the loss of bound-quality at the first step is already big, as shown in Example 5.7, which indicates that the gap between nonnegative polynomials and SONC PN-polynomials may greatly affect the quality of SONC lower bounds.

Example 5.7. Let $f = 1 + x_1^4 + x_2^4 - x_1x_2^2 - x_1^2x_2 + 5x_1x_2$. Since $\Lambda(f)$ forms a trellis, the simplex cover for f is unique. One obtains $\xi_{socp} = \xi_{sonc} \approx -6.916501$ while $\xi^* \approx -2.203372$. Hence the relative optimality gap is near 214%.

6 NUMERICAL EXPERIMENTS

Here, we present numerical results of the proposed algorithms for unconstrained POPs. Our tool, called SONCSOCP, implements the simplex cover algorithm as well as a procedure MedSet[2] computing the rational mediated set and computes the optimal value ξ_{socp} of the SOCP (7) with Mosek [3]. All experiments were performed on an Intel Core i5-8265U@1.60GHz CPU with 8GB RAM memory and WINDOWS 10 system. SONCSOCP is available at github:SONCSOCP.

Our benchmarks are issued from the database of randomly generated polynomials provided by Seidler and de Wolff in [25]. Depending on the Newton polytope, these benchmarks are divided into three classes: the ones with standard simplices, the ones with general simplices and the ones with arbitrary Newton polytopes. (We use n, d, t, l to denote the number of variables, the degree, the number of terms and the lower bound on the number of inner terms respectively. See [25] for the details on the construction of these polynomials). We compare the performance of SONCSOCP with the ones of POEM, which relies on the ECOS solver to solve geometric programs (see [25] for more details). To measure the quality of a given lower bound ξ_{lb}, we rely on the 'local_min' function available in POEM which computes an upper bound ξ_{min} on the minimum of a polynomial. The relative optimality gap is defined by $\frac{|\xi_{min}-\xi_{lb}|}{|\xi_{min}|}$. In the following tables, the column 'time' is the running time in seconds and the column 'opt' the optimal value.

Standard simplex. For the standard simplex case, we take 10 polynomials of different types (labeled by N). Running time and lower bounds obtained with SONCSOCP and POEM are displayed in Table 1. Note that for polynomials with $\Lambda(\cdot)$ forming a trellis, the simplex cover is unique, thus the bounds obtained by SONCSOCP and POEM are the same theoretically, which is also reflected in Table 1. For each polynomial, the relative optimality gap is less than 1% and for 8 out of 10 polynomials, it is less than 0.1% (see Figure 2).

	N	1	2	3	4	5	6	7	8	9	10
	n	10	10	10	20	20	20	30	30	40	40
	d	40	50	60	40	50	60	50	60	50	60
	t	20	20	20	30	30	30	50	50	100	100
time	SONCSOCP	0.04	0.04	0.04	0.14	0.14	0.13	0.43	0.40	2.23	2.21
	POEM	0.26	0.27	0.26	0.43	0.44	0.42	1.78	1.79	2.20	2.25
opt	SONCSOCP	3.52	3.52	3.52	2.64	2.64	2.64	2.94	2.94	4.41	4.41
	POEM	3.52	3.52	3.52	2.64	2.64	2.64	2.94	2.94	4.41	4.41

Table 1: Results for the standard simplex case

General simplex. Here, we take 10 polynomials of different types (labeled by N). Running time and lower bounds obtained with SONCSOCP and POEM are displayed in Table 2. As before, the SONC lower bounds obtained by SONCSOCP and POEM are the same.

[2] Algorithm 3 in https://arxiv.org/abs/1906.06179

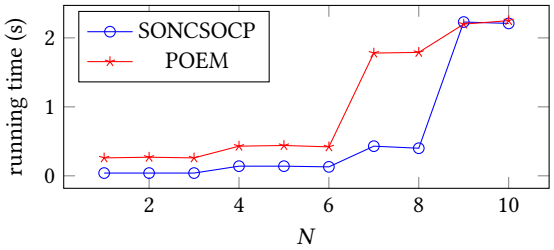


Figure 1: Running time for the standard simplex case

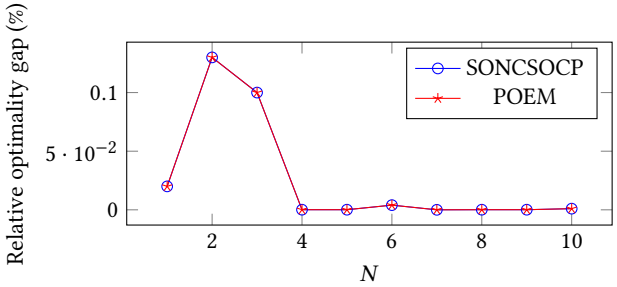


Figure 2: Relative optimality gap for the standard simplex case

For each polynomial except for the one corresponding to $N = 7$, the relative optimality gap is within 30%, and for 6 out of 10 polynomials, the gap is below 1% (see Figure 4). POEM fails to obtain a lower bound for the instance $N = 10$ by returning −Inf. Figure 3 shows that, overall, the running times of SONCSOCP and POEM are close. SONCSOCP is faster than POEM for the instance $N = 6$, possibly because better performance are obtained when the degree is relatively low.

	N	1	2	3	4	5	6	7	8	9	10
	n	10	10	10	10	10	10	10	10	10	10
	d	20	30	40	50	60	20	30	40	50	60
	t	20	20	20	20	20	30	30	30	30	30
time	SONCSOCP	0.32	0.29	0.36	0.48	0.54	0.56	0.73	0.88	1.04	1.04
	POEM	0.28	0.31	0.31	0.31	0.43	0.74	0.75	0.74	0.72	0.76
opt	SONCSOCP	1.18	0.22	0.38	0.90	0.06	4.00	−4.64	1.62	2.95	5.40
	POEM	1.18	0.22	0.38	0.90	0.06	4.00	−4.64	1.62	2.95	−Inf

Table 2: Results for the general simplex case

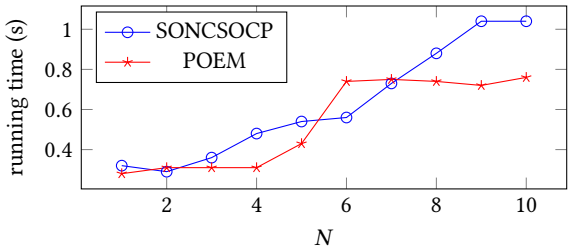


Figure 3: Running time for the general simplex case

Arbitrary polytope. We take 20 polynomials of different types (labeled by N). POEM always throws an error "expected square matrix". Running time and lower bounds obtained with SONCSOCP are displayed in Table 3. The relative optimality gap is always within 25% and within 1% for 17 out of 20 polynomials (see Figure 5).

7 CONCLUSIONS

In this paper, we provide a constructive proof that each SONC cone admits an SOC representation. Based on this, we propose an

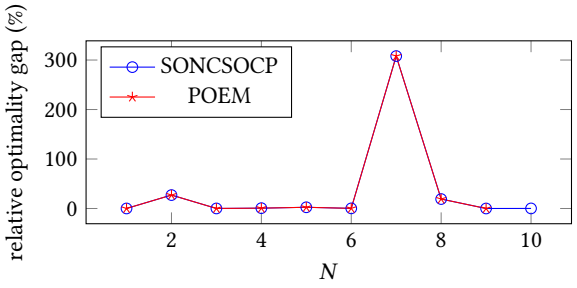


Figure 4: Relative optimality gap for the general simplex case

N		1	2	3	4	5	6	7	8	9	10
n		10	10	10	10	10	10	10	10	10	10
d		20	20	20	30	30	30	40	40	40	50
t		30	100	300	30	100	300	30	100	300	30
l		15	71	231	15	71	231	15	71	231	15
SONCSOCP	time	0.38	1.75	6.86	0.64	3.13	11.3	0.72	4.01	14.6	0.76
	opt	0.70	3.32	31.7	3.31	15.3	3.31	0.47	5.42	38.7	1.56
N		11	12	13	14	15	16	17	18	19	20
n		10	10	10	10	10	20	20	20	20	20
d		50	50	60	60	60	30	30	40	40	40
t		100	300	30	100	300	50	100	50	100	200
l		71	231	15	71	231	5	15	5	15	35
SONCSOCP	time	4.41	16.8	1.84	11.2	42.4	3.20	8.84	2.60	10.5	38.7
	opt	0.20	7.00	3.31	2.52	23.4	0.70	4.91	4.13	2.81	9.97

Table 3: Results for the arbitrary polytope case

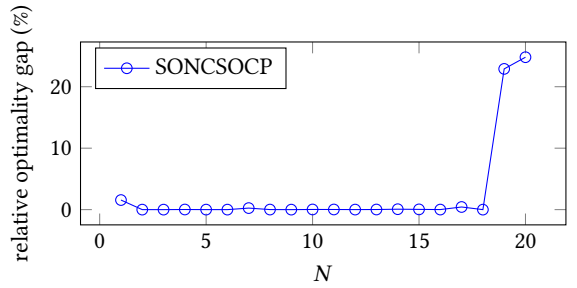


Figure 5: Relative optimality gap for the arbitrary polytope case

algorithm to compute a lower bound for unconstrained POPs via SOCP. Numerical experiments demonstrate the efficiency of our algorithm even when the number of variables and the degree are fairly large. Even though the complexity of our algorithm depends on the degree in theory, it turns out that this dependency is rather mild. For all numerical examples tested in this paper, the running time is below one minute even for polynomials of degree up to 60. Since the running time is satisfactory, the main concern of SONC-based algorithms for sparse polynomial optimization may be the quality of obtained lower bounds. For many examples tested in this paper, the relative optimality gap is within 1%. However, it can happen that the SONC lower bound is not accurate and this cannot be avoided by choosing an optimal simplex cover. To improve the quality of such bounds, it is mandatory to find more complex representations of nonnegative polynomials, which involve SONC polynomials. We also plan to design a rounding-projection procedure, in the spirit of [22], to obtain exact nonnegativity certificates for polynomials lying in the interior of the SONC cone. A related investigation track is the complexity analysis and software implementation of the resulting hybrid numeric-symbolic scheme, as well as performance comparisons with concurrent methods based on semidefinite programming [16] or geometric programming [19].

REFERENCES

[1] Amir Ali Ahmadi and Anirudha Majumdar. 2019. DSOS and SDSOS optimization: more tractable alternatives to sum of squares and semidefinite optimization. *SIAM Journal on Applied Algebra and Geometry* 3, 2 (2019), 193–230.

[2] Farid Alizadeh and Donald Goldfarb. 2003. Second-order cone programming. *Mathematical programming* 95, 1 (2003), 3–51.

[3] E. D. Andersen and K. D. Andersen. 2000. The Mosek Interior Point Optimizer for Linear Programming: An Implementation of the Homogeneous Algorithm. In *High Performance Optimization*, Hans Frenk, Kees Roos, Tamás Terlaky, and Shuzhong Zhang (Eds.). Applied Optimization, Vol. 33. Springer US, 197–232. https://doi.org/10.1007/978-1-4757-3216-0_8

[4] Gennadiy Averkov. 2019. Optimal size of linear matrix inequalities in semidefinite approaches to polynomial optimization. *SIAM Journal on Applied Algebra and Geometry* 3, 1 (2019), 128–151.

[5] Ahron Ben-Tal and Arkadi Nemirovski. 2001. *Lectures on modern convex optimization: analysis, algorithms, and engineering applications*. Vol. 2. Siam.

[6] V. Chandrasekaran and P. Shah. 2016. Relative Entropy Relaxations for Signomial Optimization. *SIAM J. Optim.* 26, 2 (2016), 1147–1173.

[7] Mareike Dressler, Sadik Iliman, and Timo De Wolff. 2019. An approach to constrained polynomial optimization via nonnegative circuit polynomials and geometric programming. *Journal of Symbolic Computation* 91 (2019), 149–172.

[8] Hamza Fawzi. 2019. On representing the positive semidefinite cone using the second-order cone. *Mathematical Programming* 175, 1-2 (2019), 109–118.

[9] Jacob Hartzer, Olivia Röhrig, Timo de Wolff, and Oğuzhan Yürük. 2019. Initial Steps in the Classification of Maximal Mediated Sets. *arXiv preprint arXiv:1910.00502* (2019).

[10] Sadik Iliman and Timo De Wolff. 2016. Amoebas, nonnegative polynomials and sums of squares supported on circuits. *Research in the Mathematical Sciences* 3, 1 (2016), 9.

[11] Sadik Iliman and Timo De Wolff. 2016. Lower bounds for polynomials with simplex newton polytopes based on geometric programming. *SIAM Journal on Optimization* 26, 2 (2016), 1128–1146.

[12] C. Josz. 2016. *Application of polynomial optimization to electricity transmission networks*. Theses. Université Pierre et Marie Curie - Paris VI. https://tel.archives-ouvertes.fr/tel-01478431

[13] L. Katthan, H. Naumann, and T. Theobald. 2019. A Unified framework of SAGE and SONC polynomials and its duality theory. *arXiv preprint arXiv:1903.08966* (2019).

[14] I. Klep, V. Magron, and J. Povh. 2019. Sparse Noncommutative Polynomial Optimization. *arXiv preprint arXiv:1909.00569* (2019).

[15] V. Magron, G. Constantinides, and A. Donaldson. 2017. Certified Roundoff Error Bounds Using Semidefinite Programming. *ACM Trans. Math. Softw.* 43, 4, Article 34 (2017), 34 pages.

[16] V. Magron and M. Safey El Din. 2018. On Exact Polya and Putinar's Representations. In *ISSAC'18: Proceedings of the 2018 ACM International Symposium on Symbolic and Algebraic Computation*. ACM, New York, NY, USA.

[17] V. Magron and M. Safey El Din. 2018. RealCertify: a Maple package for certifying non-negativity. In *ISSAC'18: Proceedings of the 2018 ACM International Symposium on Symbolic and Algebraic Computation*. ACM, New York, NY, USA.

[18] Victor Magron, Mohab Safey El Din, and Markus Schweighofer. 2019. Algorithms for weighted sum of squares decomposition of non-negative univariate polynomials. *Journal of Symbolic Computation* 93 (2019), 200–220.

[19] Victor Magron, Henning Seidler, and Timo de Wolff. 2019. Exact Optimization via Sums of Nonnegative Circuits and Arithmetic-Geometric-Mean-Exponentials. In *Proceedings of the 2019 on International Symposium on Symbolic and Algebraic Computation* (Beijing, China) *(ISSAC '19)*. New York, NY, USA, 291–298.

[20] Riley Murray, Venkat Chandrasekaran, and Adam Wierman. 2018. Newton polytopes and relative entropy optimization. *arXiv preprint arXiv:1810.01614* (2018).

[21] Dávid Papp. 2019. Duality of sum of nonnegative circuit polynomials and optimal SONC bounds. *arXiv preprint arXiv:1912.04718* (2019).

[22] H. Peyrl and P.A. Parrilo. 2008. Computing sum of squares decompositions with rational coefficients. *Theoretical Computer Science* 409, 2 (2008), 269–281.

[23] Victoria Powers and Bruce Reznick. 2019. A note on mediated simplices. *arXiv preprint arXiv:1909.11008* (2019).

[24] Bruce Reznick. 1989. Forms derived from the arithmetic-geometric inequality. *Math. Ann.* 283, 3 (1989), 431–464.

[25] Henning Seidler and Timo de Wolff. 2018. An experimental comparison of sonc and sos certificates for unconstrained optimization. *arXiv preprint arXiv:1808.08431* (2018).

[26] H. Waki, S. Kim, M. Kojima, and M. Muramatsu. 2006. Sums of Squares and Semidefinite Programming Relaxations for Polynomial Optimization Problems with Structured Sparsity. *SIAM Journal on Optimization* 17, 1 (2006), 218–242.

[27] J. Wang. 2018. Nonnegative polynomials and circuit polynomials. *arXiv preprint arXiv:1804.09455* (2018).

[28] J. Wang, V. Magron, and J.-B. Lasserre. 2019. TSSOS: a moment-SOS hierarchy that exploits term sparsity. *arXiv preprint arXiv:1912.08899* (2019).

Geometric Modeling and Regularization of Algebraic Problems

Zhonggang Zeng*
zzeng@neiu.edu
Northeastern Illinois University
Chicago, Illinois, United States

ABSTRACT

Discontinuity with respect to data perturbations is common in algebraic computation where solutions are often highly sensitive. Such problems can be modeled as solving systems of equations at given data parameters. By appending auxiliary equations, the models can be formulated to satisfy four easily verifiable conditions so that the data form complex analytic manifolds on which the solutions maintain their structures and the Lipschitz continuity. When such a problem is given with empirical data, solving the system becomes a least squares problem whose solution uniquely exists and enjoys Lipschitz continuity as long as the data point is in a tubular neighborhood of the manifold. As a result, the singular problem is regularized as a well-posed computational problem.

CCS CONCEPTS

• **Mathematics of computing → Nonlinear equations**; **Computations on matrices**; • **Theory of computation → Numeric approximation algorithms**.

KEYWORDS

system of equations, regularization, complex analytic manifold

ACM Reference Format:
Zhonggang Zeng. 2020. Geometric Modeling and Regularization of Algebraic Problems. In *International Symposium on Symbolic and Algebraic Computation (ISSAC '20), July 20–23, 2020, Kalamata, Greece.* ACM, New York, NY, USA, 8 pages. https://doi.org/10.1145/3373207.3404066

1 INTRODUCTION

Computational problems with extremely high sensitivities beyond finite bounds are known to be *ill-posed*. Such problems are abundant in algebraic computation and also referred to as being *singular*. Some of the most basic algebraic problems are ill-posed, such as matrix ranks and subspaces, solutions of singular linear systems, polynomial greatest common divisors and factorizations, defective eigenvalues and Jordan Canonical Forms. Those are the problems we inevitably encounter in symbolic, numeric and hybrid computation. Based on the current state of knowledge, however, it is inaccurately believed by many that such problems are impossible to solve from empirical data or using floating point arithmetic. Pessimistic outlooks are abundant in the literature (emphasis added): "The moral is to *avoid floating point solutions* of singular systems" [17, page 218]. "The difficulty is that the JCF *cannot be computed* using floating point arithmetic. A single rounding error may cause some multiple eigenvalue to become distinct or vice versa, altering the entire structure" [18]. "A *dramatic deterioration of the accuracy* must therefore be expected"[22, page 300]. "[S]mall variations in the [data] will result in large variations in the [solution]. *There is no hope of computing such an object* in a stable way"[14, page. 128]. "[Although such an object] is of fundamental theoretical importance *it is of little use* in practical computations, being generally very difficult to compute"[3, page 52]. "[So] that [it] is little used in numerical applications"[14, page. 128].

Are the solutions of those problems really sensitive to data perturbations as alleged? In a legendary technical report [15], Kahan argues that it is a "misconception" to consider multiple roots of polynomials hypersensitive, points out that polynomials and matrices form heuristic "pejorative manifolds" preserving root multiplicities and Jordan structures respectively, and proves that the sensitivities of roots and eigenvalues are bounded if the perturbation is constrained to preserve the multiplicity. This insight opens a possible pathway for accurate solution of such singular problems.

In this paper, we establish conditions for modeling an algebraic problem as a nonlinear system of equations in the form of solving $\mathbf{f}(\mathbf{u}, \mathbf{v}) = \mathbf{0}$ for the variable $\mathbf{v}$ at a fixed data value $\mathbf{u}$ so that we can rigorously verify that the data form a complex analytic manifold on which the solution maintains a certain algebraic structure and enjoys Lipschitz continuity.

The data of a hypersensitive problem forming smooth manifolds is crucial in the analysis and regularization the problems since its solution is of bounded sensitivity with respect to data on the manifold. We further extend this inherent stability beyond the manifold into its *tubular neighborhood*. When the problem data are given as empirical, we have a data point near the manifold in the data space. Assuming the data are reasonably accurate so that the point remains in the tubular neighborhood, the Tubular Neighborhood Theorem established in this paper ensures the projection from the data point to the manifold uniquely exists and enjoys Lipschitz continuity. Consequently, the singular problem can be regularized as a well-posed least squares problem that is accurately solvable from empirical data.

The geometric modeling and regularization from this perspective lead to robust algorithms such as those in accurate computation of multiple roots [24], greatest common divisors [16, 26], polynomial factorizations [23, 25], defective eigenvalue problems [27] and singular linear systems [29]. These algorithms are implemented in our software package NAClab [31].

*Research is supported in part by NSF under grant DMS-1620337.

ISSAC '20, July 20–23, 2020, Kalamata, Greece

ACM ISBN 978-1-4503-7100-1/20/07…$15.00
https://doi.org/10.1145/3373207.3404066

Geometric theories and methods have been applied in algebraic computing in many works such as [1, 5–7, 9–11, 19]. However, the tremendous advantage of tubular neighborhoods has not yet been harnessed partly because a general tubular neighborhood theorem for complex analytic manifolds is apparently unavailable in the literature of differential topology. Specifically taylored to the application of solving ill-posed algebraic problems in this paper, we prove a weak but sufficient version of the tubular neighborhood theorem for complex analytic manifolds in Euclidean spaces using the techniques of nonlinear least squares. The theorem and the proof fills a gap in the regularization theory of solving ill-posed algebraic problems and complete the works of numerical factorization [23, 25] and numerical greatest common divisors of polynomials [26, 30].

2 PRELIMINARIES

The space of n-dimensional vectors of complex numbers is denoted by $\mathbb{C}^n$ with the Euclidean norm $\|\cdot\|_2$. General vector spaces are denoted by, say $\mathcal{V}, \mathcal{W}$ in which vectors are denoted by boldface lower case letters while $\mathbf{0}$ is a zero vector. Any norm $\|\mathbf{v}\|$ is understood as the specified norm in the space where $\mathbf{v}$ belongs.

For a holomorphic mapping $F : \Omega \subset \mathbb{C}^n \to \mathbb{C}^m$, we may designate a variable name, say $\mathbf{z}$, for F and denote F as $\mathbf{z} \mapsto F(\mathbf{z})$. The Jacobian matrix of F at any $\mathbf{z}_0 \in \Omega$ is denoted by $F_{\mathbf{z}}(\mathbf{z}_0)$. Let $\mathcal{V}$ and $\mathcal{W}$ be vector spaces with isomorphisms $\phi : \mathcal{V} \to \mathbb{C}^n$ and $\psi : \mathcal{W} \to \mathbb{C}^m$. Assume $\mathbf{g}$ is a mapping from an open subset Σ of $\mathcal{V}$ to $\mathcal{W}$ with a representation $\mathbf{z} \mapsto G(\mathbf{z})$ where $G : \phi(\Sigma) \subset \mathbb{C}^n \to \mathbb{C}^m$ such that $\mathbf{g} = \psi^{-1} \circ G \circ \phi$. We say $\mathbf{g}$ is holomorphic in Σ if its representation G is holomorphic in $\phi(\Sigma)$. Denoting the variable of $\mathbf{g}$ as, say $\mathbf{v}$, the *Jacobian* of $\mathbf{g}$ at any particular $\mathbf{v}_0 \in \Sigma$ is defined as the linear transformation $\mathbf{g}_{\mathbf{v}}(\mathbf{v}_0) : \mathcal{V} \to \mathcal{W}$ in the form of

$$\mathbf{v} \longmapsto \mathbf{g}_{\mathbf{v}}(\mathbf{v}_0)(\mathbf{v}) := \psi^{-1}\big(G_{\mathbf{z}}(\phi(\mathbf{v}_0))\,\phi(\mathbf{v})\big).$$

The Jacobian $\mathbf{g}_{\mathbf{v}}(\mathbf{v}_0)$ as a linear transformation is invariant under change of bases. Let $G_{\mathbf{z}}(\mathbf{z}_0)^{\mathsf{H}}$ and $G_{\mathbf{z}}(\mathbf{z}_0)^{\dagger}$ be the Hermitian transpose and the Moore-Penrose inverse of the Jacobian matrix $G_{\mathbf{z}}(\mathbf{z}_0)$ respectively where $\mathbf{z}_0 = \phi(\mathbf{v}_0)$. If we further assume the isomorphisms ϕ and ψ are isometric in the sense that $\|\phi(\mathbf{v})\|_2 = \|\mathbf{v}\|$ and $\|\psi(\mathbf{w})\|_2 = \|\mathbf{w}\|$ for all $\mathbf{v} \in \mathcal{V}$ and $\mathbf{w} \in \mathcal{W}$, then $\mathbf{g}_{\mathbf{v}}(\mathbf{v}_0)^{\mathsf{H}}$ and $\mathbf{g}_{\mathbf{v}}(\mathbf{v}_0)^{\dagger}$ are well-defined as $\mathbf{g}_{\mathbf{v}}(\mathbf{v}_0)^{\mathsf{H}} = \phi^{-1} \circ G_{\mathbf{z}}(\mathbf{z}_0)^{\mathsf{H}} \circ \psi$ and $\mathbf{g}_{\mathbf{v}}(\mathbf{v}_0)^{\dagger} = \phi^{-1} \circ G_{\mathbf{z}}(\mathbf{z}_0)^{\dagger} \circ \psi$ that are invariant under isometric isomorphisms. A mapping $\mathbf{f}$ is holomorphic in a non-open domain $\Pi \subset \mathcal{V}$ if there is an open subset Ω of $\mathcal{V}$ containing Π and a holomorphic mapping $\mathbf{g}$ defined in Ω such that $\mathbf{f}(\mathbf{z}) \equiv \mathbf{g}(\mathbf{z})$ for all $\mathbf{z} \in \Pi$. For a holomorphic mapping $(\mathbf{u}, \mathbf{v}) \mapsto \mathbf{f}(\mathbf{u}, \mathbf{v})$, its Jacobian at $(\mathbf{u}_0, \mathbf{v}_0)$ is denoted by $\mathbf{f}_{\mathbf{uv}}(\mathbf{u}_0, \mathbf{v}_0)$ and its partial Jacobian with respect to, say $\mathbf{v}$, is denoted by $\mathbf{f}_{\mathbf{v}}(\mathbf{u}_0, \mathbf{v}_0)$.

3 COMPLEX ANALYTIC MANIFOLDS

For our applications, we consider complex analytic manifolds in normed vector spaces in the following definition.

Definition 3.1 (Complex Analytic Manifold). Let $\mathcal{U}$ be a finite-dimensional normed vector space over $\mathbb{C}$. A subset Π of $\mathcal{U}$ is a *complex analytic manifold* of dimension m if there is an m-dimensional normed vector space $\mathcal{V}$ over $\mathbb{C}$ and, for every $\mathbf{u} \in \Pi$, there is an open neighborhood Σ of $\mathbf{u}$ in $\mathcal{U}$ and a holomorphic mapping ϕ from $\Sigma \cap \Pi$ onto an open subset Λ of $\mathcal{V}$ with a holomorphic inverse. The dimension deficit $\dim(\mathcal{U}) - m$ is called the *codimension* of Π in $\mathcal{U}$ denoted by $\operatorname{codim}(\Pi)$.

The term *manifold* in this paper refers to a complex analytic manifold in the sense of Definition 3.1. As we shall elaborate in case studies in §4, algebraic problems whose solutions possess certain algebraic structures can often be modeled as a system of nonlinear equations in the form of *solving* $\mathbf{f}(\mathbf{u}, \mathbf{v}) = \mathbf{0}$ *for the variable* $\mathbf{v}$ *at the given data parameter value* $\mathbf{u}$. The following theorem establishes four basic conditions for such a model so that the data points form a manifold. The theorem simplifies the tedious process of establishing a manifold to verifying the four conditions.

Theorem 3.2 (Geometric Modeling Theorem). *A subset Π is a complex analytic manifold in a normed vector space $\mathcal{U}$ over $\mathbb{C}$ if and only if there are normed vector spaces $\mathcal{V}$ and $\mathcal{W}$ over $\mathbb{C}$ with $\dim(\mathcal{V}) \le \dim(\mathcal{W}) \le \dim(\mathcal{U}) + \dim(\mathcal{V}) < \infty$ such that, at every $\mathbf{u}_0 \subset \Pi$, there is a holomorphic mapping $(\mathbf{u}, \mathbf{v}) \mapsto \mathbf{f}(\mathbf{u}, \mathbf{v})$ from an open domain $\Omega \subset \mathcal{U} \times \mathcal{V}$ to $\mathcal{W}$ with the properties below:*

(i) *There is a $\mathbf{v}_0 \in \mathcal{V}$ such that $\mathbf{f}(\mathbf{u}_0, \mathbf{v}_0) = \mathbf{0}$.*
(ii) *$\mathbf{f}_{\mathbf{uv}}(\mathbf{u}_0, \mathbf{v}_0)$ is surjective and $\mathbf{f}_{\mathbf{v}}(\mathbf{u}_0, \mathbf{v}_0)$ is injective.*
(iii) *$\mathbf{f}(\mathbf{u}, \mathbf{v}) = \mathbf{0}$ implies $\mathbf{u} \in \Pi$.*
(iv) *For every open neighborhood Δ of $\mathbf{v}_0$ in $\mathcal{V}$, there is an open neighborhood Σ of $\mathbf{u}_0$ in $\mathcal{U}$ such that every $\mathbf{u} \in \Sigma \cap \Pi$ corresponds to a unique $\mathbf{v} \in \Delta$ with $\mathbf{f}(\mathbf{u}, \mathbf{v}) = \mathbf{0}$.*

Under these conditions, we have $\operatorname{codim}(\Pi) = \dim(\mathcal{W}) - \dim(\mathcal{V})$.

Proof. Let Π be a manifold in $\mathcal{U}$ with $\mathcal{W} = \mathcal{U}$ as in Definition 3.1, the mapping $(\mathbf{u}, \mathbf{v}) \mapsto \mathbf{f}(\mathbf{u}, \mathbf{v}) = \mathbf{u} - \phi^{-1}(\mathbf{v})$ from $\Sigma \times \Lambda$ in $\mathcal{U} \times \mathcal{V}$ to $\mathcal{W}$ satisfies conditions (i)-(iv).

Conversely, assume $\mathbf{f}$ satisfies all the specified conditions and we proceed to prove Π is a manifold in $\mathcal{U}$. From property (ii), we can write $\mathcal{U} = \hat{\mathcal{U}} \oplus \check{\mathcal{U}}$ with $\dim(\check{\mathcal{U}}) + \dim(\mathcal{V}) = \dim(\mathcal{W})$, regard $\mathcal{U}$ as $\hat{\mathcal{U}} \times \check{\mathcal{U}}$ and consider $\mathbf{f}$ as $(\hat{\mathbf{u}}, \check{\mathbf{u}}, \mathbf{v}) \mapsto \mathbf{f}(\hat{\mathbf{u}} + \check{\mathbf{u}}, \mathbf{v})$ from the domain $\Omega \subset \hat{\mathcal{U}} \times \check{\mathcal{U}} \times \mathcal{V}$ to $\mathcal{W}$ so that $\mathbf{f}_{\check{\mathbf{u}}\mathbf{v}}(\hat{\mathbf{u}}_0, \check{\mathbf{u}}_0, \mathbf{v}_0)$ is invertible where $\hat{\mathbf{u}}_0 + \check{\mathbf{u}}_0 = \mathbf{u}_0$. By the Implicit Mapping Theorem [21], there is a neighborhood $\Lambda \times \Delta$ of $\big(\hat{\mathbf{u}}_0,\ (\check{\mathbf{u}}_0, \mathbf{v}_0)\big)$ in $\hat{\mathcal{U}} \times (\check{\mathcal{U}} \times \mathcal{V})$, holomorphic mappings $\mathbf{g} : \Lambda \subset \hat{\mathcal{U}} \to \check{\mathcal{U}}$ and $\mathbf{h} : \Lambda \subset \hat{\mathcal{U}} \to \mathcal{V}$ such that $(\check{\mathbf{u}}_0,\ \mathbf{v}_0) = (\mathbf{g}(\hat{\mathbf{u}}_0),\ \mathbf{h}(\hat{\mathbf{u}}_0))$ and $\mathbf{f}\big(\hat{\mathbf{u}} + \check{\mathbf{u}}, \mathbf{v}\big) = \mathbf{0}$ for $\big(\hat{\mathbf{u}}, (\check{\mathbf{u}}, \mathbf{v})\big)$ in $\Lambda \times \Delta$ if and only if $(\check{\mathbf{u}}, \mathbf{v}) = (\mathbf{g}(\hat{\mathbf{u}}),\ \mathbf{h}(\hat{\mathbf{u}}))$. Without loss of generality, we assume $\Lambda \times \Delta = \Omega$ since we can redefine $\mathbf{f}$ with a restricted domain.

Let ψ be the holomorphic mapping $\hat{\mathbf{u}} \mapsto (\hat{\mathbf{u}},\ \mathbf{g}(\hat{\mathbf{u}}))$ from $\Lambda \subset \hat{\mathcal{U}}$ to $\hat{\mathcal{U}} \times \check{\mathcal{U}}$. Then $\psi(\Lambda) \subset \Pi$ since $(\psi(\hat{\mathbf{u}}), \mathbf{h}(\hat{\mathbf{u}})) = (\hat{\mathbf{u}}, \mathbf{g}(\hat{\mathbf{u}}), \mathbf{h}(\hat{\mathbf{u}}))$ is in $\mathbf{f}^{-1}(\mathbf{0})$ for all $\hat{\mathbf{u}} \in \Lambda$. We also have $\psi(\hat{\mathbf{u}}_0) = (\hat{\mathbf{u}}_0, \mathbf{g}(\hat{\mathbf{u}}_0)) = (\hat{\mathbf{u}}_0, \check{\mathbf{u}}_0)$. Let $\tilde{\Delta} = \{\mathbf{v} \in \mathcal{V} \mid (\check{\mathbf{u}}_0, \mathbf{v}) \in \Delta\}$ that is an open neighborhood $\mathbf{v}_0$ in $\mathcal{V}$. By the condition (iv), there is an open neighborhood Σ of $(\hat{\mathbf{u}}_0, \check{\mathbf{u}}_0)$ in $\hat{\mathcal{U}} \times \check{\mathcal{U}}$ such that every $(\hat{\mathbf{u}}, \check{\mathbf{u}}) \in \Sigma \cap \Pi$ corresponds to a unique $\mathbf{v} \in \tilde{\Delta}$ with $\mathbf{f}(\hat{\mathbf{u}}, \check{\mathbf{u}}, \mathbf{v}) = \mathbf{0}$. Denote $\tilde{\Lambda} = \psi^{-1}(\Sigma)$ that is open in $\hat{\mathcal{U}}$ and define the holomorphic mapping $\phi : (\hat{\mathbf{u}}, \check{\mathbf{u}}) \mapsto \hat{\mathbf{u}}$ from $\Sigma \subset \hat{\mathcal{U}} \times \check{\mathcal{U}}$ to $\hat{\mathcal{U}}$. Clearly $\phi \circ \psi(\hat{\mathbf{u}}) = \phi(\hat{\mathbf{u}}, \mathbf{g}(\hat{\mathbf{u}})) = \hat{\mathbf{u}}$ for all $\hat{\mathbf{u}} \in \tilde{\Lambda} \subset \Lambda$. Furthermore, for every $(\hat{\mathbf{u}}, \check{\mathbf{u}}) \in \Sigma \cap \Pi$, there is a unique $\mathbf{v} \in \tilde{\Delta}$ with $\mathbf{f}(\hat{\mathbf{u}}, \check{\mathbf{u}}, \mathbf{v}) = \mathbf{0}$ so that $(\check{\mathbf{u}}, \mathbf{v}) = (\mathbf{g}(\hat{\mathbf{u}}), \mathbf{h}(\hat{\mathbf{u}}))$. Namely $\psi \circ \phi(\hat{\mathbf{u}}, \check{\mathbf{u}}) = \psi(\hat{\mathbf{u}}) = (\hat{\mathbf{u}}, \mathbf{g}(\hat{\mathbf{u}})) = (\hat{\mathbf{u}}, \check{\mathbf{u}})$. Consequently, the subset Π is a manifold in $\mathcal{U} = \hat{\mathcal{U}} \times \check{\mathcal{U}}$ of dimension $\dim(\hat{\mathcal{U}})$ that equals to $\dim(\mathcal{U}) + \dim(\mathcal{V}) - \dim(\mathcal{W})$. □

Assuming the model of solving $\mathbf{f}(\mathbf{u},\mathbf{v}) = \mathbf{0}$ for $\mathbf{v}$ at the given data $\mathbf{u}$ is properly formulated so that the conditions of Theorem 3.2 are satisfied, the solution $\mathbf{v}$ is locally Lipschitz continuous with respect to the data $\mathbf{u}$ on the manifold.

Corollary 3.3. *Using the notations in Theorem 3.2, assume* $\mathbf{f}$ *satisfies the condition* (i)-(iv). *Further assume* $\mathcal{U}, \mathcal{V}$ *and* $\mathcal{W}$ *are normed and the isomorphisms from* $\mathcal{V}$ *and* $\mathcal{W}$ *to* $\mathbb{C}^{\dim(\mathcal{V})}$ *and* $\mathbb{C}^{\dim(\mathcal{W})}$, *respectively, are isometric. Then there is an open neighborhood* Ω_0 *of* $\mathbf{u}_0$ *in* $\mathcal{U}$ *such that, for every fixed parameter* $\mathbf{u}_1 \in \Omega_0 \cap \Pi$, *the equation* $\mathbf{f}(\mathbf{u}_1,\mathbf{v}) = \mathbf{0}$ *has a unique solution* $\mathbf{v}_1 \in \mathcal{V}$ *and*

$$\|\mathbf{v}_1-\mathbf{v}_0\| \le \|\mathbf{f}_\mathbf{v}(\mathbf{u}_0,\mathbf{v}_0)^\dagger\|\,\|\mathbf{f}_\mathbf{u}(\mathbf{u}_0,\mathbf{v}_0)\|\,\|\mathbf{u}_1-\mathbf{u}_0\|+o(\|\mathbf{u}_1-\mathbf{u}_0\|). \quad (1)$$

Proof. Using the notations in the proof of Theorem 3.2, we have $\mathbf{f}(\hat{\mathbf{u}}+\mathbf{g}(\hat{\mathbf{u}}),\mathbf{h}(\hat{\mathbf{u}})) \equiv \mathbf{0}$ for $\hat{\mathbf{u}} \in \Lambda$, implying the linear transformation

$$\mathbf{f}_{\hat{\mathbf{u}}}(\hat{\mathbf{u}}_0+\check{\mathbf{u}}_0,\mathbf{v}_0) + \mathbf{f}_{\check{\mathbf{u}}}(\hat{\mathbf{u}}_0+\check{\mathbf{u}}_0,\mathbf{v}_0)\circ\mathbf{g}_{\hat{\mathbf{u}}}(\hat{\mathbf{u}}_0) + \mathbf{f}_\mathbf{v}(\hat{\mathbf{u}}_0+\check{\mathbf{u}}_0,\mathbf{v}_0)\circ\mathbf{h}_{\hat{\mathbf{u}}}(\hat{\mathbf{u}}_0)$$

maps $\hat{\mathbf{u}}_1-\hat{\mathbf{u}}_0$ to $\mathbf{0}$ from all $\hat{\mathbf{u}}_1 \in \hat{\mathcal{U}}$. Furthermore, from

$$\begin{aligned}\mathbf{v}_1-\mathbf{v}_0 &= \mathbf{h}_{\hat{\mathbf{u}}}(\hat{\mathbf{u}}_0)\,(\hat{\mathbf{u}}_1-\hat{\mathbf{u}}_0) + h.o.t \text{ and}\\ \check{\mathbf{u}}_1-\check{\mathbf{u}}_0 &= \mathbf{g}_{\hat{\mathbf{u}}}(\hat{\mathbf{u}}_0)\,(\hat{\mathbf{u}}_1-\hat{\mathbf{u}}_0) + h.o.t.\end{aligned}$$

where $h.o.t.$ denotes the sum of higher order terms of $\hat{\mathbf{u}}_1-\hat{\mathbf{u}}_0$. Since $\mathcal{V}$ and $\mathcal{W}$ are isometrically isomorphic to $\mathbb{C}^{\dim(\mathcal{V})}$ and $\mathbb{C}^{\dim(\mathcal{W})}$ respectively so that $\mathbf{f}_\mathbf{v}(\mathbf{u}_0,\mathbf{v}_0)^\dagger$ is well-defined and we have (1) from

$$\begin{aligned}\mathbf{v}_1-\mathbf{v}_0 &= -\mathbf{f}_\mathbf{v}(\hat{\mathbf{u}}_0+\check{\mathbf{u}}_0,\mathbf{v}_0)^\dagger\left(\mathbf{f}_{\hat{\mathbf{u}}}(\hat{\mathbf{u}}_0+\check{\mathbf{u}}_0,\mathbf{v}_0)\,(\hat{\mathbf{u}}_1-\hat{\mathbf{u}}_0)\right.\\ &\qquad\left.+\,\mathbf{f}_{\check{\mathbf{u}}}(\hat{\mathbf{u}}_0+\check{\mathbf{u}}_0,\mathbf{v}_0)\,(\check{\mathbf{u}}_1-\check{\mathbf{u}}_0)\right)+h.o.t\\ &= -\mathbf{f}_\mathbf{v}(\hat{\mathbf{u}}_0+\check{\mathbf{u}}_0,\mathbf{v}_0)^\dagger\,\mathbf{f}_{\hat{\mathbf{u}}\check{\mathbf{u}}}(\hat{\mathbf{u}}_0+\check{\mathbf{u}}_0,\mathbf{v}_0)\left((\hat{\mathbf{u}}_1,\check{\mathbf{u}}_1)-(\hat{\mathbf{u}}_0,\check{\mathbf{u}}_0)\right)+h.o.t\\ &= -\mathbf{f}_\mathbf{v}(\mathbf{u}_0,\mathbf{v}_0)^\dagger\,\mathbf{f}_\mathbf{u}(\mathbf{u}_0,\mathbf{v}_0)\,(\mathbf{u}_1-\mathbf{u}_0)+h.o.t \quad \square\end{aligned}$$

The solution of a singular problem is known to be infinitely sensitive to *arbitrary* perturbations. In [15], Kahan discovers an inherently bounded stability under perturbations *constrained* on certain heuristically conceived "pejorative manifolds" for the root-finding and the eigenvalue problems. Theorem 3.2 rigorously establishes the conditions for modeling general algebraic problems so that data points indeed form manifolds on which the solutions maintain certain structures. Corollary 3.3 further quantifies the bounded sensitivity on those manifolds. More importantly, the bounded sensitivity can be extended beyond the manifold into its tubular neighborhood, making it possible to harness the stability in practical computation from empirical data as we shall elaborate in §6.

4 GEOMETRIC MODELING CASE STUDIES

Algebraic problems are often phrased in a pattern of finding a certain solution at a data point, such as "find the kernel of a matrix", "find the greatest common divisor of a polynomial pair", "find the Jordan Canonical Form of a matrix", "find the factorization of a polynomial". The data point can usually be represented as a vector $\mathbf{u} = \hat{\mathbf{u}}$ in a vector space $\mathcal{U}$. The key to the geometric analysis and the accurate solution of those problems is to model the solution as a vector $\mathbf{v}$ in a vector space $\mathcal{V}$ in a zero-finding problem:

$$\text{At } \hat{\mathbf{u}} \in \mathcal{U}, \text{ solve the equation } \mathbf{f}(\hat{\mathbf{u}},\mathbf{v}) = \mathbf{0} \text{ for } \mathbf{v} \in \mathcal{V} \quad (2)$$

where $\mathbf{f} : (\mathbf{u},\mathbf{v}) \mapsto \mathbf{f}(\mathbf{u},\mathbf{v})$ is a holomorphic mapping from an open domain $\Omega \subset \mathcal{U}\times\mathcal{V}$. By adding proper auxiliary equations, the model can be set up so that the mapping $\mathbf{f}$ satisfies the conditions (i)-(iv) in Theorem 3.2. Consequently, a collection of the data points at which the solutions possess a specific algebraic structure can be established as a *structure-preserving manifold*, making it possible to apply the Tubular Neighborhood Theorem (Theorem 6.2). The model (2) also enables computation of an approximate solution as the least squares solution $\mathbf{v} = \tilde{\mathbf{v}}$ of the equation $\mathbf{f}(\tilde{\mathbf{u}},\mathbf{v}) = \mathbf{0}$. We elaborate such geometric modeling in case studies in this section.

4.1 The matrix rank-revealing problem

In $\mathbb{C}^{m\times n}$ of $m\times n$ matrices of complex entries with the Frobenius norm $\|\cdot\|_F$, the subset $\mathcal{C}_r^{m\times n} = \left\{A \in \mathbb{C}^{m\times n} \,\middle|\, \text{rank}\,(A) = r\right\}$ is a manifold of codimension $(m-r)(n-r)$. This result is proved in [7] and can be easily verified via using Theorem 3.2 as follows.

Let O and I denote the zero and identity matrices, respectively, in $\mathbb{C}^{m\times n}$. At a matrix $A \in \mathbb{C}^{m\times n}$ of rank-r, consider the rank-revealing problem as finding the kernel $\mathcal{K}(A)$ of dimension $n-r$. The fundamental equation is $GX = O$ for $X \in \mathbb{C}^{n\times(n-r)}$ at the data point $G = A$. The crucial auxiliary equation that ensures proper modeling under Theorem 3.2 can be derived from the fact that, for almost all $C \in \mathbb{C}^{n\times(n-r)}$, there is an $N \in \mathbb{C}^{n\times(n-r)}$ whose columns form a basis for $\mathcal{K}(A)$ such that $C^{\mathrm{H}}N = I$. Finding the kernel of A can then be modeled as a zero-finding problem:

$$\text{Solve } \mathbf{f}(A,X) = (O,O) \text{ for } X \in \mathbb{C}^{n\times(n-r)}$$

where, with a fixed parameter $C \in \mathbb{C}^{n\times(n-r)}$, the mapping $\mathbf{f}$ from $\Omega \subset \mathbb{C}^{m\times n}\times\mathbb{C}^{n\times(n-r)}$ to $\mathbb{C}^{(n-r)\times(n-r)}\times\mathbb{C}^{m\times(n-r)}$ is

$$\mathbf{f} : (G,\,X) \longmapsto (C^{\mathrm{H}}X - I,\; GX). \quad (3)$$

Here Ω is an open neighborhood of (A,N) and, for every $(G,X) \in \Omega$, we have $\|A-G\|_F < \|A^\dagger\|_2^{-1}$. Clearly $\mathbf{f}(A,N) = (O,O)$ and $(G,X) \in \mathbf{f}^{-1}(O,O)$ implies G has the desired algebraic structure of rank r, leading to the condition (i) and (iii) of Theorem 3.2. The Jacobian $\mathbf{f}_{GX}(A,N) : (G,X) \mapsto (C^{\mathrm{H}}X, GN + AX)$ is surjective since both C and N are of full rank $n-r$. The partial Jacobian $\mathbf{f}_X(A,N) : X \mapsto (C^{\mathrm{H}}X, AX)$ is injective since $(C^{\mathrm{H}}X, AX) = (O,O)$ implies $X = NT$ for a certain $T \in \mathbb{C}^{(n-r)\times(n-r)}$, $O = C^{\mathrm{H}}X = T$ and $X = O$, leading to the condition (ii) of Theorem 3.2. Furthermore, every matrix $G \in \mathcal{C}_r^{m\times n}$ sufficiently close to A corresponds to a matrix $X \in \mathbb{C}^{n\times(n-r)}$ whose column span $\mathcal{K}(G)$ and $C^{\mathrm{H}}X = I$ so $\mathbf{f}(G,X) = (O,O)$ and $\|X-N\|_F$ can be as small as we wish, validating the condition (iv) of Theorem 3.2. As a result, the subset $\mathcal{C}_r^{m\times n}$ is a manifold of codimension

$$\text{codim}(\mathcal{C}_r^{m\times n}) = (n-r)^2 + m\,(n-r) - n\,(n-r) = (m-r)\,(n-r).$$

The subset $\mathcal{C}_r^{m\times n}$ for every r is a structure-preserving manifold for the rank-revealing problem and the desired solution (rank and kernel) is modeled in the vector $X \in \mathbb{C}^{n\times(n-r)}$ as the zero of the mapping $X \mapsto \mathbf{f}(\hat{G},X)$ at $\hat{G} \in \mathcal{C}_r^{m\times n}$.

4.2 The root-finding problem

A polynomial can be considered as a data vector in the vector space $\mathbb{P}_n$ of polynomials with degrees up to n and the norm

$$\|a_0 + a_1\,x + \cdots + a_n\,x^n\| := \|(a_0,a_1,\ldots,a_n)\|_2$$

that makes $\mathbb{P}_n$ isometrically isomorphic to $\mathbb{C}^{n+1}$. The complete root-finding problem of a polynomial is equivalent to its factorization.

For any positive integers $\ell_1 + \cdots + \ell_k = n$, denote

$$\mathcal{F}_{\ell_1 \cdots \ell_k} := \tag{4}$$
$$\left\{ \alpha\,(x - z_1)^{\ell_1} \cdots (x - z_k)^{\ell_k} \,\middle|\, \alpha, z_1, \ldots, z_k \in \mathbb{C}, z_i \neq z_j,\ \forall i \neq j \right\}.$$

Every polynomial $p \in \mathbb{P}_n$ belongs to one of such a subset in which the factorization structure is preserved. The root-finding problem of p becomes calculating the distinct roots $\mathbf{z}_1, \ldots, z_k$ and multiplicities $\ell_1, \ldots, \ell_k$. At any $p \in \mathcal{F}_{\ell_1 \cdots \ell_k}$ with leading coefficient u_0 and distinct roots $u_1, \ldots, u_k$ of multiplicities $\ell_1, \ldots, \ell_k$ respectively, the root-finding problem of p can thus be modeled as identifying $\mathcal{F}_{\ell_1 \cdots \ell_k}$ and solving a zero-finding problem in the form of the modified Vieťe's equation

$$\phi(\mathbf{z}, p) = 0 \quad \text{for} \quad \mathbf{z} = (z_0, z_1, \ldots, z_k) \in \mathbb{C}^{k+1} \tag{5}$$

with the holomorphic mapping from $\Omega \subset \mathbb{C}^{k+1} \times \mathbb{P}_n$ to $\mathbb{P}_n$

$$\phi \ : \ (\mathbf{z},\ g) \ \longmapsto \ z_0\,(x - z_1)^{\ell_1} \cdots (x - z_k)^{\ell_k} - g \tag{6}$$

where Ω is an open neighborhood of $\mathbf{u} = (u_0, u_1, \ldots, u_k)$ in $\mathbb{C}^{k+1}$ in which every $\mathbf{y} = (y_0, y_1, \ldots, y_k) \in \Omega$ implies $(y_0, y_{i_1}, \ldots, y_{i_k}) \notin \Omega$ whenever the permutation $(i_1, \ldots, i_k) \neq (1, \ldots, k)$. Such a geometric modeling leads to the geometric insight in the following theorem along with a proof that is made simple by Theorem 3.2. The theorem sets the foundation for the accurate solution of root-finding problem in the presence of multiple roots. The theorem is proposed in [25] by this author with an incomplete proof due to necessary abbreviation under the page limit.

Theorem 4.1. *The subset $\mathcal{F}_{\ell_1 \cdots \ell_k}$ is a complex analytic manifold in $\mathbb{P}_n$ of codimension $n - k$ where $n = \ell_1 + \cdots + \ell_k$.*

Proof. For any $p = u_0\,(x - u_1)^{\ell_1} \ldots (x - u_k)^{\ell_k} \in \mathcal{F}_{\ell_1 \cdots \ell_k}$ with distinct roots $u_1, \ldots, u_k$, define ϕ as in (6) at p so $\phi(\mathbf{u}, p) = 0$ where $\mathbf{u} = (u_0, \ldots, u_k)$. For any $g \in \mathbb{P}_n$, we have $\phi_{\mathbf{z}g}(\mathbf{u}, p)(\mathbf{0}, g) \equiv -g$, implying $\phi_{\mathbf{z}g}(\mathbf{u}, p)$ is surjective. With a proof nearly identical to that of Theorem 3.3 in [24], the partial Jacobian $\phi_{\mathbf{z}}(\mathbf{u}, p)$ is injective. Moreover, the continuity of polynomial roots with respect to the coefficients ensures the condition (iv) of Theorem 3.2 is satisfied, concluding the proof. □

We call $\mathcal{F}_{\ell_1 \cdots \ell_k}$ a *factorization manifold* in $\mathbb{P}_n$. Factorization manifolds serve as structure-preserving manifolds for polynomials in $\mathbb{P}_n$. The desired factorization is represented by the vector $(z_0, z_1, \ldots, z_k)$ in $\mathbb{C}^{k+1}$ in the zero-finding model (5). The root-finding problem is thus equivalent to identifying the factorization manifold $\mathcal{F}_{\ell_1 \cdots \ell_k}$ along with the zero-finding problem (5).

Modeling the factorization problem for polynomials including multivariate cases is given in [23] where the proof of the Factorization Manifold Theorem can be substantially simplified by citing Theorem 3.2 rather than essentially mirroring its proof.

4.3 The greatest common divisor problem

We say two polynomials are $\sim$-equivalent if they are constant multiples of each other. For every $(p, q) \in \mathbb{P}_m \times \mathbb{P}_n$, let $\gcd(p, q)$ denote the greatest common divisor (GCD) of p and q as an equivalent class under $\sim$. The subset $\mathcal{P}^k_{m,n}$ defined as

$$\left\{ (p, q) \in \mathbb{P}_m \times \mathbb{P}_n \,\middle|\, \deg(p) = m, \deg(q) = n, \deg(\gcd(p, q)) = k \right\}$$

is a manifold of codimension k in $\mathbb{P}_m \times \mathbb{P}_n$ where $\deg(\cdot)$ is the degree of any polynomial $(\cdot)$, as asserted in [26]. To establish this result, we model the GCD computation as a zero-finding problem. At any particular $(\hat{p}, \hat{q}) \in \mathcal{P}^k_{m,n}$, there is a $(u, v, w) = (\hat{u}, \hat{v}, \hat{w})$ satisfying the quations $u\,v - p = u\,w - q = 0$ at the data $(p, q) = (\hat{p}, \hat{q})$ with $\hat{u} \in \gcd(\hat{p}, \hat{q})$. To ensure proper modeling, we need an auxiliary equation $r \odot u = \beta \neq 0$ for almost all $r \in \mathbb{P}_k$ such as a random polynomial where $\odot$ is the dot-product between two polynomials defined as the dot-product between the corresponding coefficient vectors. Using such r and β as parameters, the GCD problem of the pair $(\hat{p}, \hat{q})$ can be modeled as identifying the GCD degree k and

$$\text{Solve} \ \ \psi(u, v, w, \hat{p}, \hat{q}) = (0, 0, 0) \ \ \text{for} \ \ (u, v, w) \in \mathbb{P}_k \times \mathbb{P}_{m-k} \times \mathbb{P}_{n-k}$$

with the holomorphic mapping

$$\psi : \Omega \subset \mathbb{P}_k \times \mathbb{P}_{m-k} \times \mathbb{P}_{n-k} \times \mathbb{P}_m \times \mathbb{P}_n \longrightarrow \mathbb{C} \times \mathbb{P}_m \times \mathbb{P}_n$$
$$(u, v, w, p, q) \longmapsto (r \odot u - \beta,\ u\,v - p,\ u\,w - q) \tag{7}$$

where Ω is a neighborhood of $(\hat{u}, \hat{v}, \hat{w}, \hat{p}, \hat{q})$ in the product space $\mathbb{P}_k \times \mathbb{P}_{m-k} \times \mathbb{P}_{n-k} \times \mathbb{P}_m \times \mathbb{P}_n$ such that every $(u, v, w, p, q) \in \Omega$ satisfies $\deg(p) = m$, $\deg(q) = n$ and $\deg(u) = k$ with the pair (v, w) being coprime. Clearly $\psi(\hat{u}, \hat{v}, \hat{w}, \hat{p}, \hat{q}) = (0, 0, 0)$. The Jacobian

$$\psi_{uvwpq}(\hat{u}, \hat{v}, \hat{w}, \hat{p}, \hat{q}) \ :$$
$$(u, v, w, p, q) \ \mapsto \ (r \odot u,\ \hat{u}\,v + u\,\hat{v} - p, \hat{u}\,w + u\,\hat{w} - q)$$

can be easily verified to be surjective. The injectivity of the partial Jacobian $\psi_{uvw}(\hat{u}, \hat{v}, \hat{w}, \hat{p}, \hat{q})$ is established by [26, Corollary 4.1]. At every $(u, v, w, p, q) \in \Omega$, the equality $\psi(u, v, w, p, q) = (0, 0, 0)$ implies $(p, q) \in \mathcal{P}^k_{m,n}$. It is also a straightforward verification that, for every $(p, q) \in \mathcal{P}^k_{m,n}$ sufficiently close to $(\hat{p}, \hat{q})$, there is a unique $(u, v, w) \in \mathbb{P}_k \times \mathbb{P}_{m-k} \times \mathbb{P}_{n-k}$ such that $\psi(u, v, w, p, q) = (0, 0, 0)$ with $\|(u, v, w) - (\hat{u}, \hat{v}, \hat{w})\|$ as small as we wish. By Theorem 3.2, the subset $\mathcal{P}^k_{m,n}$ is a manifold in $\mathbb{P}_m \times \mathbb{P}_n$ of the codimension

$$\dim(\mathbb{C} \times \mathbb{P}_m \times \mathbb{P}_n) - \dim(\mathbb{P}_k \times \mathbb{P}_{m-k} \times \mathbb{P}_{n-k}) = k.$$

Each manifold amoung $\mathcal{P}^0_{m,n}$, $\mathcal{P}^1_{m,n}$, …, $\mathcal{P}^{\min\{m,n\}}_{m,n}$ preserves a GCD structure (degree) for polynomial pairs on it.

4.4 The Jordan Canonical Form problem

The collection of $n \times n$ matrices with a fixed structure of Jordan Canonical Form (JCF) in terms of the Segre characteristics is called a *bundle* that is proved to be a manifold [2, 12] through differential geometry. Bundles can be established as manifolds using the geometric modeling approach and Theorem 3.2 but the complete proof is beyond the scope of this paper. We illustrate the geometric modeling of a bundle using a specific JCF structure here. Let

$$\Pi = \left\{ X\,J_n(\lambda)\,X^{-1} \,\middle|\, \lambda \in \mathbb{C}, X \in \mathbb{C}^{n \times n} \text{ is invertible} \right\}$$

where $J_n(\lambda)$ denotes the $n \times n$ elementary Jordan block with the eigenvalue λ. Namely Π is the collection of all $n \times n$ matrices with a single eigenvalue in a single Jordan block. The JCF problem with respect to this Jordan structure can be modeled as follows. At any $A \in \Pi$, pick a random vector $\mathbf{c} \in \mathbb{C}^n$. For almost all such $\mathbf{c}$, there is a unique invertible matrix $X \in \mathbb{C}^{n \times n}$ whose columns are eigenvector and generalized eigenvectors such that $A\,X = X\,J_n(\lambda_*)$ along with the auxiliary equation $\mathbf{c}^{\mathrm{H}}\,X = [1, 0, \cdots, 0]$.

$$\text{Solve} \ \ \mathbf{g}(A, \lambda, Z) = (\mathbf{0}, O) \ \ \text{for} \ \ (\lambda, Z) \in \mathbb{C} \times \mathbb{C}^{n \times n}$$

with the holomorphic mapping from $\Omega \subset \mathbb{C}^{n \times n} \times \mathbb{C} \times \mathbb{C}^{n \times n}$ to $\mathbb{C}^{1 \times n} \times \mathbb{C}^{n \times n}$ as $\mathbf{g} : (G, \lambda, Z) \mapsto \left(\mathbf{c}^{\mathrm{H}} Z - [1, 0, \cdots, 0], G\,Z - Z\,J_n(\lambda)\right)$

where Ω is a neighborhood of (A, λ_*, X) in which all (G, λ, Z) has an invertible Z and nonzero dot-product between $\mathbf{c}$ and the lone eigenvector of G. The fact that the subset Π is a manifold can be established by verifying the four conditions in Theorem 3.2 on $\mathbf{g}$ using common techniques in linear algebra, and

$$\operatorname{codim}(\Pi) = \dim(\mathbb{C}^{1\times n} \times \mathbb{C}^{n\times n}) - \dim(\mathbb{C} \times \mathbb{C}^{n\times n}) = n - 1.$$

5 THE LEAST SQUARES PROBLEM

As elaborated in §4, an algebraic problems can be modeled as a zero-finding problem in the form of $\mathbf{f}(\mathbf{u}, \mathbf{v}) = \mathbf{0}$ for the variable $\mathbf{v}$ at a certain fixed parameter $\mathbf{u}$, and the equation is often overdetermined. In practical computation, the parameter $\mathbf{u}$ is expected to be represented via empirical data $\tilde{\mathbf{u}}$ at which the exact solution $\mathbf{v}$ generally does not exist for the perturbed equation $\mathbf{f}(\tilde{\mathbf{u}}, \mathbf{v}) = \mathbf{0}$. The resulting model becomes a least squares problem.

Let $\mathcal{V}$ and $\mathcal{W}$ be normed vector spaces isometrically isomorphic to $\mathbb{C}^n$ and $\mathbb{C}^m$ respectively with $m > n$. Let $\mathbf{x} \mapsto \mathbf{f}(\mathbf{x})$ be a mapping from an open subset Ω of $\mathcal{V}$ to $\mathcal{W}$. Since $\mathbf{f}(\Omega)$ is of dimension at most n in $\mathcal{W}$ with $\dim(\mathcal{W}) = m > n$, conventional solutions to the equation $\mathbf{f}(\mathbf{x}) = \mathbf{b}$ do not exist in general. Instead, we seek a *least squares solution* $\mathbf{x}_* \in \Lambda$ of $\mathbf{f}(\mathbf{x}) = \mathbf{b}$ such that

$$\left\|\mathbf{f}(\mathbf{x}_*) - \mathbf{b}\right\|^2 = \min_{\mathbf{x}\in\Lambda} \left\|\mathbf{f}(\mathbf{x}) - \mathbf{b}\right\|^2$$

where $\Lambda \subset \Omega$ is an open neighborhood of $\mathbf{x}_*$. In other words, we seek $\mathbf{x}_*$ so that $\mathbf{f}(\mathbf{x}_*)$ is the projection of $\mathbf{b}$ to the surface $\mathbf{f}(\Omega)$, minimizing the distance from $\mathbf{b}$ to $\mathbf{f}(\Omega)$. Further assume $\mathcal{V}$ and $\mathcal{W}$ are isometrically isomorphic to $\mathbb{C}^n$ and $\mathbb{C}^m$ respectively so that $\mathbf{f}_\mathbf{x}(\mathbf{z})^{\mathsf{H}}$ and $\mathbf{f}_\mathbf{x}(\mathbf{z})^\dagger$ are well defined. Then a least squares solution is a *critical point* for the equation $\mathbf{f}(\mathbf{x}) = \mathbf{b}$, namely (c.f. [24])

$$\mathbf{f}_\mathbf{x}(\mathbf{x}_*)^{\mathsf{H}} \left(\mathbf{f}(\mathbf{x}_*) - \mathbf{b}\right) = \mathbf{0}. \tag{8}$$

The *Gauss-Newton iteration*[1]

$$\mathbf{x}_{k+1} = \mathbf{x}_k - \mathbf{f}_\mathbf{x}(\mathbf{x}_k)^\dagger \left(\mathbf{f}(\mathbf{x}_k) - \mathbf{b}\right) \text{ for } k = 0, 1, \ldots \tag{9}$$

is effective in finding the least squares solution of $\mathbf{f}(\mathbf{x}) = \mathbf{b}$ and is locally convergent. The following lemma provides detailed convergence conditions in Kantorovich style.

Lemma 5.1. [25] *Let $\mathcal{V}$ and $\mathcal{W}$ be finite-dimensional normed vector spaces isometrically isomorphic to $\mathbb{C}^n$ and $\mathbb{C}^m$ respectively. Assume $\mathbf{x} \mapsto \mathbf{f}(\mathbf{x})$ is a holomorphic mapping from an open domain $\Omega \subset \mathcal{V}$ to $\mathcal{W}$ with a critical point $\mathbf{x}_* \in \Omega$ of the system $\mathbf{f}(\mathbf{x}) = \mathbf{b}$ and $\mathbf{f}_\mathbf{x}(\mathbf{x}_*)$ is injective. Then there is an open neighborhood $\Lambda \subset \Omega$ of $\mathbf{x}_*$ along with constants ζ, $\gamma > 0$ such that*

$$\left\|\mathbf{f}_\mathbf{x}(\mathbf{z})^\dagger\right\| \le \zeta \text{ and } \left\|\mathbf{f}(\mathbf{z}) - \mathbf{f}(\tilde{\mathbf{z}}) - \mathbf{f}_\mathbf{x}(\tilde{\mathbf{z}})(\mathbf{z} - \tilde{\mathbf{z}})\right\| \le \gamma \left\|\mathbf{z} - \tilde{\mathbf{z}}\right\|^2 \tag{10}$$

for all $\mathbf{z}$, $\tilde{\mathbf{z}} \in \Lambda$. Further assume $\|\mathbf{f}(\mathbf{x}_*) - \mathbf{b}\|$ is small so that

$$\left\|\left(\mathbf{f}_\mathbf{x}(\mathbf{z})^\dagger - \mathbf{f}_\mathbf{x}(\mathbf{x}_*)^\dagger\right)\left(\mathbf{f}(\mathbf{x}_*) - \mathbf{b}\right)\right\| \le \sigma \left\|\mathbf{z} - \mathbf{x}_*\right\| \tag{11}$$

for a constant $\sigma < 1$ at every $\mathbf{z} \in \Lambda$. Then for all $\mathbf{x}_0 \in \Lambda$ satisfying

$$\left\|\mathbf{x}_0 - \mathbf{x}_*\right\| < \tfrac{1-\sigma}{\zeta\gamma} \text{ and } \{\mathbf{x} \in \mathcal{V} \mid \|\mathbf{x} - \mathbf{x}_*\| < \|\mathbf{x}_0 - \mathbf{x}_*\|\} \subset \Lambda,$$

the iteration (9) *is well defined in Λ, converges to $\mathbf{x}_*$, and satisfies*

$$\left\|\mathbf{x}_{k+1} - \mathbf{x}_*\right\| \le \left(\sigma + \zeta\gamma \left\|\mathbf{x}_k - \mathbf{x}_*\right\|\right) \left\|\mathbf{x}_k - \mathbf{x}_*\right\|$$

for $k = 0, 1, \ldots$ with $\sigma + \zeta\gamma \left\|\mathbf{x}_0 - \mathbf{x}_*\right\| < 1$.

[1] A general purpose MATLAB module `GaussNewton` is implemented in the package NAClab [31] with an intuitive interface [28].

6 TUBULAR NEIGHBORHOOD THEOREM

The very reason we need to establish manifolds in regularizing ill-posed algebraic problems lies in one of the fundamental theorems in differential geometry: A smooth manifold is contained in an open *tubular neighborhood* in which every point can be uniquely projected onto the manifold following a normal line and the projection mapping possesses certain desired properties. The concept of tubular neighborhood is also regarded as "one of the most useful notions in the theory of differential manifolds" [8]. Standard versions of the tubular neighborhood theorem for *real* smooth manifolds can be found in textbooks of differential geometry (see e.g. [4]). Those versions are presented in abstract forms for general purposes and do not appear to be applicable to our geometric modles involving *complex* analytic manifolds. For the applications in regularization of ill-posed algebraic problems, the projection to the manifold does not need to be holomorphic and it suffices to be Lipschitz continuous with the Lipschitz constant serving as a condition number measuring the sensitivity of the underlying problem.

Lemma 6.1. *Let $\mathcal{U}$, $\mathcal{V}$ and $\mathcal{W}$ be normed vector spaces over $\mathbb{C}$ that are isometrically isomorphic to $\mathbb{C}^l$, $\mathbb{C}^m$ and $\mathbb{C}^n$ respectively with $m \le n \le l + m$. Assume Π is a complex analytic manifold in $\mathcal{U}$ and, for every $\mathbf{u}_0 \in \Pi$, there is a holomorphic mapping $(\mathbf{u}, \mathbf{v}) \mapsto \mathbf{f}(\mathbf{u}, \mathbf{v})$ from an open domain $\Omega \subset \mathcal{U} \times \mathcal{V}$ to $\mathcal{W}$ satisfying the conditions* (i)-(iv) *in Theorem 3.2. Then the following assertions hold:*

(i) *There are open neighborhoods Ψ of $\mathbf{u}_0$ in $\mathcal{U}$ and Φ of $\mathbf{v}_0$ in $\mathcal{V}$ along with $\pi : \Psi \subset \mathcal{U} \to \mathcal{V}$ whose image $\tilde{\mathbf{v}} = \pi(\tilde{\mathbf{u}}) \in \Pi$ is the unique least squares solution to the equation $\mathbf{f}(\tilde{\mathbf{u}}, \mathbf{v}) = \mathbf{0}$ in Φ at every $\tilde{\mathbf{u}} \in \Psi$. Furthermore, for every open neighborhood $\check{\Phi} \subset \Phi$ of $\mathbf{v}_0$ in $\mathcal{V}$, there is an open neighborhood $\check{\Psi} \subset \Psi$ of $\mathbf{u}_0$ such that $\pi(\check{\Psi}) \subset \check{\Phi}$.*

(ii) *The mapping π is locally Lipschitz continuous in Ψ.*

(iii) *From every $\tilde{\mathbf{u}} \in \Psi$ serving as empirical data for $\mathbf{u}_0$, the least squares solution $\pi(\tilde{\mathbf{u}}) = \tilde{\mathbf{v}}$ satsifies*

$$\|\tilde{\mathbf{v}} - \mathbf{v}_0\| \le \|\mathbf{f}_\mathbf{v}(\mathbf{u}_0, \mathbf{v}_0)^\dagger\| \, \|\mathbf{f}_\mathbf{u}(\mathbf{u}_0, \mathbf{v}_0)\| \, \|\tilde{\mathbf{u}} - \mathbf{u}_0\| + o(\|\tilde{\mathbf{u}} - \mathbf{u}_0\|) \tag{12}$$

Proof. Using the notations in the proof of Theorem 3.2, there exists a bounded open neighborhood Σ of $\mathbf{v}_0$ in $\mathcal{V}$ such that the subset $\{\hat{\mathbf{u}}_0\}\times(\{\check{\mathbf{u}}_0\}\times\overline{\Sigma}) \subset \Lambda\times\Lambda$. For any $r > 0$ and the subset $\Phi_r := \{\mathbf{v} \in \Sigma \mid \|\mathbf{v} - \mathbf{v}_0\| < r\}$, we claim there is an $s > 0$ such that, at every $\tilde{\mathbf{u}} \in \Psi_s := \{\mathbf{u} \in \mathcal{U} \mid \|\mathbf{u} - \mathbf{u}_0\| < r\}$, the minimum $\min_{\mathbf{v}\in\overline{\Phi_r}} \|\mathbf{f}(\tilde{\mathbf{u}}, \mathbf{v})\|$ occurs at a certain $\tilde{\mathbf{v}} \in \Phi_r$ that is a least squares solution of $\mathbf{f}(\tilde{\mathbf{u}}, \mathbf{v}) = \mathbf{0}$. Assume otherwise. Then there is a sequence $\{\mathbf{u}_j\}_{j=1}^\infty$ converging to $\mathbf{u}_0$ such that $\min_{\mathbf{v}\in\overline{\Phi_r}} \|\mathbf{f}(\mathbf{u}_j, \mathbf{v})\| = \|\mathbf{f}(\mathbf{u}_j, \mathbf{v}_j)\|$ at $\mathbf{v}_j \in \overline{\Phi_r} \setminus \Phi_r$ for every $j = 1, 2, \ldots$. Since $\overline{\Phi_r} \setminus \Phi_r$ is compact, we can assume $\mathbf{v}_j$ converges to a certain $\check{\mathbf{v}}$. Thus

$$\|\mathbf{f}(\mathbf{u}_0, \check{\mathbf{v}})\| = \lim_{j\to\infty} \|\mathbf{f}(\mathbf{u}_j, \mathbf{v}_j)\| \le \lim_{j\to\infty} \|\mathbf{f}(\mathbf{u}_j, \mathbf{v}_0)\| = 0,$$

implying $\check{\mathbf{v}} = \mathbf{v}_0$ that contradicts to $\check{\mathbf{v}} \in \overline{\Phi_r} \setminus \Phi_r$.

We can assume $r_1 > 0$ is sufficiently small so that, for every $\mathbf{v}_1, \mathbf{v}_2 \in \Phi_{r_1}$ and $\mathbf{u} \in \Psi_{s_1}$, there exist constants $\zeta, \gamma > 0$ such that

$$\|\mathbf{f}(\mathbf{u}, \mathbf{v}_2) - \mathbf{f}(\mathbf{u}, \mathbf{v}_1) - \mathbf{f}_\mathbf{v}(\mathbf{u}, \mathbf{v}_1)(\mathbf{v}_2 - \mathbf{v}_1)\| < \gamma \|\mathbf{v}_2 - \mathbf{v}_1\|^2$$
$$\left\|\left(\mathbf{f}_\mathbf{v}(\mathbf{u}, \mathbf{v}_2)^\dagger - \mathbf{f}_\mathbf{v}(\mathbf{u}, \mathbf{v}_1)^\dagger\right)\mathbf{f}(\mathbf{u}, \mathbf{v}_1)\right\| < \tfrac{1}{2}\|\mathbf{v}_2 - \mathbf{v}_1\|$$
$$\|\mathbf{f}_\mathbf{v}(\mathbf{u}, \mathbf{v}_1)^\dagger\| < \zeta, \quad \|\mathbf{v}_2 - \mathbf{v}_1\| < \tfrac{1}{2\zeta\gamma}.$$

Let $r_2 = \frac{1}{3} r_1$, $\Phi = \Phi_{r_2}$ and $\Psi = \Psi_{s_1} \cap \Psi_{s_2}$. For every $\hat{\mathbf{u}} \in \Psi$, the minimum $\min_{\mathbf{v}\in\overline{\Phi}} \|\mathbf{f}(\hat{\mathbf{u}}, \mathbf{v})\|$ is attainable at a certain $\hat{\mathbf{v}} \in \Phi$ and, for any initial iterate $\mathbf{v}_1 \in \Phi$, we have $\|\mathbf{v}_1 - \hat{\mathbf{v}}\| < \frac{1}{2\zeta\gamma} = (1-\frac{1}{2})\frac{1}{\zeta\gamma}$ and the set $\Omega = \{\mathbf{v} \in \mathcal{V} | \|\mathbf{v} - \hat{\mathbf{v}}\| < \|\mathbf{v}_1 - \hat{\mathbf{v}}\|\}$ is a subset of Φ_{r_1} since, for every $\mathbf{v} \in \Omega$, we have

$$\begin{aligned}\|\mathbf{v} - \mathbf{v}_0\| &\le \|\mathbf{v} - \hat{\mathbf{v}}\| + \|\hat{\mathbf{v}} - \mathbf{v}_0\| < \|\mathbf{v} - \hat{\mathbf{v}}\| + r_2 < \|\mathbf{v}_1 - \hat{\mathbf{v}}\| + r_2 \\ &\le \|\mathbf{v}_1 - \mathbf{v}_0\| + \|\mathbf{v}_0 - \hat{\mathbf{v}}\| + r_2 < r_2 + r_2 + r_2 = r_1\end{aligned}$$

By Lemma 5.1, for every initial iterate $\mathbf{v}_1 \in \Phi$, the Gauss-Newton iteration on the equation $\mathbf{f}(\hat{\mathbf{u}}, \mathbf{v}) = \mathbf{0}$ converges to $\hat{\mathbf{v}}$. This local minimum is unique in Φ because, assuming there is another minimum point $\check{\mathbf{v}} \in \Phi$ of $\|\mathbf{f}(\hat{\mathbf{u}}, \mathbf{v})\|$, the Gauss-Newton iteration converges to $\hat{\mathbf{v}}$ from the initial iterate $\check{\mathbf{v}}$. On the other hand, the Gauss-Newton iteration from the local minimum point $\check{\mathbf{v}}$ stays at $\check{\mathbf{v}}$, implying $\check{\mathbf{v}} = \hat{\mathbf{v}}$ and thus the existence of the mapping π. Given any open subset $\check{\Phi}$ of Φ, there is an open subset $\check{\Psi}$ of Ψ for the same reason that Ψ_s exists such that the minimum $\min_{\mathbf{v}\in\overline{\check{\Phi}}} \|\mathbf{f}(\tilde{\mathbf{u}}, \mathbf{v})\|$ is attainable at a certain $\tilde{\mathbf{v}} \in \check{\Phi}$ for every fixed $\tilde{\mathbf{u}} \in \check{\Psi}$. This $\tilde{\mathbf{v}}$ is unique in Φ since $\tilde{\mathbf{u}} \in \Psi$, and thus $\tilde{\mathbf{v}}$ is unique in $\check{\Phi}$, implying $\tilde{\mathbf{v}} = \pi(\tilde{\mathbf{u}})$ so that $\pi(\check{\Psi}) \subset \check{\Phi}$.

On the Lipschitz continuity the mapping π, let $\tilde{\mathbf{u}}, \hat{\mathbf{u}} \in \Psi$ with $\pi(\tilde{\mathbf{u}}) = \tilde{\mathbf{v}}$ and $\pi(\hat{\mathbf{u}}) = \hat{\mathbf{v}}$. The one-step Gauss-Newton iteration $\mathbf{v}_1 = \tilde{\mathbf{v}} - \mathbf{f}_{\mathbf{v}}(\hat{\mathbf{u}}, \tilde{\mathbf{v}})^\dagger \mathbf{f}(\hat{\mathbf{u}}, \tilde{\mathbf{v}})$ from $\tilde{\mathbf{v}}$ on the equation $\mathbf{f}(\hat{\mathbf{u}}, \mathbf{v}) = \mathbf{0}$ toward $\hat{\mathbf{v}}$ yields the iniquality $\|\mathbf{v}_1 - \hat{\mathbf{v}}\| \le \mu \|\tilde{\mathbf{v}} - \hat{\mathbf{v}}\|$ with $\mu < 1$ by Lemma 5.1. Thus

$$\|\tilde{\mathbf{v}} - \hat{\mathbf{v}}\| \le \|\hat{\mathbf{v}} - \mathbf{v}_1\| + \|\mathbf{v}_1 - \tilde{\mathbf{v}}\| \le \mu \|\tilde{\mathbf{v}} - \hat{\mathbf{v}}\| + \|\mathbf{v}_1 - \tilde{\mathbf{v}}\|.$$

Using the identity $\mathbf{f}_{\mathbf{v}}(\tilde{\mathbf{u}}, \tilde{\mathbf{v}})^\dagger \mathbf{f}(\tilde{\mathbf{u}}, \tilde{\mathbf{v}}) = \mathbf{0}$, the Lipschitz continuity of $\mathbf{f}$ and $\mathbf{f}_{\mathbf{v}}$ along with

$$\begin{aligned}&\|\mathbf{f}_{\mathbf{v}}(\hat{\mathbf{u}}, \tilde{\mathbf{v}})^\dagger - \mathbf{f}_{\mathbf{v}}(\tilde{\mathbf{u}}, \tilde{\mathbf{v}})^\dagger\| \\ &\le 3\,\|\mathbf{f}_{\mathbf{v}}(\tilde{\mathbf{u}}, \tilde{\mathbf{v}})^\dagger\|^2\, \|\mathbf{f}_{\mathbf{v}}(\hat{\mathbf{u}}, \tilde{\mathbf{v}}) - \mathbf{f}_{\mathbf{v}}(\tilde{\mathbf{u}}, \tilde{\mathbf{v}})\| \quad \text{(c.f. [20, Theorem 3.4])} \\ &\le 3\,\|\mathbf{f}_{\mathbf{v}}(\tilde{\mathbf{u}}, \tilde{\mathbf{v}})^\dagger\|^2\, \|(\mathbf{f}_{\mathbf{v}})_{\mathbf{u}}(\tilde{\mathbf{u}}, \tilde{\mathbf{v}})\|\, \|\hat{\mathbf{u}} - \tilde{\mathbf{u}}\| + O(\|\hat{\mathbf{u}} - \tilde{\mathbf{u}}\|^2)\end{aligned}$$

for sufficiently small $\|\hat{\mathbf{u}} - \tilde{\mathbf{u}}\|$ where $(\mathbf{f}_{\mathbf{v}})_{\mathbf{u}}(\tilde{\mathbf{u}}, \tilde{\mathbf{v}})$ is the Jacobian of the holomorphic mapping $\mathbf{u} \mapsto \mathbf{f}_{\mathbf{v}}(\mathbf{u}, \tilde{\mathbf{v}})$ at $\tilde{\mathbf{u}}$, we have

$$\begin{aligned}\|\tilde{\mathbf{v}} - \hat{\mathbf{v}}\| &\le \tfrac{1}{1-\mu}\, \|\mathbf{v}_1 - \tilde{\mathbf{v}}\| \\ &= \tfrac{1}{1-\mu}\, \|\mathbf{f}_{\mathbf{v}}(\hat{\mathbf{u}}, \tilde{\mathbf{v}})^\dagger\, \mathbf{f}(\hat{\mathbf{u}}, \tilde{\mathbf{v}}) - \mathbf{f}_{\mathbf{v}}(\tilde{\mathbf{u}}, \tilde{\mathbf{v}})^\dagger\, \mathbf{f}(\tilde{\mathbf{u}}, \tilde{\mathbf{v}})\| \\ &\le \tfrac{1}{1-\mu} \big(\|\mathbf{f}_{\mathbf{v}}(\tilde{\mathbf{u}}, \tilde{\mathbf{v}})^\dagger\|\, \|\mathbf{f}(\hat{\mathbf{u}}, \tilde{\mathbf{v}}) - \mathbf{f}(\tilde{\mathbf{u}}, \tilde{\mathbf{v}})\| \\ &\quad + \|\mathbf{f}_{\mathbf{v}}(\hat{\mathbf{u}}, \tilde{\mathbf{v}})^\dagger - \mathbf{f}_{\mathbf{v}}(\tilde{\mathbf{u}}, \tilde{\mathbf{v}})^\dagger\|\, \|\mathbf{f}(\hat{\mathbf{u}}, \tilde{\mathbf{v}})\| \big) \\ &\le \tfrac{\|\mathbf{f}_{\mathbf{v}}(\tilde{\mathbf{u}}, \tilde{\mathbf{v}})^\dagger\|}{1-\mu} \big(\|\mathbf{f}_{\mathbf{u}}(\tilde{\mathbf{u}}, \tilde{\mathbf{v}})\| + 3\,\|\mathbf{f}_{\mathbf{v}}(\tilde{\mathbf{u}}, \tilde{\mathbf{v}})^\dagger\|\, \|(\mathbf{f}_{\mathbf{v}})_{\mathbf{u}}(\tilde{\mathbf{u}}, \tilde{\mathbf{v}})\|\, \|\mathbf{f}(\tilde{\mathbf{u}}, \tilde{\mathbf{v}}\| \big) \\ &\quad \times \|\hat{\mathbf{u}} - \tilde{\mathbf{u}}\| + O(\|\hat{\mathbf{u}} - \tilde{\mathbf{u}}\|^2). \qquad (13)\end{aligned}$$

As a result, there is a constant $\theta > 0$ such that $\|\tilde{\mathbf{v}} - \hat{\mathbf{v}}\| \le \theta \|\tilde{\mathbf{u}} - \hat{\mathbf{u}}\|$ when $\|\tilde{\mathbf{u}} - \hat{\mathbf{u}}\|$ is sufficiently small, leading to the assertion (ii). Set $(\tilde{\mathbf{u}}, \tilde{\mathbf{v}}) = (\mathbf{u}_0, \mathbf{v}_0)$ and $(\hat{\mathbf{u}}, \hat{\mathbf{v}}) = (\mathbf{u}_0 + \Delta\mathbf{u}, \mathbf{v}_0 + \Delta\mathbf{v})$ in (13) and apply $\mathbf{f}(\mathbf{u}_0, \mathbf{v}_0) = \mathbf{0}$ and $\mu = O(\|\hat{\mathbf{u}} - \tilde{\mathbf{u}}\|)$. The inequality (12) holds. □

Based on Lemma 6.1, the following Theorem 6.2 is a version of the Tubular Neighborhood Theorem for manifolds in normed vector spaces isometrically isomorphic to $\mathbb{C}^n$'s. It is specifically tailored for the application of solving ill-posed algebraic problems from empirical data. There appears to be no such a version in the literature of differential geometry since some analytic structures can not be preserved in the tubular neighborhood and not needed in our application. We provide a proof based on the Gauss-Newton iteration and Lemma 6.1.

Theorem 6.2 (Tubular Neighborhood Theorem). *Let Π be a complex analytic manifold in a vector space $\mathcal{U}$ that is isometrically isomorphic to $\mathbb{C}^n$. There is a* tubular neighborhood, *namely an open subset $\Omega \supset \Pi$ of $\mathcal{U}$ such that every $\mathbf{b} \in \Omega$ has a unique projection $\mathbf{x}_\mathbf{b} \in \Pi$ of minimum distance to $\mathbf{b}$, that is*

$$\big\|\mathbf{x}_\mathbf{b} - \mathbf{b}\big\| = \inf_{\mathbf{x}\in\Pi} \big\|\mathbf{x} - \mathbf{b}\big\| =: \operatorname{dist}(\mathbf{b}, \Pi). \qquad (14)$$

Furthermore, the projection $\mathbf{b} \mapsto \mathbf{x}_\mathbf{b}$ from Ω to Π is locally Lipschitz continuous.

Proof. Let $\mathbf{u}_0$ be any particular point in Π. Since Π is a complex analytic manifold in $\mathcal{U}$, there is an open neighborhood $\mathcal{M}$ of $\mathbf{u}_0$ in $\mathcal{U}$, an open subset $\mathcal{N}$ of $\mathbb{C}^m$ and a holomorphic mapping $\mathbf{v} \mapsto \phi(\mathbf{v})$ from $\mathcal{N} \subset \mathbb{C}^m$ onto $\mathcal{M} \cap \Pi$ with a holomorphic inverse ϕ^{-1} from $\mathcal{M} \cap \Pi$ onto $\mathcal{N}$. Let the holomorphic mapping $\mathbf{f} : (\mathbf{u}, \mathbf{v}) \mapsto \phi(\mathbf{v}) - \mathbf{u}$ from $\mathcal{M} \times \mathcal{N} \subset \mathcal{U} \times \mathbb{C}^m$ to $\mathcal{U}$. Then $\mathbf{f}$ satisfies all the conditions of Lemma 6.1. As a result, there is an open neighborhood $\Psi \subset \mathcal{M}$ of $\mathbf{u}_0$ in $\mathcal{U}$ such that, for every $\hat{\mathbf{u}} \in \Psi$, there exists a unique least squares solution $\mathbf{v} = \hat{\mathbf{v}}$ for the equation $\mathbf{f}(\hat{\mathbf{u}}, \mathbf{v}) = \mathbf{0}$ so that

$$\|\mathbf{f}(\hat{\mathbf{u}}, \hat{\mathbf{v}})\| = \min_{\mathbf{v}\in\Phi} \|\mathbf{f}(\hat{\mathbf{u}}, \mathbf{v})\| = \min_{\mathbf{v}\in\Phi} \|\phi(\mathbf{v}) - \hat{\mathbf{u}}\|.$$

We can assume Ψ is sufficiently small so that any $\hat{\mathbf{u}} \in \Psi$ satisfies $\|\hat{\mathbf{u}} - \mathbf{u}\| > \|\phi(\hat{\mathbf{v}}) - \hat{\mathbf{u}}\|$ for all $\mathbf{u} \in \Pi \setminus \phi(\Phi)$, implying the local minimum $\|\phi(\hat{\mathbf{v}}) - \hat{\mathbf{u}}\| = \min_{\mathbf{u}\in\Pi} \|\mathbf{u} - \hat{\mathbf{u}}\|$ is the global minimum. □

From computational point of view, the desired solution $\hat{\mathbf{v}} \in \mathcal{V}$ at a data point $\hat{\mathbf{u}} \in \mathcal{U}$ may be modeled as the zero of a holomorphic mapping $\mathbf{v} \mapsto \mathbf{f}(\hat{\mathbf{u}}, \mathbf{v})$ with $\hat{\mathbf{u}}$ in a structure-preserving manifold Π in $\mathcal{U}$. When $\hat{\mathbf{u}}$ is not known exactly but represented by its empirical data in $\tilde{\mathbf{u}} \approx \hat{\mathbf{u}}$, the Tubular Neighborhood Theorem ensures the projection $\check{\mathbf{u}}$ of $\tilde{\mathbf{u}}$ to Π uniquely exists, enjoys Lipschitz continuity and is independent of choices of the mapping $\mathbf{f}$ in the model. As a result, the solution $\check{\mathbf{v}}$ at the parameter value $\check{\mathbf{u}}$ can be defined as the *regularized solution* at $\tilde{\mathbf{u}}$. From Lemma 6.1, the regularized solution $\check{\mathbf{v}}$ can be accurately approximated by the least squares solution $\tilde{\mathbf{v}}$ of the equation $\mathbf{f}(\tilde{\mathbf{u}}, \mathbf{v}) = \mathbf{0}$ as long as $\mathbf{f}$ is properly constructed following the Geometric Modeling Theorem.

7 THE GEOMETRIC REGULARIZATION: CONCLUDING REMARKS AND EXAMPLES

As a notion attributed to Hadamard, a mathematical model is a *well-posed problem* if its solution satisfies existence, uniqueness and Lipschitz continuity with respect to data perturbations. Those problems may also be loosely referred to as being *regular*. Otherwise, the problem is ill-posed or often called *singular*. In general, singular problems are difficult to solve accurately from empirical data and require some form of regularization.

Algebraic problems such as the polynomial GCD/factorization, the matrix rank/kernel and the matrix Jordan Canonical Form (JCF) are not all singular. For each problem, the data space is partitioned by manifolds and, on every manifold, the solutions maintains a specific algebraic structure. Data associated with regular problems are open dense in the data space, forming a manifold of codimension zero. A problems is singular when the data point lies on a manifold

of a positive codimension. Due to the dimension deficit, a perturbation generically pushes the data away from the native manifold and the alters the structure of the solution, implying the solution is highly sensitive to *arbitrary* data perturbations.

However, the solutions of those singular problems are locally Lipschitz continuous if the data are constrained on a structure-preserving manifold. By the Geometric Modeling Theorem, algebraic problem on any such manifold Π can be modeled as a zero finding problem $\mathbf{f}(\mathbf{u}, \mathbf{v}) = \mathbf{0}$ for the variable $\mathbf{v}$ at $\mathbf{u} \in \Pi$. If an underlying data point $\hat{\mathbf{u}}$ is known with limited accuracy through empirical data $\tilde{\mathbf{u}}$, Lemma 6.1 and the Tubular Neighborhood Theorem (Theorem 6.2) ensure that solving for the least squares solution $\tilde{\mathbf{v}}$ of the equation $\mathbf{f}(\tilde{\mathbf{u}}, \mathbf{v}) = \mathbf{0}$ is a well-posed problem. In other words, a singular algebraic problem can be regularized if it can be properly modeled following the Geometric Modeling Theorem assuming the structure of the solution is known.

Detailed elaboration on the identification of the solution structure is beyond the scope of this paper. In a nutshell, we can quantify the *singularity* of each data point as the codimension of the manifold on which the data point resides. The structure-preserving manifolds are entangled to form a strata in which every manifold is embedded in the closures of some manifolds of lower codimensions. In other words, such an algebraic problem is highly sensitive but the sensitivity is *directional* such that sufficiently small perturbations can only reduce the singularity and never increase it.

At an underlying data point $\hat{\mathbf{u}}$ on a structure-preserving manifold Π, the given empirical data point $\tilde{\mathbf{u}}$ is a small perturbation from $\hat{\mathbf{u}}$. Assuming the perturbation is sufficiently small so that $\tilde{\mathbf{u}}$ stays in the tubular neighborhood, the underlying manifold Π is of the highest singularity (codimension) among all the manifolds intersect a small neighborhood of $\tilde{\mathbf{u}}$. Identification of the solution structure becomes a discrete optimization problem in maximizing the codimension (singularity) of the manifolds within an error tolerance of the empirical data point $\tilde{\mathbf{u}}$. Consequently, a natural strategy for computing the regularized solution at an empirical data parameter $\tilde{\mathbf{u}}$ is a two-staged process in either symbolic, numerical or hybrid computation:

- **Stage I**. Within an error tolerance of the data $\tilde{\mathbf{u}}$, find the nearby structure-preserving manifold of the highest singularity.
- **Stage II**. Solve the equation $\mathbf{f}(\tilde{\mathbf{u}}, \mathbf{v}) = \mathbf{0}$ that is properly formulated based on the Geometric Modeling Theorem for its least squares solution $\mathbf{v} = \tilde{\mathbf{v}}$.

The least squares solution $\tilde{\mathbf{v}}$ is a *regularized solution* at the empirical data $\tilde{\mathbf{u}}$ *within the error tolerance*. It is *not* a solution at $\tilde{\mathbf{u}}$ in conventioanl sense but accurately approximates the exact solution at the underlying data $\hat{\mathbf{u}}$ with an error bound proportional to the data error $\|\tilde{\mathbf{u}} - \hat{\mathbf{u}}\|$.

This regularization strategy has been applied to many singular algebraic problems such as computing multiple roots of univariate polynomials [25], approximate polynomial GCD [26, 30], factorization of multivariate polynomials [23] from empirical data. The resulting algorithms are implemented in the MATLAB package NAClab [31] including a preliminary module for computing the Jordan Canonical Form from possibly perturbed matrices. We illustrate the strategy the following examples.

Example 7.1. Let the polynomial pair $(p, q) \in \mathbb{P}_{13} \times \mathbb{P}_{11}$ where

$$\tilde{p} = 1 - 0.333x + 0.667x^3 + x^{10} - 0.333x^{11} + 0.666x^{13}$$
$$\tilde{q} = 1.429 + 3.571x + 1.429x^{10} + 3.571x^{11}$$

serving as empirical data of the pair (p, q) that equals

$$\left(1 - \tfrac{1}{3}x + \tfrac{2}{3}x^3 + x^{10} - \tfrac{1}{3}x^{11} + \tfrac{2}{3}x^{13},\ \tfrac{10}{7} + \tfrac{25}{7}x + \tfrac{10}{7}x^{10} - \tfrac{25}{7}x^{11}\right).$$

In exact sense, we have $\gcd(p, q) = 1+x^{10}$ but $\gcd(\tilde{p}, \tilde{q}) = 1$ that are far apart due to the singularity of the GCD even though the data error is about 10^{-3}. The computing objective is to find an approximate GCD $\approx 1 + x^{10}$ from the empirical data $(\tilde{p}, \tilde{q})$ by calculating a regularized GCD within the data error bound 10^{-3}.

At Stage I, the GCD degree (i.e. structure) is identified by computing the numerical nullity of the Sylvester matrix $S(\tilde{p}, \tilde{q})$ within the error tolerance 10^{-3}. This numerical nullity is identical to the degree 10 of $\gcd(p, q)$. Therefore, the native GCD manifold is $\mathcal{P}^{10}_{13,11}$. Further more, initial approximation (u_0, v_0, w_0) of the GCD and cofactors can be obtained by solving two linear systems (c.f. [26]).

At Stage II, we formulate the geometric model by constructing the mapping ψ as in (7) for $k = 10$, $m = 13$, $n = 11$ and solve the equation $\psi(u, v, w, \tilde{p}, \tilde{q}) = (0, 0, 0)$ for the least squares solution $(u, v, w) \in \mathbb{P}_2 \times \mathbb{P}_{11} \times \mathbb{P}_9$ with $u \sim 1 + 0.9998x^{10}$ with an accuracy in the order of the data error bound. The regularized GCD computation is implemented in NAClab so that the above computation can be carried out in simple MATLAB sequence:

```
>> p = '1-.333*x+0.667*x^3+x^10-0.333*x^11 + 0.666*x^13'; % enter polynomial p
>> q = '-1.429 - 3.571*x - 1.429*x^10 - 3.571*x^11';      % enter polynomial q
>> pgcd(p, q, 0.001) % regularized GCD of p and q within error tolerance 0.001
ans =
    '-1.24459473398662 - 1.24432753501985*x^10'
```

The result is a multiple of $1 + 0.9998x^{10}$.

Computing Jordan Canonical Forms of matrices from empirical data is known to be a tremendous challenge. We conclude this paper with two examples of the module `RegularizedJCF` in NAClab based on the geometric modeling elaborated in this paper.

Example 7.2. There are applications where empirical data may even be preferred over exact ones. Consider the matrix

$$A(r, s, t) = \begin{bmatrix} 2r\text{-}2s\text{+}t & 1\text{-}s\text{+}t & r\text{-}3\text{-}3s\text{+}2t & r\text{-}1\text{-}2s\text{+}t & \text{-}1 & \text{-}1\text{-}s\text{+}t \\ r\text{+}4\text{+}s\text{-}2t & 3r\text{+}2\text{-}2t & 2r\text{+}10\text{+}2s\text{-}4t & r\text{+}5\text{+}s\text{-}2t & \text{-}r\text{+}s & r\text{+}3\text{+}s\text{-}2t \\ 1\text{+}4r\text{-}3s\text{-}t & 1\text{+}3r\text{-}2s\text{-}t & 1\text{+}7r\text{-}4s\text{-}2t & 1\text{+}4r\text{-}3s\text{-}t & \text{-}r\text{-}1\text{+}s & 2r\text{-}s\text{-}t \\ 7s\text{-}t\text{-}6r\text{-}2 & 4s\text{-}t\text{-}3r\text{-}3 & 10s\text{-}2t\text{-}8r\text{+}1 & 7s\text{-}t\text{-}5r\text{-}1 & 3\text{+}r\text{-}s & 3s\text{-}t\text{-}2r\text{+}1 \\ r\text{+}3\text{+}2s\text{-}3t & 3r\text{+}1\text{-}3t & 2r\text{+}9\text{+}4s\text{-}6t & r\text{+}4\text{+}2s\text{-}3\ t & 1\text{-}r\text{+}2s & r\text{+}3\text{+}2s\text{-}3t \\ \text{-}5r\text{-}4\text{+}5t & s\text{+}5t\text{-}6r\text{-}2 & 10t\text{-}9r\text{-}10\text{-}s & \text{-}5r\text{-}5\text{+}5t & 2r\text{-}2s & 5t\ \text{-}3r\text{-}3\text{-}s \end{bmatrix}$$

whose JCF is known to be $J_3(r) \oplus J_2(s) \oplus J_1(t)$. When the parameter values r, s and t are exact, say $\sqrt{k + \sqrt{k + \sqrt{k}}}$ for $k = 2, 3, 5$, test on Maple 17 could not finish after hours of computation. We can instead use approximation by rounding to, say 5 digits after decimal and find the regularized JCF within the error tolerance 10^{-4}. The following is a demo of using `RegularizedJCF` in NAClab that takes negligible elapsed time 0.03 second.

```
>> A = [ 214636   149815  -231707   -81521  -100000   -50185 % enter matrix data
         269034   233854   738068   369034    31336   169034
         -75161   -43824     8509   -75161   -68664  -112488
        -061796  -255806   251061   234361   268664   112858
         119219  -143454   538438   219219   358830   119219
           5757   237093  -219823   -94243   -62673   270577]/100000;
>> [J,X] = RegularizedJCF(A,1e-4);   % call the software module
>> single(J)                         % display JCF in single precision
ans =
   1.9615549   0.4031104           0           0           0           0
           0   1.9615549   3.7739313           0           0           0
           0           0   1.9615549           0           0           0
           0           0           0   2.2749500  -1.2751906           0
           0           0           0           0   2.2749500           0
           0           0           0           0           0   2.7730999
```

obtaining the exact JCF structure and eigenvalues of an accuracy that is moderately proportional to that of the data.

Example 7.3. Godunov [13, page 10] uses the matrix

$$G = \begin{pmatrix} 289 & 2064 & 336 & 128 & 80 & 32 & 16 \\ 1152 & 30 & 1312 & 512 & 288 & 128 & 32 \\ -29 & -2000 & 756 & 384 & 1008 & 224 & 48 \\ 512 & 128 & 640 & 0 & 640 & 512 & 128 \\ 1053 & 2256 & -504 & -384 & -756 & 800 & 208 \\ -287 & -16 & 1712 & -128 & 1968 & -30 & 2032 \\ -2176 & -287 & -1565 & -512 & -541 & -1152 & -289 \end{pmatrix}$$

to illustrate the difficulties in computing eigenvalues. The eigenvalues $0, \pm 1, \pm 2, \pm 4$ of G are simple but extremely sensitive with condition numbers arround 4×10^{12}, implying G is a small perturbation from a matrix on a singular bundle. Trying an error tolerance, say 10^{-9}, the module `RegularizedJCF` in NAClab finds the regularized JCF of G within 10^{-9} as a direct sum of of a 4×4 and a 3×3 elementary Jordan blocks

$$\tilde{J} = J_4(-2.121366210414752) \oplus J_3(2.828488280553040).$$

This is the JCF of a nearby matrix $\hat{G}$ of singularity 5 with a distance $\|G - \hat{G}\|_F / \|G\|_F \approx 3.13 \times 10^{-10}$. The condition number of the JCF of $\hat{G}$ is much smaller at 3.6×10^6. There is another nearby bundle of even higher singularity. Seting an error tolerance, say 0.005, the regularized JCF of G becomes a single 7×7 elementary Jordan block $J_7(0.000000000001459)$. with a moderate condition number 4268.5. In 9-digit integer representation, there is a matrix $\tilde{G} = XJX^{-1}$ with an exact eigenvalue zero in a 7×7 elementary Jordan block and a relative distance $\|G - \tilde{G}\|_F / \|G\|_F \approx 5.3 \times 10^{-7}$ where

$$X = \begin{bmatrix} -500000494 & 499231619 & 475440501 & 550430216 & 249476819 & 344543097 & 244097 \\ 244296 & 39690036 & 305346098 & 418811015 & 245808894 & 229491349 & 499999756 \\ 499998993 & -499191461 & -425110651 & -12743120 & -32442421 & -25936159 & 500000266 \\ 406 & 240 & -293020 & 209706406 & 260079479 & 212082338 & -33 \\ -499998969 & 499191323 & 425111477 & 11515088 & 494926230 & 166312083 & 500000239 \\ 500001425 & -499232519 & -475441212 & -550432966 & -249025474 & 786194594 & 244019 \\ -244271 & -39689958 & -305347057 & -417583165 & -708293582 & -370419391 & 499999620 \end{bmatrix}$$

$$J = \frac{1}{10^6} \begin{bmatrix} 0 & -163589092 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1279307109 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 2151028721 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 113025963 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 2502078868 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 3622414612 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

We now have a geometric interpretation on the sensitivity of G in an eigenproblem. Let Π_7, $\Pi_{4\,3}$ and $\Pi_{1111111}$ be bundles corresponding to Jordan structures $J_7(\lambda)$, $J_4(\lambda_1) \oplus J_3(\lambda_2)$ and $J_1(\lambda_1) \oplus \cdots \oplus J_1(\lambda_7)$ respectively. The bundle Π_7 is of the highest singularity (i.e. codimension) 6 among all bundels in $\mathbb{C}^{7\times 7}$ and is embedded in the closure of $\Pi_{4\,3}$ with a lower singularity 5 while both bundles are in the closure of the open dense bundle $\Pi_{1111111}$ of singularity zero. Although $G \in \Pi_{1111111}$ is regular, its eigenproblem is highly ill-conditioned because G is a tiny distance 10^{-10} from the bundle $\Pi_{4\,3}$ of singularity 5 and 10^{-7} from the most singular bundle Π_7. With proper geometric modeling, the regularized JCF problem of G is not as ill-conditioned as the straightforward eigenproblem.

8 ACKNOWLEDGMENTS

The author is indebted to his former colleague Marian Gidea for introducing the Tubular Neighborhood Theorem in a conversation leading to this work.

REFERENCES

[1] P.-A. Absil, R. Mahony, and R. Sepulchre. 2008. *Optimization Algorithms on Matrix Manifolds.* Princeton University Press, Princeton and Oxford.

[2] V. I. Arnold. 1971. On matrices depending on parameters. *Russian Math. Surveys* 26 (1971), 29–43.

[3] Stephen Barnett and R. G. Cameron. 1985. *Introduction to Mathematical Control Theory.* Oxford University Press, New York.

[4] Keith Burns and Marian Gidea. 2005. *Differential Geometry and Topology: with a view to dynamical systems.* CRC Press.

[5] Robert M. Corless, A. Galligo, I.S. Kotsireas, and S.M. Watt. 2002. A geometric-numeric algorithm for factoring multivariate polynomials. (2002). Proc. ISSAC'02, ACM Press, pages 37-45.

[6] Jean-Pierre Dedieu. 1996. Approxiate solutions of Numerical Problems, condition number analysis and condition number theorems. In *The Mathematics of Numerical Analysis, Lectures in Applied Math., 32.* Amer. Math.Soc., 263–283.

[7] James W. Demmel and A. Edelman. 1995. The dimension of matrices (matrix pencils) with given Jordan (Kronecker) Canonical Forms. *Linear Alg. and its Appl.* 230 (1995), 61–87.

[8] Jean Alexandre Dieudonné. 1989. *A History of Algebraic and Differential Topology, 1900-1960.* Birkhäuser, Boston.

[9] A. Edelman, T. A. Arias, and S. T. Smith. 1998. The geometry of algorithms with orthogonality constraints. *SIAM J. Matrix Anal. Appl.* 20 (1998), 303–353.

[10] Alan Edelman, Erik Elmroth, and Bo Kågström. 1997. A geometric approach to perturbation theory of matrices and and matrix pencils. Part I: Versal deformations. *SIAM J. Matrix Anal. Appl.* 18 (1997), 653–692.

[11] Alan Edelman, Erik Elmroth, and Bo Kågström. 1999. A geometric approach to perturbation theory of matrices and and matrix pencils. Part II: a stratification-enhanced staircase algorithm. *SIAM J. Matrix Anal. Appl.* 20 (1999), 667–699.

[12] C. G. Gibson. 1976. Regularity of the Segre stratification. *Math. Proc. Cambridge Phil. Soc* 80 (1976), 91–97.

[13] S. K. Godunov. 1998. *Modern Aspects of Linear Algebra.* Translations of Math. Monographs, v. 175, Amer. Math. Soc., Providence, RI.

[14] R. A. Horn and C. R. Johnson. 1985. *Matrix Analysis.* Cambridge University Press.

[15] W. Kahan. 1972. Conserving Confluence Curbs Ill-Condition. (1972). Technical Report 6, Computer Science, University of California, Berkeley.

[16] E. Kaltofen, Z. Yang, and L. Zhi. 2006. Approximate greatest common divisor of several polynomials with linearly constrained coefficients and singular polynomials. (2006). Proc. ISSAC'06, ACM Press, pp 169–176.

[17] Carl D. Meyer. 2000. *Matrix Analysis and Applied Linear Algebra.* SIAM, Philadelphia.

[18] Cleve Moler and Charles Van Loan. 2003. Nineteen dubious ways to compute the exponential of a matrix, twenty-five years later. *SIAM Review* 45 (2003), 3–49.

[19] A. J. Sommese, J. Verschelde, and C. W. Wampler. 2005. Introduction to numerical algebraic geometry. In *Solving Polynomial Equations*, A. Dickenstein and Ioannis Z. Emiris (Eds.). Springer-Verlag Berlin Heidelberg, 301–337.

[20] G. W. Stewart. 1977. On the perturbation of pseudo-inverses, projections, and linear least squares problems. *SIAM Review* 19 (1977), 634–662.

[21] J.L. Taylor. 2000. *Several Complex Variables with Connections to Algebraic Geometry and Lie Groups.* Amer. Math. Soc., Providence, Rhode Island.

[22] C. W. Ueberhuber. 1997. *Numerical Computation 2.* Springer-Verlag, Berlin, Heidelberg, New York.

[23] Wenyuan Wu and Zhonggang Zeng. 2017. The numerical factorization of polynomials. *J. Foundation of Computational Mathematics* 17 (2017), 259–286.

[24] Zhonggang Zeng. 2005. Computing multiple roots of inexact polynomials. *Math. Comp.* 74 (2005), 869–903. DOI. 10.1090/S0025-5718-04-01692-8.

[25] Zhonggang Zeng. 2009. The approximate irreducible factorization of a univariate polynomial. Revisited. (2009). Proc. of ISSAC '09, ACM Press, pp. 367–374.

[26] Zhonggang Zeng. 2011. The numerical greatest common divisor of univariate polynomials. In *Contemporary Mathematics Vol. 556, Amer. Math. Society,* Randomization, Relaxation and Complexity in Polynomial Equation Solving, J.M. Rojas L. Gurvits, P. Pébay and D. Thompson (Eds.). Providence, RI, 187–217.

[27] Zhonggang Zeng. 2016. Sensitivity and computation of a defective eigenvalue. *SIAM J. Matrix Analysis and Applications* 37, 2 (2016), 798–817. DOI. 10.1137/15M1016266.

[28] Zhonggang Zeng. 2018. Intuitive interface for solving linear and nonlinear system of equations. In *Mathematical Software — ICMS 2018 (LNCS 10931),* J. H. Davenport, M. Kauers, G. Labahn, and J. Urban (Eds.). Springer International AG, 495–506.

[29] Zhonggang Zeng. 2019. On the sensitivity of singular and ill-conditioned linear systems. *SIAM J. Matrix Anal. Appl.* 40, 3 (2019), 918–942. DOI. 10.1137/18M1197990.

[30] Zhonggang Zeng and B.H. Dayton. 2004. The approximate GCD of inexact polynomials. II: A multivariate algorithm. (2004). Proceedings of ISSAC'04, ACM Press, pp 320-327.

[31] Zhonggang Zeng and Tien-Yien Li. 2013. NAClab: A Matlab toolbox for numerical algebraic computation. *ACM Communications in Computer Algebra* 47 (2013), 170–173. `http://homepages.neiu.edu/~naclab.`

Author Index