MITACS SHORT Project CV and Business Plan

September 1, 2010

Title of Project:

Mathematics of Computer Algebra and Analysis
(MOCAA)

Project leaders:

George Labahn, School of Computer Science, University of Waterloo.
Michael Monagan, Department of Mathematics, Simon Fraser University.

Project website:

`www.cecm.sfu.ca/~pborwein/MITACS/index.htm`

1 Project Investigators

Changes over period :

Keith Geddes (Waterloo) retired.
Jon Borwein (Dalhousie) moved to The University of Newcastle, Newcastle, Australia but remains adjunct at Dalhousie.
Vahid Dabbaghian (Simon Fraser) joined the project as an adjunct professor.

Project core investigators:

Simon Fraser University:
Peter Borwein, Mathematics
Petr Lisonek, Mathematics
Marnie Mishna, Mathematics
Michael Monagan, Mathematics

University of Waterloo:
Mark Giesbrecht, Computing Science
George Labahn, Computing Science
Arne Storjohann, Computing Science

University of Western Ontario:
Robert Corless, Applied Mathematics
David Jeffrey, Applied Mathematics
Marc Moreno-Maza, Computer Science
Greg Reid, Applied Mathematics
Eric Schost, Computer Science
Stephen Watt, Computer Science

University of Calgary:
Wayne Eberly, Computer Science

McMaster University
Jacques Carette, Computer and Software

University of Lethbridge:
Howard Cheng, Mathematics and Computer Science

Project collaborators:

Francois Bergeron, University of Quebec at Montreal
Jon Borwein, Adjunct member at Dalhousie University
Vahid Dabbaghian, Simon Fraser University
Ilias Kotsirias, Wilfred Laurier University

1.1 Non-Academic Partners

No change in period.

Organization Name: Maplesoft Inc.
Name of Primary Contacts: Laurent Bernardin, Jürgen Gerhard.
Mailing Address: Maplesoft, 615 Kumpf Drive, Waterloo, Ontario, N2V 1K8
Phone Number: (519) 747-2373
Fax Number: (519) 747-5284
E-mail Address: laurent@maplesoft.com, jgerhard@maplesoft.com
Web Page: `http://www.maplesoft.com`

2 Progress Report

Project 3.1 : High Performance Computer Algebra

Subproject 3.1.1. The modpn and sdmp libraries.

sdmp: In our previous MITACS project (2007–2009) we had developed the best available software for multiplication and division of sparse polynomials in one or more variables with integer coefficients. The goal of Subproject 3.1.1 was to try to design and implement fast *parallel* algorithms. We can report that this goal was essentially reached. In 2009 Monagan and Pearce [33] parallelized sparse polynomial multiplication and in 2010 [34] they succeeded in parallelizing sparse division which is much more difficult. The algorithms were implemented in C using P-threads. Remarkably, a super-linear speedup was achieved, typically a factor of 5 on a 4 core desktop.

modpn: In [38] Moreno Maza and Xie present efficient implementation strategies for dense polynomial multiplication targeting multi-cores. They show that *balanced input data* can maximize parallel speedup and minimize cache complexity for bivariate multiplication. However, unbalanced input data, which are common in symbolic computation, are challenging. The authors provide efficient techniques to reduce multivariate (and univariate) multiplication to *balanced bivariate multiplication*. Their implementation in `Cilk++` demonstrates good speedup (often super-linear) on multi-cores. At the heart of their software, the `modpn` [27] library relies on Fast Fourier Transforms which are now taking advantage of the auto-tuning techniques from the SPIRAL group [30].

Subproject 3.1.3. Transposition techniques

De Feo and Schost [11] have designed `transalpyne`, a small scripting language, that is specifically conceived for automatic transposition of linear functions. Its type system is able to automatically infer all the possible linear functions realized by a computer program. The key feature of `transalpyne` is its ability to transform a program computing a linear function in another computer program computing the transposed linear function. The time and space complexity of the resulting program are similar to the original ones. The result is a tool which is used to design asymptotically fast algorithms which can incorporate the transposition principle as an operation (in the same way that FFT is used).

Project 3.2 : Polynomial Algebra and Solvers

Subproject 3.2.1. Triangular decomposition and solvers

In [8] Chen, Davenport, May, Moreno Maza, Xia and Xiao present a new algorithm, called RealTriangularize, for solving any systems of polynomial equations, inequations and inequalities. Under genericity assumptions which are often met in practice, this algorithm runs in singly exponential time in terms of the number of variables. This breakthrough is confirmed experimentally: RealTriangularize outperforms all available software with similar specifications. We remark that RealTriangularize is currently being integrated into the `RegularChains` library in Maple with the assistance of J.P. May and J. Gerhard from Maplesoft.

Dahan, Moreno Maza, Poteaux and Schost [10] have generalized Kedlaya and Umans' modular composition algorithm to the multivariate case. As an application, they have obtained fast algorithms for operations modulo triangular sets (over a finite field), such as multiplication, inversion, equiprojectable decomposition or change of order. For the first time, they are able to exhibit running times that are almost linear for these operations, without any overhead exponential in the number of variables.

Subproject 3.2.3. Lacunary and space-efficient polynomial algebra

In the papers [12] and [13], Giesbrecht and Roche describe arithmetic with lacunary (or supersparse) polynomials. A C++/NTL implementation of the first work is available on Giesbrecht's website. Roche [41] examines the tradeoff between sparse and dense polynomial arithmetic, and through an amortized analysis demonstrates a hybrid algorithm providing a smooth gradient between the best algorithms in the two circumstances. The results in [40] and [20] demonstrate low-space and in-place asymptotically fast algorithms for polynomial arithmetic. These are effective for large sizes and when space is at a premium, such as hardware implementations for cryptography.

Project 3.3 : Symbolic Linear Algebra

Subproject 3.3.1. Fast linear algebra over number fields and function fields.

The central problem is how to interpolate a sparse polynomial $f(x_1, x_2, \ldots, x_n)$ (or rational function) with t non-zero terms modulo a prime p. One way to compare algorithms is to count the number of evaluations, called "probes", of f. For polynomials, Zippel's 1979 algorithm in [44] does $O(ndt)$ probes. Since it works one variable at a time, it has limited parallelism. The best result is that of Kaltofen, Lee and Lobo ([23]) in 2000 which does $O(nt)$ probes. However their approach is incremental which results in a completely sequential algorithm. In 2010 Monagan and Javadi [21]) found a new method which also does $O(nt)$ probes but which allows one to interpolate the degrees of each term in each variable in parallel – a breakthrough. Their parallel implementation in Cilk gave a respectable speedup of 3.2 on a 4 core machine. To improve on this, they are currently looking at how best to compute the roots of the so-called linear generator polynomial, a polynomial in $GF(p)[z]$ of degree t, since this is the sequential bottleneck. The next step is to develop a sparse rational function interpolation procedure based on this polynomial interpolation.

Subproject 3.3.2. Symbolic matrix analysis

Classes of matrices are often presented with symbolic dimensions using a mixture of terms and ellipsis symbols to describe their internal structure. While working with such classes of matrices is everyday mathematical practice, it has little support in computer algebra systems. Sexton, Sorge and Watt [42] describe an algebraic encoding of such matrices in terms of support functions and define the corresponding addition and multiplication algorithms. The encoding of these abstract matrices has the important property that they enable simple recovery of the structural properties. As a result they define arithmetic algorithms for abstract matrices as extensions of those for support function combinations using a normalizing term rewrite system.

Carette, Sexton and Watt [7] have investigated the use of multisets to express symbolic domain decompositions in an efficient, elegant and uniform way, simplifying both computation and reasoning. This has been applied to the arithmetic of symbolic matrices mentioned above with the result that certain operations may be reduced from exponential to linear complexity. A prototype implementation in Maple has been produced and an investigation is now underway to achieve a more complete and general implementation.

Subproject 3.3.3. Matrix normal forms of Ore matrices

In [14] Giesbrecht and Kim give the first polynomial-time algorithm for the computation of the Hermite form of a matrix of differential polynomials. The approach reduces the problem to one of linear solving over $\mathbb{Q}(t)$, the usual (commutative) rational functions. Significantly the cost is polynomial *both* in terms of its degree and coefficient growth.

Subproject 3.3.4. Applications of the outer product adjoint formula

Significant progress has been made in understanding the complexity of linear algebra problems over the domain of integers. For a *nonsingular* input matrix, the determinant and the solution of a linear system can be computed in a certified fashion (i.e., correctness of the result is guaranteed) in approximately the same time as multiplying two integer matrices of the same dimension and with the same size of entries as the input matrix. However, for problems on input matrices that are *singular* of unknown rank, a main roadblock to obtaining certified algorithms has been the lack of a certified algorithm for rank. In [43], Storjohann finally rectifies this situation by giving the first reduction to matrix multiplication for the integer rank certification problem. Rank certification is required to obtain fast certified algorithms for more complicated problems such as computing the null-space and Smith normal form, both of which reveal the rank of an input matrix.

Subproject 3.3.5. Vector rational reconstruction

In [45], Zhou and Labahn give a new algorithm for the effective computation of *order bases*. Order bases represent all possible solutions to matrix rational approximation problems (including for example vector rational reconstruction). The algorithm generalizes a previous method due to Storjohann and has optimal complexity for the class of problems where coefficient growth is not an issue. In the case where coefficient growth is an issue, a second, completely different, algorithm by Beckermann and Labahn [2] was developed. The new method reduces the bit-complexity cost of computing simultaneous Padé approximants for a vector of power series by a factor of m^3, where m is the size of the vector.

Project 3.4 : Additional projects in computer algebra

Subproject 3.4.3. Numeric and symbolic integration

Bailey, Borwein and Crandall [1] study the expected distance of a two-dimensional walk in the plane with n unit steps in random directions. While the statistics and large n behaviour are well understood, the precise behaviour of the first few steps is quite remarkable and less tractable. Series evaluations and recursions are obtained making it possible to explicitly determine this distance for small number of steps. The big breakthrough in this paper are the hypergeometric and elliptic hyper-closed form expressions for all the moments of a $2, 3$ or 4-step walk. What is particularly interesting in this paper is the heavy use of the Meijer G functions and tools for solving recurrence equations with polynomial coefficients, both of which play prominent roles in the integration of special functions in Maple.

Subproject 3.4.4. Application of the geometry of curves to handwriting analysis

We have developed a new geometric theory for recognizing mathematical symbols [18]. Characters are represented as parametric curves approximated by truncated orthogonal series, mapping symbols to low-dimensional vector space of series coefficients. Euclidean distance in this space is used to find similar symbols accurately and efficiently. Training data sets with hundreds of classes are seen to be almost linearly separable, allowing classification by ensembles of linear SVMs. The distance to separating planes provides a reliable confidence measure for classifications [17]. The series coefficients can be computed in real-time, as the symbol is being written [15] and orientation-independent recognition is achieved [16]. We are currently investigating how orthogonal series representations may be used to compress ink traces in a form that may allow recognition without decompression of the database. Preliminary work on this problem is reported in [29].

3 Research Plan (new projects for 2011 – 2012)

1 Fast sparse polynomial interpolation using discrete logarithms

In applications where one can choose the prime p, such as modular algorithms for polynomial GCD computation, we propose to investigate a new method to interpolate a sparse polynomial with t terms in n variables of degree d which would do only $O(t)$ probes (i.e. linear in t hence optimal) but would need to compute discrete logarithms. The idea is to choose the prime p such that $p-1 = q_1 \cdot q_2 \cdots q_n$ where $q_i > d$ and $\gcd(q_i, q_j) = 1$ and the q_i have no large prime divisors. [The method would then evaluate f at powers of $w_1, w_2, \ldots w_n$ where w_i are elements of $GF(p)$ of order q_i, i.e., of relatively prime order.] Computing the discrete logarithms is possible to do efficiently because of the way we have constructed p. The downside of this method is that it requires $p > (d+1)^n$ which is considerably larger than other methods. Therefore we propose to also develop a 63 bit prime library (Maple currently has a 31.5 bit prime library) of fast algorithms for arithmetic in $\mathrm{GF}(p)[z]$. This would be an excellent project for two students.

2 Triangular decomposition techniques for multi-homogeneous systems

We have developed efficient all-purpose solvers for polynomial systems. A new objective is to develop algorithms to handle structured polynomial systems. We plan to study how triangular decomposition techniques can be used to solve multi-homogeneous systems and systems with symmetries (such systems usually arise when modeling problems of a geometric nature). In both cases, the set of variables carries a natural block structure and we expect an improvement in computation time that would reflect this structure (through a multi-homogeneous Bézout number in the first case, or the size of the symmetry group in the second).

3 Triangular decomposition techniques for real algebraic geometry

Many real-world applications require exact computation with real algebraic numbers. Yet the available computer algebra packages are unable to efficiently test even simple properties such the inclusion of a real curve on a real surface. The objective of this subproject is to develop software tools for computing geometric and topological information for the real components of the solution sets of polynomial systems. We shall draw on recent work on cylindrical algebraic decomposition [9, 5] and the RealTriangularize Algorithm [8]. We will enhance the `RegularChains` Maple library with new functionality such as set-theoretical operations on semi-algebraic sets, computation of the real dimension of an algebraic set, projection of a semi-algebraic set and, more generally, quantifier elimination. This toolkit will allow the Maple user to attack challenging application problems.

4 Removing randomization from polynomial matrix arithmetic

A common feature of linear algebra problems on nonsingular matrices of polynomials with coefficients from a field K, such as system solving and row reduction, is that the most effective algorithms for them employ techniques that are deterministic, provided that a non-root $\gamma \in K$ of the determinant of the input matrix is provided. Currently, the only known method to find such a γ in the allotted time is to use randomization. Randomly choosing γ is highly effective when K is large enough, but for fields K that are too small, the usual solution of working over a sufficiently large extension field complicates implementation of the algorithms and can adversely affect running times. The goal of this project is to remove the need for randomization in recent fast techniques for linear algebra on polynomial matrices. Our approach is to factor an input matrix into the product of two matrices, such that 0 is not a root of one of the determinant of one of the factors, and 1 is not a root of the determinant of the other.

5 Matrix normal forms in non-commutative domains

Work will continue on Hermite and Smith normal forms over differential and difference algebras. In particular, we will focus on non-commutative coefficient fields, which are the natural domains when localizing PBW extensions and other rings of PDEs. Some of this work will be done in collaboration with Viktor Levandovskyy, who heads the non-commutative library within the Singular project, and his student Daniel Andres at Aachen. Major efforts will focus on management of coefficient size and efficient arithmetic and linear algebra over non-commutative fields.

6 Solving higher order differential systems via matrix normal forms

Current symbolic computation software for higher order linear differential systems converts equations to first order and then applies a number of known methods (e.g. super reducibility, Moser reduction, etc). Unfortunately this conversion to first order is both inefficient and also obscures the understanding of what is being transformed in order to arrive at a simpler system of equations. We plan on making use of normal forms of matrices of differential operators in order to develop efficient algorithms for solving such systems directly (both locally (i.e. power series) and globally (i.e. rational functions)) without the need to convert to first order.

7 Symbolic computation of convolution integrals

While computer algebra systems are quite good at producing closed form solutions of indefinite integrals the story is not so rosy with respect to definite integration, particularly improper definite integrals. In these cases the result often depends on the type of parameter values present in the integrands. We plan on continuing the work of Peasgood [39] in extending the Salvy approach to solving definite integrals of convolution type. Such integrals are prominent in many applications. For example all well known integral transforms can be represented as such integrals. We propose to extend the method to a much wider class of functions. This is particularly important as we wish the method to be both implemented in Maple and included in the new electronic version of the famous Handbook of Mathematical Functions.

8 Differential equations in automatic combinatorics

The goal of this research is to create tools for enumerative combinatorics using algebraic and symbolic computation methods, and mine the analytic nature of generating functions for additional combinatorial information [4]. The focus is on classes of objects whose generating functions satisfy systems of linear differential equations with polynomial coefficients, known as D-finite classes. Combinatorially, we try to predict which classes of objects have D-finite generating functions, as in [32]. Analytically, the emphasis is on finding symbolic techniques and manipulations to make the closure properties of D-finite functions effective.

A key family of objects under consideration is lattice paths with restricted steps. The papers [3, 31, 32] constitute an algorithmic approach to their exact enumeration using a technique called the kernel method. This has led to strongly supported conjectures on the nature of functional equations, arising in combinatorics, which have D-finite generating functions. The work of Bostan and Kauers uses symbolic computation methods, specifically the idea of controlled guess and verification to show properties of the generating function. Future work in this project will unite these approaches.

4 Significant Contributions

These five publications may be found on our project website under “Sample Papers”.

1 Bernhard Beckermann and **George Labahn**. Fraction-Free Computation of Simultaneous Pade Approximants, *Proceedings of ISSAC 2009*, Seoul, Korea, ACM Press, 15-22, 2009.

2 C. Chen, J. H. Davenport, J. M. May, **M. Moreno Maza**, B. Xia, and R. Xiao. Triangular Decomposition of Semi-Algebraic Systems. *Proc. of ISSAC 2010*, ACM Press, 187-194, 2010.

3 **Mark Giesbrecht** and Daniel S. Roche. Detecting lacunary perfect powers and computing their roots, To appear: Journal of Symbolic Computation.

4 Oleg Golubitsky and **Stephen M. Watt**. Distance-Based Classification of Handwritten Symbols. *International J. Document Analysis and Recognition*, 13(2) 133-146 2010.

5 Mahdi Javadi and **Michael Monagan**. Parallel Sparse Polynomial Interpolation over Finite Fields. *Proceedings of PASCO 2010*, ACM Press, 105–111, 2010.

4.1 Technology Transfer

Our industrial partner is Maplesoft whose primary product is the Maple software system. For our mitacs project, technology transfer mostly happens when we integrate our software into new releases of Maple and when we share new ideas with company personnel at lab meetings and conferences.

The sdmp library and the *Immediate monomials* subproject.

Our software for multiplication and division of polynomials with integer coefficients was integrated into Maple 14 which was released May 2010. It included our parallel multiplication from Monagan and Pearce in [33]. As an example, the following benchmark shows that Maple 14 is 50 times faster than Maple 13 and 7 times faster than Magma 2.16 for multiplying $f \times (f+1)$ with 1 core. With 4 cores this increases to 161 times and 24 times respectively. We are currently working to eliminate conversion overhead for Maple 15 which should result in about a 40% gain in speed on 4 cores.

	Maple 13	Maple 14 (1 core)	(4 cores)	Magma 2.16	Mathematica 7.0
$h = f \times (f+1)$	26.76s	0.53s	0.17s	4.09s	50.36s
factor(h)	391.44s	17.15s	15.59s	117.53s	164.49s

Timings in CPU seconds for $f = (1+x+y+z)^{30}+1$ on an Intel Core i7

Additional benchmarks and details about the integration into Maple 14 can be found in [35] on our project website. Currently we are integrating our parallel division from [34] into Maple.

Integration of the modpn library.

This library is dedicated to fast polynomial arithmetic based on Fast Fourier Transforms (FFTs) and Straight-Line Programs. The code consists of 36,000 lines of C code and 3,000 lines of Maple code. The experimental results reported in [27, 28] demonstrate that this library is competitive and often outperforms the available packages with similar functionality. The `modpn` library was integrated in Maple 13. For this work, each of the PhD students Xin Li and Wei Pan received a *MITACS Best Use of Mathematics in Technology Transfer Award.* Recently, the `modpn` library has been enhanced with GPU code (3,000 lines of CUDA code) in order to accelerate computations of FFTs and polynomial GCDs modulo regular chains [36]. This is also being integrated into Maple.

Mitacs Internships

Our mitacs project was awarded an internship cluster. All internships involve a significant amount of technology transfer. These include:

1 Jason Peasgood (PhD student (Labahn), Waterloo)
Symbolic Computation of Convolution Integrals

2 Changbo Chen (PhD student (Moreno-Maza), UWO)
Real Triangular Decomposition

3 Wei Pan (PhD student (Moreno-Maza), UWO)
Solving Polynomial Systems on a GPU

4 Andrew Arnold (MSc student (Monagan), SFU)
Parallel Integer Factorization

5 Niloofar Mani (MSc student (Reid), UWO)
Symbolic Numerical Algorithms for DAE

6 Alex Korobkine (PhD student (Carette), McMaster):
Automatic Verifiable Synthesis of Implementations from Mathematical Models.

7 Madhi Javadi (PhD student (Monagan), SFU):
Factoring Polynomials over Algebraic Number and Function Fields.

5 Proposed Budget

We request $130,000 for 2011–2012 from mitacs.
Our industrial partner, Maplesoft will contribute $65,000 cash + 25% university overhead.

Total available: $195,000.

We propose the following expenditures.

$120,000 for graduate student research assistantships
(for 10 graduate students)
$45,000 for a research associate (to oversee tech transfer)
$20,000 for travel expenses for students
$10,000 for 4 computers (workstations) for students.

Total expenses: $195,000.

References

[1] D.H. Bailey, **J.M. Borwein** and R.E. Crandall, Advances in the theory of box integrals, Mathematics of Computation, 79 (2010), 1839-1866.

[2] Bernhard Beckermann and **George Labahn**, Fraction-Free Computation of Simultaneous Pade Approximants, *Proceedings of ISSAC'09*, Seoul, Korea, ACM Press, 15-22, 2009.

[3] M. Bousquet-Melou and **Marnie Mishna**, Walks with small steps in the quarter plane, Algorithmic Probability and Combinatorics (special volume of the Contemporary Mathematics series of the Amer. Math. Soc.) 520, 2010, 1–40.

[4] M. Bouvel, C. Chauve, **Marnie Mishna** and D. Rossin, Average-case analysis of perfect sorting by reversals, CPM 09 20th Annual Symposium on Combinatorial Pattern Matching, Lille, France 2009.

[5] François Boulier, Changbo Chen, François Lemaire and **Marc Moreno Maza**, Real Root Isolation of Regular Chains, Proc. Asian Symposium on Computer Mathematics, 2009, 1–15.

[6] François Boulier, François Lemaire and **Marc Moreno Maza**, Computing differential characteristic sets by change of ordering, Journal of Symbolic Computation, 45(1):124–149, 2010.

[7] Jacques Carette, Alan P. Sexton, Volker Sorge and **Stephen M. Watt**. Symbolic Domain Decomposition. Proc. Calculemus 2010, 172–188, Springer Verlag LNAI 6167, 2010.

[8] Changbo Chen, James H. Davenport, John P. May, **Marc Moreno Maza**, Bican Xia, and Rong Xiao. Triangular decomposition of semi-algebraic systems. Proceedings of ISSAC'10, 187–194, 2010.

[9] Changbo Chen, **Marc Moreno Maza**, Bican Xia, and Lu Yang, Computing cylindrical algebraic decomposition via triangular decomposition. Proceedings of ISSAC'09, 95–102, 2009.

[10] Xavier Dahan, **Marc Moreno Maza**, Adrien Poteaux and **Éric Schost**, Almost-linear time operations with triangular sets, Poster at ISSAC'10. 2010.

[11] Luca De Feo and **Éric Schost**, *Transalpyne*: a language for automatic transposition, Communications of Computer Algebra, 44:1-2, 59-71, ACM Press, 2010.

[12] **Mark Giesbrecht** and Daniel S. Roche, Detecting lacunary perfect powers and computing their roots, To appear: Journal of Symbolic Computation, 2010.

[13] **Mark Giesbrecht** and Daniel S.. Roche, Interpolation of shifted-lacunary polynomials, To appear in Computational Complexity, 2010.

[14] **Mark Giesbrecht** and Myung Sub Kim, On computing the Hermite form of a matrix of differential polynomials, Computer Algebra and Scientific Computation (CASC) Workshop, Lecture Notes in Computer Science, 118-129, 2009

[15] Oleg Golubitsky and **Stephen M. Watt**, Online Computation of Similarity between Handwritten Characters. Proc. DRR 2009, pp. C1–C10 Vol. 7247, SPIE and IS&T, 2009.

[16] Oleg Golubitsky, Vadim Mazalov and **Stephen M. Watt.** Orientation-Independent Recognition of Handwritten Characters with Integral Invariants. Proc. ASCM 2009, 252–261, COE Lecture Note Vol. 22, Kyushu University, ISSN 1881-4042, 2009.

[17] Oleg Golubitsky and **Stephen M. Watt.** Confidence Measures in Recognizing Handwritten Mathematical Symbols. Proc. CICM 2009, 460–466, Springer Verlag LNAI 5625, 2009.

[18] Oleg Golubitsky and **Stephen M. Watt.** Distance-Based Classification of Handwritten Symbols. International J. Document Analysis and Recognition, Vol. 13, No. 2, pp. 133-146, June, 2010, Springer.

[19] Sardar Anisul Haque, Shahadat Hossain and **Marc Moreno Maza**. Cache friendly sparse matrix-vector multiplication.. Proceedings of PASCO'10. ACM Press, 176–175, 2010.

[20] David Harvey and Daniel S. Roche, An in-place truncated Fourier transform and applications to polynomial multiplication, Proceedings of ISSAC'09, 325-329, 2010

[21] Mahdi Javadi and **Michael Monagan**. Parallel Sparse Polynomial Interpolation over Finite Fields. *Proceedings of PASCO '2010*, ACM Press, 105–111, 2010.

[22] Mahdi Javadi and **Michael Monagan**. On Factorization of Multivariate Polynomials over Algebraic Number and Function Fields. *Proceedings of ISSAC '09*, ACM Press, 199–206, 2009.

[23] E. Kaltofen, W. Lee, and A. Lobo. Early termination in Ben-Or/Tiwari sparse interpolation and a hybrid of Zippel's algorithm. *Proceedings of ISSAC '00*, ACM Press, 192–201, 2000.

[24] François Lemaire, **Marc Moreno Maza**, Wei Pan and Yuzhen Xie. When does (T) equal Sat(T)? , To appear in Journal of Symbolic Computation, 2010.

[25] Charles E. Leiserson, Liyun Li, **Marc Moreno Maza**, and Yuzhen Xie. Parallel computation of the minimal elements of a poset. Proceedings of PASCO'10. ACM Press, 53–62, 2010.

[26] Charles E. Leiserson, Liyun Li, **Marc Moreno Maza**, and Yuzhen Xie. Efficient Evaluation of Large Polynomials. Proceedings of ICMS, LNCS 6327, Springer, 2010.

[27] Xin Li, **Marc Moreno Maza**, Raqeeb Rasheed and **Éric Schost**, The `Modpn` library: Bringing Fast Polynomial Arithmetic into Maple, To appear: Journal of Symbolic Computation, 2010.

[28] Xin Li, **Marc Moreno Maza**, and Wei Pan. Computations modulo regular chains. Proceedings of ISSAC'09, 239–246 2009.

[29] Vadim Mazalov and **Stephen Watt**. Digital Ink Compression via Functional Approximation. Proc. ICFHR 2010 (to appear).

[30] Lingchua Meng, Yevgen Voronenko, Jeremy R. Johnson, **Marc Moreno Maza**, Franz Franchetti and Yuzhen Xie. Spiral-generated modular FFT algorithms. Proceedings of PASCO'10. ACM Press, 2010, pp. 169-170.

[31] **Marnie Mishna** and A. Rechnitzer, Two non-holonomic lattice walks in the quarter plane Theoretical Computer Science, 410(38-40):3616–3630, 2009.

[32] **Marnie Mishna**, Classifying lattice walks restricted to the quarter plane, Journal of Combinatorial Theory, Series A, 116(2):460–477, 2009.

[33] **Michael Monagan** and Roman Pearce. Parallel Sparse Polynomial Multiplication using Heaps. *Proceedings of ISSAC'09*, Seoul, Korea, ACM Press, 263-269, 2009.

[34] **Michael Monagan** and Roman Pearce. Parallel Sparse Polynomial Division using Heaps. Proceedings of PASCO'10, ACM Press, 105–111, 2010.

[35] **Michael Monagan** and Roman Pearce. Sparse Polynomial Multiplication and Division in Maple 14. To Appear in *Communications of Computer Algebra.* Presented at the ISSAC 2010 software presentation session, April 2010.

[36] **Marc Moreno Maza** and Wei Pan. Fast polynomial multiplication on a GPU. Submitted to the post-conference proceedings of *High-Performance Computing Symposium 2010.*

[37] **Marc Moreno Maza** and Jean-Louis Roch. Proceedings of PASCO'10. ACM Press, 2010, 192 pages.

[38] **Marc Moreno Maza** and Yuzhen Xie. Balanced dense polynomial multiplication on multi-cores. To appear in International Journal of Foundations of Computer Science., 2010.

[39] Jason Peasgood, Symbolic Integration of Convolution Integrals, MMath (Computer Science), University of Waterloo, 2009.

[40] Daniel S. Roche, Space- and Time-Efficient Polynomial Multiplication, Proceedings of ISSAC'09, ACM Press, 295-302, 2009

[41] Daniel S. Roche, Chunky and Equal-Spaced Polynomial Multiplication, To appear in Journal of Symbolic Computation, 2010.

[42] Alan P. Sexton, Volker Sorge and **Stephen M. Watt**, Computing with Abstract Matrix Structures, Proceedings of ISSAC'09, ACM Press, 325-332, 2009

[43] **Arne Storjohann**. Integer matrix rank certification. *Proceedings of ISSAC '09*, ACM Press, 333–340, 2009.

[44] Richard Zippel. Probabilistic algorithms for sparse polynomials. *Proceedings of EUROSAM '79*, Springer-Verlag LNCS **72**, 216–226, 1979.

[45] Wei Zhou and **George Labahn**, Efficient Computation of Order Bases, *Proceedings of ISSAC'09*, Seoul, Korea, ACM Press, 375-382, 2009.