MITACS Project CV and Business Plan

November 1, 2008

Title of Project:

Mathematics of Computer Algebra and Analysis
(MOCAA)

Project website:

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

1. Project Investigators

Project leaders:

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

Project core investigators:

Contact information for network investigators may be found on the project website.

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

University of Waterloo:
Keith Geddes, Computing Science
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

Dalhousie University
Jon Borwein, Computer Science

University of Lethbridge:
Howard Cheng, Mathematics and Computer Science

Project colaborators:

Francois Bergeron, University of Quebec at Montreal
Ilias Kotsirias, Wilfred Laurier University

2. Non-Academic Partners

a) Industrial partners.

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 Addresss: lbernardin@maplesoft.com, jgerhard@maplesoft.com
Web Page: `http://www.maplesoft.com`

3. Science

3 (a) Summary of the state of the knowledge of the field

Symbolic algebra systems such as Maple and Mathematica have achieved a remarkable degree of sophistication over the last twenty years. Difficult problems, such as exact indefinite integration of elementary functions and polynomial factorization, have been attacked with considerable success. These systems have incorporated many of the most important algorithms of the twentieth century, including the Fast Fourier Transform, lattice reduction with the LLL algorithm, algorithms for computing Gröbner bases, and the Risch decision procedure for elementary function integration. As a result, they can now effectively deal with large parts of the standard mathematics curriculum and have become a central research tool in many subareas of mathematics both from an exploratory and formal point of view.

Of course there are many places where symbolic systems need improvement. For example, while these systems can all factor large multivariate polynomials with integer coefficients most still cannot do any algebra with polynomials like $x^{(n^2-n)/2} - y^n$ where n is a symbolic exponent (or in general any sort of algebra with $n \times n$ matrices with n indeterminant). Symbolic systems have limited capabilities for efficiently handling polynomials having algebraic functions such as $\sqrt{1-uv}$ in their coefficients. Similarly, while able to determine if closed form solutions can be given in terms of elementary functions, these systems typically cannot find definite integrals in terms of say elliptic functions. Our project has contributed new algorithms to answer such questions and also the software to do the actual computations, primarily in the form of Maple programs. Our first goal remains one of continuing to expand the scope of symbolic algebra systems so that they can provide useful answers to wider classes of input.

While symbolic algebra systems have had a large impact in education and research they also all have the goal to solve problems in industrial settings. One area which holds considerable promise is in engineering modeling environments. An example is the work currently being done by Maplesoft with Toyota. Here components in a given model retain the differential and algebraic equations (with symbolic parameters) that define what they do. Components are combined, with the resulting equations requiring symbolic simplification and other manipulations. Of course simulations later give values to the parameters and the defining equations are in turn solved numerically. The key is that the components of the models always keep their mathematical representations so that the equations can be viewed and understood in symbolic rather than numeric form. One can think of this as a symbolic version of Matlab's Simulink where boxes represent exact rather than approximate models and where no information is lost during the modeling. Maplesoft is releasing a new product, MapleSim, for this in early 2009.

The second goal of our project is to strengthen and extend the capabilities of symbolic computation for industrial settings. Currently, the biggest stumbling block to the use of symbolic computation in an industrial setting is the efficiency of exact algorithms. Used naively, symbolic computation can generate huge formulae that fill memory. Our plan for realizing this goal is to increase the size of problems that can be efficiently handled by core operations in symbolic algebra systems. These core operations include linear algebra, symbolic and numeric polynomial algebra and symbolic differential algebra.

High performance computer algebra focuses on techniques such as parallel computation, the use compiled rather than interpreted libraries for linear and polynomial algebra and the construction of optimal algorithms using bit-complexity models. The optimal algorithms are built using theoretic advances (for example, give procedures in terms of matrix multiplication or polynomial multiplication where fast methods exist) but which ultimately also provide practical advances (for example, making use of highly tuned numerical libraries at the hardware level - e.g. BLAS and ATLAS).

3 (b) Summary of main achievements during last project CV

In this section we summarize the project's main achievements since November 2006 and the project's progress towards objectives set at that time. We will give highlights in five areas : (i) linear algebra, (ii) high performance computer algebra, (iii) polynomial algebra, (iv) symbolic-numeric computation and (v) emerging technologies of computer algebra.

In the case of linear algebra we have work to report on fast algorithms for working with integer matrices, Ore matrices and linear system solving. Some highlights include:

(1) The paper "Faster algorithms for the characteristic polynomial" (2007) by Pernet and Storjohann gives an asymptotically optimal, randomized algorithm for computing the characteristic polynomial of an arbitrary matrix over a field. The improvement over the earlier near optimal algorithm of Keller-Gehrig in 1982 gives the first complexity breakthrough for this problem in over twenty years. The new algorithm is also practical. An implementation of the new algorithm in LinBox demonstrates significant running time improvements compared to previously most efficient implementations.

A copy of this paper has been included with the submission.

(2) In "Faster inversion and other black box matrix computations using efficient block projections" (2007), Eberly, Giesbrecht, Giorgi, Storjohann and Villard give a proof of the existence of efficient block projections for arbitrary non-singular matrices over sufficiently large fields. The result provides the final tool needed to complete the probabilistic Las Vegas algorithm for solving *sparse* linear systems over the integers in sub cubic time given earlier by the authors in their ISSAC paper of 2006 (where its runtime was validated using LinBox).

(3) The paper "Solving linear systems over cyclotomic fields" ([23], 2008) by Chen and Monagan describes three modular algorithms for solving $Ax = b$ modulo $\Phi_k(x)$, the k'th cyclotomic polynomial. The fastest method uses a representation for the solutions $x_i \in \mathbb{Q}(z)$ that is a factor of $d = \deg \Phi_k(z)$ more compact than the standard representation (in general, e.g. for random inputs). However, on real problems given to us to solve, the standard representation was often much more compact. Thus the code, installed in Maple in 2008, simultaneously computes the solution in both representations and stops when the first method succeeds.

(4) The paper "Output-sensitive Modular Algorithms for Polynomial Matrix Normal Forms" (2007) by Cheng and Labahn gives algorithms for modular computation of row reduced, weak Popov and Popov forms of polynomial matrices and the corresponding unimodular transformation matrices. The modular algorithms are output-sensitive and can be used for to solve one-sided matrix gcd and lcm problems and give irreducible matrix-fraction descriptions of matrix rational functions.

(5) In the paper "Solving structured linear systems with large displacement rank" (in press), by Bostan, Jeannerod and Schost, the authors discuss the complexity of solving structured linear systems. One way to measure structure is through "displacement rank". As it turns out, none of the previous algorithms adequately managed systems featuring a large rank. The algorithms proposed here handle this question. They feature the best known complexity for large classes of systems, with applications to multivariate interpolation, Hermite-Pade approximation, etc.

(6) The paper "The Solution of $S\exp(S) = A$ is not always the Lambert W Function of A" (ISSAC 2007) by R. M. Corless, H. Ding, N. J. Higham, and D. J. Jeffrey represents the first attempt to

systematically solve transcendental matrix equations. The matrix exponential, matrix logarithm, and certain other notable matrix functions, and the numerical solution of important matrix equations, such as the Sylvester equation, have been well studied numerically. In contrast, this work represents a beginning of the study of certain transcendental matrix equations. Recent work by Higham on matrix functions (his Chapter in the book *The Handbook of Linear Algebra*) demonstrate that this work is timely.

(7) The paper "Linearization of Matrix Polynomials Expressed in Polynomial Bases" (2008) by A. Amiraslani, P. Lancaster and R. Corless establishes new strong linearizations for matrix polynomials. This includes a linearization for matrix polynomials expressed in the Lagrange basis, including the case where the leading coefficient is singular.

Considerable work has also been done during the past two years in the area of high performance computer algebra. In particular we mention the following highlights:

(1) The paper "Fast arithmetic for triangular sets: from theory to practice" by Li, Moreno Maza and Schost ([45], in press) proposes highly efficient algorithms for low-level operations supporting triangular decompositions such as polynomial multiplication modulo a triangular set. The complexity results given there are the best known so far for this operation. Our implementation outperforms the packages in `Magma` or MAPLE with similar specifications. This advance extends a series of papers by the same authors on implementation techniques for fast polynomial arithmetic.

A copy of this paper has been included with the submission.

(2) Our research on high-performance computer algebra has given birth to `modpn`, a C library of *fast arithmetic for multivariate polynomials over finite fields*. The main objective of `modpn` is to provide highly efficient routines for supporting the implementation of modular methods in MAPLE, and hence obtain considerable efficiency improvements. Timings as a result are quite impressive. For example, the `modpn` library solves a random dense bivariate system of two polynomials f_1, f_2 of total degree 100 (over a prime field) in the same time that MAPLE expands the product of $f_1 f_2$. The system in question has 10,000 solutions and of course no other tool in MAPLE can solve it. The `modpn` library consists of over 35,000 lines of C code along with 5,000 lines of MAPLE code. It has been submitted to the next release of MAPLE. The articles [47, 48] report on the design and performances of `modpn`.

(3) The paper "Sparse Polynomial Pseudo Division Using a Heap" ([54], 2008) by Monagan and Pearce gives a new *fraction-free* algorithm for *multivariate* polynomial division over the integers which uses a binary heap, and an optimized heap based multivariate polynomial multiplication algorithm. On a variety of benchmarks, the multiplication and division codes are 10 to 100 times faster than those in Maple, Magma, Pari and Singular.

A copy of this paper has been included with the submission.

(4) The article "Memory efficient scheduling of the Strassen-Winograd matrix multiplication algorithm" by Dumas, Pernet and Zhou (submitted [31]) looks at the problem of allocating extra memory for the Strassen-Winograd matrix multiplication algorithm and reduced the memory requirements by a factor of 3. They also propose a fully in-place algorithm for multiplication (when allowed) and finally an $O(n^{\log_2 7})$ in-place algorithm for computing the product with no overwriting of the inputs.

In the area of polynomial algebra we can report the following highlights:

(1) The paper "A Sparse Modular GCD Algorithm for Polynomial GCD Computation over Algebraic Function Fields" ([38], 2007) by Monagan and Javadi presents a first sparse modular GCD algorithm for computing the GCD of two polynomials in $L[x_1, ..., x_n]$ where L is an algebraic function field in $k \geq 0$ parameters with $l \geq 0$ field extensions. A complete implementation was installed in Maple in the summer of 2008. Benchmarks demonstrating the performance of the algorithm, including the resulting improvement of Maple's polynomial factorization code, can be found in `http://www.cecm.sfu.ca/~pborwein/MITACS/AchievementsSummary.htm`

(2) In two companion papers by Bostan, Salvy and Schost, "Fast conversion algorithms for orthogonal polynomials" ([9], accepted) and "Power series composition and change of basis" ([12], 2008), we give low-complexity algorithms for some basic operations on polynomials and power series, such as change of basis between orthogonal and monomial bases (such as so-called "discrete polynomial transforms").

The first paper discusses the general case of an arbitrary sequence of (formal) orthogonal polynomials: we give algorithms of cost close to optimal; in the second one, we focus on some specific families (e.g., Jacobi polynomials) and give yet better algorithms, using the nice properties of their generating series. Notably, in both cases, beyond classical techniques such as divide-and-conquer and Newton iteration, a crucial use is made of duality techniques (we rely heavily on the "transposition" of algorithms).

(3) Two recent contributions which add to the toolkit of tractable algorithms for *lacunary* polynomials. In the paper "Interpolation of Shifted-Lacunary Polynomials" by Giesbrecht and Roche (2007), we show how to perform lacunary sparse shift interpolation. This allows us to reconstruct efficiently lacunary polynomials from interpolation points, even when the polynomial is only sparse in a shifted basis 1, $x-\alpha$, $(x-\alpha)^2$, ..., for an *unknown* shift α. This addresses an open question of Grigoriev & Karpinski (1993). In the paper "On Lacunary Polynomial Perfect Powers", Giesbrecht and Roche (2008) present new polynomial-time algorithms to determine if a lacunary polynomial is a perfect power, and find the sparse polynomial root. This is surprising in that evidence suggests related problems such as irreducibility testing are intractable. This algorithm is implemented in NTL and shows dramatic improvement in practice, even for dense polynomials.

(4) The paper "Comprehensive Triangular Decomposition (CTD)" ([20], 2007) by Chen, Lemaire, Golubitsky, Moreno Maza and Pan proposes a new algorithmic approach for studying polynomial systems with parameters. Our implementation of the CTD has been integrated into the `RegularChains` library in the release 12 of MAPLE. The CTD also brought new algorithmic tools allowing us to realize the first package dedicated to the manipulation of (parametric or not) constructible sets: the `ConstructibleSetTools` module of the `RegularChains` library. As a byproduct, we obtained the first software tool for verifying polynomial system solvers computing decompositions (triangular, equidimensional, etc.). reported in "On the verification of polynomial system solvers" ([22], 2007) by Chen, Moreno Maza, Pan and Xie.

(5) Significant advances on the theory of polynomial system solving have been made by our group. They are necessary progress toward better algorithms. One highlight is The paper "When does (T) equal SAT(T)?" ([42], 2008) by Lemaire, Moreno Maza, Pan and Xie proposes an algorithm to decide whether a regular chain does or does not generate its saturated ideal. This is an essential question for handling redundant components when decomposing a polynomial system. This advance is a step toward solving deep questions in polynomial algebra such as deciding whether an ideal is in *complete intersection*.

In the area of Symbolic-Numeric computation the projects included approximate polynomial interpolation. Here we mention:

(1) The paper "Symbolic-numeric Sparse Interpolation of Multivariate Polynomials" (to appear) by Giesbrecht, Labahn and Lee looks at the problem of sparse interpolation of multivariate polynomials represented as black-boxes over floating point numerical environments. It is shown that interpolation at random roots of unity combined with generalized eigenvalue computation results in efficient and numerically robust solutions.

A copy of this paper has been included with the submission.

(2) The paper "On the numerical condition of a generalized Hankel eigenvalue problem" (2007) by Beckermann, Golub and Labahn looks at the numerical sensitivity of certain structured eigenvalue problems. Results of this paper show that in general such problems as sparse interpolation of black-boxes and reconstruction of the shape of a polygon from its moments are very sensitive to numerical errors.

(3) The paper "Symbolic-numeric Computation of Implicit Riquier Bases for PDE" ([72], 2007) by Wu and Reid and the paper "Implicit Riquier bases for PDAE and their semi-discretizations" ([73], accepted, 2008) by Wu, Reid and Ilie generalize a fast prolongation technique of Pryce from DAE and PDAE and represents another major practical advance. They show that for a generic class of differential polynomials, prolongation with respect to one independent variable yields a polynomial time algorithm for computing an implicit Riquier Basis (a type of formally integrable system). Examples in [72] demonstrate that this class occurs frequently in applications. The technique completely avoids elimination, a major source of expression swell in symbolic prolongation-elimination approaches.

Highlights in the area of Emerging Technologies include:

(1) The paper "Elliptic integral representation of Bessel moments" ([3], 2008) by Bailey, Borwein, Broadhurst and Glasser provides a major advance in our understanding of the periods and number theoretic content of massive amplitudes in perturbative quantum field theory and condensed matter physics. The paper contains many new formulae for definite integrals involving products of Bessel functions which were found using tools from computer algebra.

A copy of this paper has been included with the submission.

(2) In the paper "Two Families of Algorithms for Symbolic Polynomials"(2007) by Watt studies multivariate polynomials with exponents that are themselves integer-valued multivariate polynomials, and presents algorithms to compute their GCD and factorization. In "Symbolic Polynomials with Sparse Exponents" (2008) Watt studied the ring of *symbolic* multivariate polynomials, that is multivariate polynomials like $x^{(n^2-n)/2} - y^2$ with exponents that are integer-valued multivariate polynomials. For "symbolic" univariate polynomials with coefficients from a field of characteristic zero and multivariate integer-valued polynomials as exponents, we have characterized when and how they may be functionally decomposed (in the sense of Ritt) and present an algorithm that computes the decomposition when it exists.

(3) The paper "On the zeros of cosine polynomials: solution of an an old problem of Littlewood" (2008) by P. Borwein, T. Erdélyi, R. Ferguson and R. Lockhart, to quote the referees, "spectacularly disproves an old conjecture of Littlewood". This was based on very extensive numerical and mathematical experimentation, primarily using Maple which included developing probabilistic algorithms in computational analysis.

3 (c) Objectives

The emphasis of the various projects is the development and implementation of software for finding exact as opposed to numerical solutions to mathematical problems. This is an overlap of computer science with pure and applied mathematics, requiring deep analysis and sophisticated algorithmic development. Many researchers develop software tools for their specific needs. The investment required to adapt these tools for wider applicability is huge. However, the importance and benefits of providing such good working tools to the general mathematical community is difficult to overestimate. This is an expanding area of expertise, offering considerable opportunity in both academic and industrial environments.

Objectives of the projects in this consortium may be summarized as follows:

- new algorithm development in computational algebra and related areas
- software development in our various projects
- applying our software in other research projects
- delivery of software to general users through Maple and/or over the web

3 (d) Methodology

Canada's position as a major participant in the development of mathematical software stems from the early eighties and the very successful creation of Maple at the University of Waterloo. Systems such as Maple, and its main competitor Mathematica, and the related numerically-oriented system Matlab, are now primary research and development tools in mathematics. These are now substantial sized businesses, with Maplesoft employing well over one hundred personnel. Even at its current size, Maplesoft interacts closely with the research groups at the three principal sites of this proposal, Waterloo, Simon Fraser, and Western. Indeed these three labs are Maple's primary source of both leading-edge research and prototype development. This unique relationship was recognized in 2004 with an NSERC Synergy award between Maplesoft and these three computer algebra research groups. The problems under investigation are determined in conjunction with Maplesoft Inc. on the basis of our common interests. As a corporation, Maplesoft is comfortable with and understands that it will get the best product from people doing what most interests them. It is a very satisfactory partnership.

One of the important directions in computer algebra is to provide algorithms that allow symbolic computation to play a significant role in solving industrial problems. In order to achieve this, computer algebra systems need to address problems of efficiency. Current PC technology has moved to multi-core processors with dual-core and quad-core processors now commonplace, and machines with 80 cores are now anticipated in 2011. Our approach takes the point of view that overall performance gains will be obtained by making core operations fast, particularly those dealing with multivariate polynomial algebra and linear algebra. This includes, for example, parallel algorithms and parallel support tools for implementing them at low levels. Finally, while performance gains are important, we also plan to make progress on extending the applicability of symbolic computation by developing new algorithms for fundamental computations.

3 (e) Descriptions of Subprojects

Our subprojects are presented under four headings: *high-performance computer algebra*, *polynomial algebra and solvers*, *symbolic linear algebra* and *additional projects in computer algebra*.

3.1 High-performance computer algebra.

When applied to computer algebra algorithms, high-performance computing offers challenges which are specific to our discipline. First, intermediate data size renders the usual challenge of memory traffic even more dramatic. Second, dynamically spawned tasks (in a parallel run) tend to have very irregular work, in particular in high-level algorithms, which make them difficult to schedule. Third, computer algebra programs are better written in high-level programming languages offering genericity and abstraction. But a tight management of computing resources is much easier to achieve in low-level languages like C.

With our first two subprojects, we address these issues by focusing on efficiency-critical low-level routines: multivariate polynomial arithmetic over finite fields, over the rationals and modulo regular chains (thus covering towers of algebraic and transcendental extensions of prime fields). Indeed, the efficiency of many facilities in computer algebra systems like Maple depends a lot on the efficiency of the underlying polynomial arithmetic. Implementation techniques, including code generation, code optimization and platform adaptation for a specific domain of routines, are important considerations for our other two subprojects.

3.1.1 The `modpn` and `sdmp` libraries.

Our MOCAA project during the period 2006 - 2008 brought two C libraries providing highly optimized sequential code for multivariate polynomial arithmetic:

`modpn`: for fast arithmetic (SLPs and FFT techniques) modulo regular chains over finite fields, and
`sdmp`: offering sparse arithmetic (using heaps) over prime fields, and $\mathbb{Z}$.

We shall draw on the sequential C code of `modpn` [43, 47, 48] and `sdmp` [55, 54] to design and implement new parallel algorithms for operations with sparse multivariate polynomials and modulo regular chains.

In a preliminary step, we plan to generalize our sequential codes such that `modpn` and `sdmp` could work over $\mathbb{Q}$ and modulo a regular chain, respectively. Together, they would provide a fairly complete set of efficient low-level routines in support of higher-level algorithms.

While there are several "obvious" approaches for parallel multiplication, it's not clear which is the best for sparse polynomials. And for sparse polynomials there are no "obvious" approaches for division. For dense operations, we shall extend the parallelization of the `modpn` library initiated in [44], for normal form computations, to all core operations (subresultant chain computation, regularity test, etc).

Based on those enhancements of `modpn` and `sdmp`, we anticipate that higher-level code can gain efficient parallel execution and, more generally, higher performance.

3.1.2 Immediate monomials.

In [54] Monagan and Pearce implemented, in C, new polynomial multiplication and polynomial division algorithms and found them to be over 100 times faster than those in MAPLE. Their attempt to integrate this into MAPLE showed that on large sparse problems, up to 90% of the time was spent converting the result into MAPLE's polynomial representation. In [45] X. Li, M. Moreno Maza, and É. Schost observed, in a different context, a similar overhead of 90%. The problem is caused by MAPLE's representation of monomials and the high cost of simplifying them and entering them into MAPLE's "simpl" table.

This project is to develop a new top level default representation for polynomials in one or more variables for MAPLE that uses immediate monomials. An immediate monomial is a monomial x^3yz^2 which has been encoded (packed) into one machine word. Immediate monomials will avoid MAPLE's simplifier and simpl table. To make this work inside MAPLE we will use the number of variables in the polynomial to fix the packing; if there are too many variables, or they have too high degree, then the existing representation would be used.

This project will be undertaken jointly with Maplesoft personnel and the group at Simon Fraser. Our goal is to realize the full factor of 100 gain on core polynomial operations. The new representation will also result in big efficiency gains for other basic MAPLE operations on polynomials. We think this could be the biggest single improvement ever made to MAPLE's performance.

3.1.3 Transposition techniques.

Future work on some foundational aspects of computer algebra includes the development of a formal context in which to write "transposed algorithms". Indeed, it is known that given an algorithm to perform a linear operation, there exists an algorithm of the same cost for the transposed operation. This has proved useful on many occasions, such as [9, 11, 12] among many others. However, the proof of this general result, while constructive, uses an elementary model for algorithms, without such basic tools as recursion or loops. Memory aspects are also not covered. As a consequence, most uses made of transposed algorithms actually involved tailor-made code transposition.

In the next two years, we plan to obtain a much more versatile form of the transposition principle. This would allow to actually transpose code written in a C-like dialect, without manual intervention. Similar frameworks already exist for the related question of automatic differentiation of programs. A first step in this direction is currently taking place in the M.Sc. thesis of L. Ding (UWO), who considers transposition of polynomial multiplication algorithms.

3.1.4 Compilable numerical evaluation routines for bivariate functions.

The motivation for this project stems from the fact that the speed of numerical evaluation in MAPLE (and other languages) for bivariate functions such as `BesselJ(v, x)`, and many other bivariate functions, is much slower than would be desired. By *numerical evaluation* we mean evaluation at hardware floating point precision. As is well known, efficient numerical evaluation routines for various mathematical (univariate) functions is achieved via a library which exploits polynomial or rational function approximations of the desired functions. The state of the art for developing numerical libraries for functions of more than one variable is much less well developed.

The solution being developed in this project is to exploit approximations derived via natural tensor product series, as introduced in the doctoral thesis of F. Chapman [18] supervised by K. Geddes. Work during 2006-08 by Geddes and Chapman, along with Masters students, on both the convergence theory and on development of code has shown the feasibility of this approach. We plan to develop as production code a MAPLE package for generating numerical evaluation routines, based on the above prototype code realized by X. Wang [70]. This package would be exploited for two purposes. One purpose is to use the MAPLE package to generate subroutines and produce a "standard numerical library" (via compiled C code) for the efficient evaluation (in hardware floating point precision) of various specific bivariate functions. Secondly, the package would be available in MAPLE for users who may wish to generate efficient numerical evaluation routines for various user-defined bivariate functions.

3.2 Polynomial algebra and solvers.

Solving systems of polynomial equations, linear or non-linear, algebraic, semi-algebraic or differential, using symbolic or symbolic-numeric methods, remains the driving subject in computer algebra. Certain types of input systems and algorithms to solve them are well understood. Thus, they are now subjects of study in high-performance computing. Other types of input systems still require theoretical and algorithmic advances; they are addressed with the our six subprojects below. The first two projects deal with two types of algebraic systems which are more and more important for applications: *parametric systems* and *large systems* (in the sense of many variables and equations). The last three subprojects propose to enhance MAPLE's capabilities in solving algebraic systems with *approximate coefficients* and *systems of differential polynomials*, two major areas for applications.

3.2.1 Comprehensive triangular decomposition.

A first step toward high-performance solvers is to design algorithms which can create opportunities to make use of fast arithmetic and modular methods. In [29], using the notation of *equiprojectable (triangular) decomposition*, we achieved this goal for polynomial systems with finitely many solutions. Unfortunately, the techniques do not generalize to polynomial systems with infinitely many solutions.

The first goal of our project is to specify a type of triangular decomposition which will fills this gap. The notion of *comprehensive triangular decomposition* that we introduced in [20] for parametric polynomial systems could be a first building block. An important idea emerged from this work: for polynomial systems of arbitrary dimension, it is necessary to extend the scope of the study from algebraic varieties to constructible sets.

The work of [20] is limited to the complex case and the second goal of our project is to extend comprehensive triangular decompositions to semi-algebraic sets, following the path initiated by Lu Yang, Xiaorong Hou, and Bican Xia [77]. This is needed to attack "real world" applications. A preliminary step in this direction is the introduction of the module `SemiAlgebraicSetTools` and the command `RealRootClassification` (from the package `Discoverer` of Bican Xia) in the `RegularChains` library.

These advances for solving parametric polynomial systems indicate that a *software expert system* could efficiently generate an automatic and systematic discussion of the locus of the equilibria, together with the normal forms at these equilibria, of a given (polynomial) dynamical system, depending on the values of its bifurcation parameters.

3.2.2 Automatic selection of solving strategies.

Given non-constant polynomials $F = f_1, \ldots, f_m \in \mathbb{K}[x_1, \ldots, x_n]$ over a field $\mathbb{K}$, we aim at choosing a total order on the set of variables $X = \{x_1, \ldots, x_n\}$ such as to "minimize" the cost of computing a triangular decomposition of the system $f_1 = \cdots = f_m = 0$. This choice should rely only on considerations on the partial and total degrees in F and it should not involve any operations on the coefficients.

By minimizing the cost of solving, we mean minimizing the running time of a software solver, like the `Triangularize` command of the `RegularChains` library in MAPLE. This project has, actually, developed jointly with Maplesoft personnel and the groups at the University of Western Ontario and University of Waterloo; some significant experimental results have already been obtained and reported in a poster at the *Milestones in Computer Algebra 2008* workshop.

The question of selecting variable ordering (heuristically) arises in at least two occasions. First, if any variable orders would be of interest in order to solve F, in which case one would like to select a "minimizing" one. Secondly, if one variable order is prescribed and one could make use of a solving strategy based on a change of variable order such as the one of [28].

For systems with a large number of equations and unknowns, it is desirable to decompose the input system into sub-systems or "blocks" to be solved one-after-another, generalizing the notion of a triangular system. (The concept of "good specialization" as introduced in [20] helps formalizing this generalization.) This idea was successfully experimented in [33] on a particular family of systems. This project aims at turning these ideas into an algorithm capable of selecting a "solving strategy" for a large input system.

3.2.3 Polynomial algebra by values and derivatives.

Polynomial Algebra by Values refers to doing polynomial algebra solely given the values of the polynomial and/or its derivative —that is, working in a Lagrange or Hermite basis. In a numerical environment it is numerically unstable to convert to a standard representation and hence one must remain in the alternate basis. The question naturally arises on how much can be done directly using such a basis.

Recent progress by Corless and his co-authors in constructing a generalized companion matrix pencil for polynomials expressed by values and derivatives provides the motivation for several further projects. In particular we are currently investigating an extension to the Birkhoff interpolation problem. This is of interest for the numerical solution of ODEs, as well as other algebraic applications. This is a joint investigation with Corless, Butcher (Auckland) and Gonzalez-Vega (Santander). Further extensions of this approach to multivariate algebraic equations by use of the Bézout matrix are also planned.

3.2.4 Numerical algebraic geometry in MAPLE.

MAPLE has made a dedicated effort to have the best polynomial system solvers. We wish to develop numerical algebraic geometry in MAPLE which has been having significant impact in recent times. Numerical algebraic geometry, a new area co-created by Sommese & Wampler. Their 2005 book *The Numerical Solution of Systems of Polynomials*, is the only book in the area. Numerical algebraic geometry gives the first stable global methods to characterize all positive dimensional irreducible components (manifolds) of solutions of general approximate polynomial systems. It uses homotopy methods that compute points on the components called *witness points* to represent the components. Encoded with straight-line programs, homotopy methods can be computed at low polynomial cost (Ilie, Corless & Reid [37]), which maybe regarded as a development of the early complexity results of Shub and Smale.

3.2.5 Differential algebra.

Some recent complexity results in solving arbitrary systems of differential equations symbolically (see for instance [35]) suggest that this task is significantly harder than solving systems of algebraic equations. Nevertheless, symbolic differential algebra which has obtained major successes for some specific tasks such as preprocessing ODEs (used in the MAPLE's ODE tool kit), *Cartan's equivalence method* [13] (used in [61] by Petitot and Neut) or modeling in biological systems (Boulier and Lemaire, [10]).

It is therefore necessary to improve the performances of our tools for symbolic differential algebra. The flexibility of the rifsimp algorithm (implemented in MAPLE) to choose alternative symbolic methods for its leading nonlinear algebraic equation processor will be exploited. We plan to interface the rifsimp package with the `RegularChains` library (with its fast arithmetic support `modpn`) and the `Groebner` library

(powered by the `Fgb` solver of Faugère). Both these packages are available in MAPLE. The possibility of incorporating geometric resolutions will also be explored. On the symbolic-numeric side, we wish to take approximate systems of differential-algebraic equations, and implement an environment with the basic operations of linearization, discretization, and deformation, linked with the numerical algebraic geometry package project of the previous section.

3.3 Symbolic linear algebra.

Linear algebra over a field or a ring is one of the most important research areas in mathematics with many applications. Solving linear systems of equations over a field and finding bases of modules are only two examples of common operations encountered often in computer algebra. Linear algebra is also a difficult area computationally if only because of the size of its basic object, an $m \times n$ matrix. Computations quickly get large. The aim of Symbolic Linear Algebra is to both improve the efficiency of basic linear algebra operations and to provide the mathematical functionality needed to tackle new problems.

3.3.1 Fast linear algebra over number fields and function fields.

Our modular methods for linear algebra and polynomial algebra over the integers and rationals can also be applied to do linear algebra over number fields, polynomial rings, and algebraic function fields.

The simplest number fields in this context are the cyclotomic fields. For a prime p, the cyclotomic polynomial $\Phi_k(x)$ factors modulo p into distinct linear factors whenever $k|p-1$. In this case computing modulo $\Phi_k(x)$ can be reduced to $\deg(\Phi_k)$ computations modulo p, that are done in parallel. In [23] we used this to develop three modular methods for solving $Ax = b \mod \Phi_k(x)$. At the same time in [1] we have developed two sequential methods for computing $\Phi_k(x)$ of very high order – $k > 10^9$.

Computing over extension fields (algebraic or transcendental extensions) is a particular case of computing modulo regular chains. The `MatrixTools` module of the `RegularChains` library [41] already contains tools for doing linear algebra in such a context, albeit not using modular techniques or fast arithmetic [29, 28].

We propose two new projects that are natural in this context. The first is to build general tools for solving $Ax = b$ over rational function and algebraic function fields. The second is to improve the algorithms and implementation of our modular tools.

For the first project, one of the tools needed is *sparse* rational function interpolation. In [39] we developed our own algorithm called RATZIP for this purpose, while an alternative algorithm can be found in [40]. We shall develop a good C implementation for this so that we can use it in a variety of situations.

For the second project we will work to improve the performance of `MatrixTools` for computations modulo small primes, using the existing `Modpn` C library. Non-trivial algorithms will need to be devised, along the lines of of [27]. Finally, a last ingredient will be to use Hensel lifting techniques for regular chains, relying on the implementation already present in `RegularChains`.

3.3.2 Symbolic Matrix Analysis

It is often possible to describe the structure of matrices of indeterminant size. For example, one often refers to the Sylvester matrix of two polynomials $\sum_{i=0}^{n} a_i x^i$ and $\sum_{i=0}^{m} b_i x^i$ (for undetermined m and n) having size $(m+n) \times (m+n)$. Another example is a matrix having one formula for the diagonal entries, another expression for k super diagonals (for undetermined k) and is zero everywhere else. Earlier work by Sexton

and Sorge [65] has studied the problem of representing such *symbolic matrices*. Symbolic matrices appear often in textbooks and are natural objects in future pen-based math systems [69, 67].

It is natural to want to do arithmetic with symbolic matrices if possible. Unfortunately this can be very difficult because the usual approach of maintaining case structure leads to severe problems of computational complexity. For example, if we were adding N different vectors $v_i = [v_{i1}, ..., v_{ih_i}, w_{ih_{i+1}}, ..., w_{in}]$ where each vector v_i has components v_{ij} for $1 \leq j \leq h_i$ and has components given by w_{ij} for $h_i < j \leq n$, then the sum of these vectors would have $N!$ cases, each corresponding to an ordering of the h_i.

We propose a project to research algorithms for vector and matrix addition, to do this via a good choice of basis tensors. For example, addition of vectors can be written as simple sums with $O(N)$ terms by defining the sum as $s_i = [v_{ij}\xi_{i,1,h_i+1} + w_{ij}\xi_{i,h_i+1}, n+1]_{j=1..n}$ where $\xi_{i,a,b}$ is 1 if $a \leq i \leq b$, -1 if $b \leq i < a$ and 0 otherwise.

3.3.3 Matrix Normal Forms of Ore Matrices

Normal forms (such as Hermite, Popov, shifted Popov, or Smith) for matrix polynomials appear in many areas of mathematics and engineering. For example, the Popov normal form is the central tool used in the conversion of input-output linear systems represented as transfer functions (rational matrix polynomials) into state-space representations (first order systems). The normal forms in question can also be defined in the case of matrices of non-commutative elements, for example matrices of differential and recurrence operators or an Ore algebra.

We propose three projects. The first two projects involve separate approaches for the efficient computation of these normal forms, in particular in exact arithmetic environments where coefficient growth is a concern. This has been done in the case of matrix polynomials by Beckermann, Labahn and Villard [8] and in the case of matrices of shift operators by Cheng and Labahn [24, 25]. These algorithms determine normal forms via an associated null space computation. Our first project will be to see if it is possible to extend these results to the general Ore case. At the same time the null space approach is an indirect method which does not work by reducing degrees of entries. As such the computation may be quite inefficient for matrices which are already close to normal form. A direct approach such as that of Mulders and Storjohann [60] avoids such a problem, but does not handle coefficient growth. At the same time there is no clear way to make [60] handle such growth. Our second project is to create a procedure which is both direct (degree reductive) and which controls intermediate expression size. Both fraction-free and modular approaches will be examined.

A third project involves the use of the Popov form for differential operators in order to find efficient methods for solving systems of higher order linear differential equations. The use of Popov forms in the commutative case to convert input-output systems from transfer functions to state-space representations has a parallel in the non-commutative, differential operator case in the conversion of higher order differential systems $L(D)y(x) = w(x)$ to a first order system $z'(x) = A(x)z(x) + b(x)$. Existing algorithms and software for solving linear differential systems of equations typically require that the input is first order [4]. We plan to investigate algorithms for solving higher order linear differential systems without the need for first order conversion. The particular algorithms would include finding solutions within the domain or extensions. We believe that this may be possible by making use of the Popov form of the input matrix $L(D)$ and taking advantage of its special structure.

3.3.4 Applications of the outer product adjoint formula

A feature of many linear algebra problems involving polynomials with coefficients from a field is growth in the degrees of the polynomials in the output. For example, while the space required to represent an $n \times n$ polynomial matrix of degree less than d is exactly $n^2 d$ field elements, the inverse of the same matrix requires up to $n^3 d$ field elements. More importantly, classical algorithms to compute the inverse require time proportional to more than $n^4 d$ field operations, a factor of n more than the space required to write the inverse down. The extra factor of n means that inverse computation is impractical even for modest input sizes, for example $n = 1000$.

We have recently discovered the *outer product adjoint formula* [68] that allows the exact inverse of a polynomial matrix to be computed in time proportional to approximately $n^3 d$ field operations. The matrix inverse is the most fundamental concept in linear algebra. The goal of this research project is to explore applications of the outer product adjoint formula to obtain faster algorithms for a wide variety of other linear algebra problems. We are currently working on extending the formula to the case of integer matrices, and will investigate consequences of the formula for black box computations.

One concrete project is to apply this outer product adjoint formula to obtain an optimal algorithm for computing the nullspace of a polynomial matrix having rational number coefficients. Such nullspace computations arise as the bottleneck step in symbolic summation problems, for example, but are especially difficult because there is growth both in the degrees of polynomials as well as in the size of the rational coefficients. Our plan is to develop and implement an algorithm that is a factor of n faster than standard techniques.

3.3.5 Vector rational reconstruction

In order to reduce computational cost many algorithms compute in a ring and then at a final step reconstruct a rational form of the required solution. In the case of vectors (e.g. linear solving) this final step involves reconstructing a vector of rational numbers or functions having a single common denominator modulo a given modulus. Interestingly enough the use of rational approximations with common denominators modulo a modulus to vectors of functions dates back to the time of Hermite where he used such a construction in his famous proof of the transcendence of e. That construction was later formalized by Padé into the notion of simultaneous Padé approximants.

We have two projects in mind with respect to the vector rational reconstruction problem. The first is to extend the vector rational function reconstruction algorithm of Olesh and Storjohann [62] to the integer case (and so solve the vector rational number problem). The second is to improve the efficiency of the order basis algorithm of Beckermann and Labahn [7] in the case of simultaneous Padé approximation.

The overall cost of the many modular computations is not typically dominated by the rational reconstruction, but rather by the required size of the modulus and the computation of the images themselves. The algorithm of [62] shows how to perform this reconstruction with significantly smaller size. The goal of extending this to the rational number case is highly nontrivial. The two major difficulties are the presence of carries in integer arithmetic and the fact that a polynomial time lattice basis reduction for integer matrices is only approximate and not optimal as in the polynomial case.

In the related case of simultaneous rational approximation (where both numerators and a common denominator are constructed), the algorithm of [7] gives an efficient fraction-free algorithm which controls coefficient growth in the computation (an important consideration for computation in exact arithmetic environments). Our goal in the second project is to develop a more efficient fraction-free method for the computation of simultaneous rational approximation or interpolation functions by taking advantage of both

the added structure of the input matrix along with making use of its duality with the Hermite-Padé [6] problem.

3.4 Additional projects in computer algebra.

3.4.1. Zeros and poles of Padé approximants.

The goal of this project is to explore the location of the zeros and poles of the Padé approximations to the (suitably normalized) Riemann Zeta function. Why look at the Padé approximants to the Riemann zeta function? The first reason, obviously, is the relationship to the Riemann hypothesis. The patterns are striking and the computations difficult. Little is possible to prove, but much is suggested. If one really understood any of the diagrams of the zeros and poles one would be able to prove the Riemann hypothesis. A worthwhile but rather too lofty goal.

However, even assuming the Riemann hypothesis the particular behavior of the Padé approximations is not obvious. Clearly there are limit curves both of the zeros and the poles. One goal is to figure out what these limit curves are. Generating such plots are more difficult than it first might appear. For example, standard symbolic packages fail. As such another goal is to describe the necessary computations. There is an interesting body of theory due originally to Szegö that describes the zeros of the partial sums of the power series expansion of the exponential function. This extends to the zeros and poles of the Padt'e approximants to the exponential function and a few related functions. In order to get limit curves one scales the zeros and poles by dividing by the degree. The analysis is possible because there are explicit integral representations of the numerators and denominators. However there are no useful explicit representations known for the Padt'e approximants to the zeta function or even for the Taylor series for that matter. Indeed the principal problem in generating the approximations numerically is to derive large Taylor expansions.

Finally, we also wish to explore general routines for computing and plotting, within Maple, the zeros of analytic functions. This research continues with work already in place from students and research staff at SFU (including partial implementations of relevant tools).

3.4.2. Automatic combinatorics.

The goal in this research is to create tools for enumerative and bijective combinatorics using algebra and symbolic computation methods, and to better understand the analytic nature of generating functions. Indeed, the tools under present development center around the generating function, a formal power series associated to a class of combinatorial objects, and the various analytic and algebraic properties that they satisfy. In particular, many mathematical algorithms and properties for linear differential equations are useful when the generating functions are known to satisfy differential equations. We would like to understand from a combinatorial standpoint when the algorithms are appropriate, and what, exactly, they can tell us about the combinatorial objects in question.

We have two projects in mind. The first works within the algebra of differential operators, and involves developing new algorithms, and applying known algorithms to the generating functions, and to improve computations. This is the approach under way in "Taming Apparent singularities in Ore Algebras", a collaboration primarily between Mishna and the Algorithms Project in INRIA France. Chyzak, Mishna and Salvy have successfully used their algorithms on large systems originating from statistical mechanics, in particular certain Fuchsian differential equations arising from polygon enumeration. These algorithms allow one to find different differential equations that may better suit a users need, say one with no apparent singularities.

The second project is the development of direct combinatorial models that are well suited to computational applications. Eric Fusy, a post doc working under the direction of Mishna has made considerable progress on defining new operators in a combinatorial calculus that allow one to randomly generate at uniform wide new classes of combinatorial objects. For example, the results in [15] lead to the success of [19], a promising new technique for enumerating planar graphs, and may well lead to improved algorithms for unlabelled graphs. We also intend to examine how to use this calculus to discover information about reconstructions of ancestral genomic histories. This is done using a particular data structure (PQ-Trees) which is well suited to this context.

3.4.3. Efficient tools for arbitrary precision numeric computation.

We are interested in providing efficient tools for arbitrary precision computation of one-dimensional integrals and very high precision (over 100 digits in 2-5 dimensions) multi-dimensional integrals that arise in many areas of mathematical physics. Such precision is necessary, for example, to apply integer relation methods to identify relevant constants. In the past twelve months considerable success has been achieved on a variety of physically meaningful problems. A particular accomplishment featured in *IOP Select* is [3] where substantial progress was made in quantum field theory and statistical mechanics.

Bailey, Borwein and Crandall [2] also showed the existence of a phase-transition value above which loosely couple-oscillators self-organize as the coupling-level increases, as conjectured by Quinn-Rand-Strogatz. Indeed Strogatz at Cornell requested our assistance. Key to the proof was experimental identification of the transition value as the smallest positive zero of the Hurwitz-zeta function $\zeta(1/2, x/2)$.

At this point, as described in [3], we have reached the boundary of problems for which doubly-exponential substitution integration works in more that two variables—even with massive symbolic pre-computation and highly parallel implementation of the resulting integrals. Along with a PhD student, Ye at Dalhousie, we are now integrating recent ideas from sparse-grid integration (a major workshop on such hybrid techniques will be held in Sydney in Feb 2009). Attention continues to be paid to the correlate special function algorithms needed to make such integration efficiently parallelizable.

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

The principal problem we plan to study is that of relating parameterizations of two curves. If the curves are parameterized (in time) very differently, then they may be far apart in the distance induced by a simple functional inner product.

One appoach to overcome this problem is to use Legendre-Sobolev inner products $\langle f, g\rangle = \int fgdt + \mu \int f'g'dt$ in place of Legendre inner products. This will cause curves that certain geometric features aligned to be measured as closer to each other.

A second approach is to generalize the notion of "dynamic time warping," which is a sequence alignment algorithm that finds a correspondence between sample points on the two curves. If instead we consider the ink traces as parameterized curves, we may define "continuous dynamic time warping" as a problem in variational calculus, varying the function relating the parameterizations of the two curves. It appears that the continuous problem may give a better solution than the discrete problem and have a lower computational complexity.

Our plan for the next two years is to investigate the mathematical structure of these two methods and to evaluate their effectiveness for classification based on coordinate curves and based on integral and joint invariant curves.

References

[1] A. Arnold and M. Monagan. Computing cyclotomic polynomials of very large height. Submitted Oct. 2008 to *Mathematics of Computation*.

[2] D. Bailey, J. Borwein and R. Crandall, Resolution of the Quinn-Rand-Strogatz constant of nonlinear physics, *Experimental Math*, accepted Jan. 2008.

[3] D. Bailey, J. Borwein, D. Broadhurst and L. Glasser, Elliptic integral representation of Bessel moments, *J. Phys. A: Math. Theor*, 41 (2008), 5203-5231.

[4] M.A. Barkatou and E. Pflügel, An Algorithm Computing the Regular Formal Solutions of a System of Linear Differential Equations. *Journal of Symbolic Computation*, **28**(4–5) (1999) 569–587.

[5] B. Beckermann and G. Labahn, A uniform approach for Hermite Padé and simultaneous Padé approximants and their matrix generalizations, *Numerical Algorithms 3* (1992), 45–54.

[6] B. Beckermann & G. Labahn, Recursiveness in Matrix Rational Interpolation Problems, *J. Comput. Appl. Math.* **77** (1997) 5–34.

[7] B. Beckermann and G. Labahn, Fraction-free Computation of Matrix Gcd's and Rational Interpolants, *SIAM Journal of Matrix Analysis and its Applications*, **22**:1 (2000) 114–144.

[8] B. Beckermann, G. Labahn and G. Villard, Normal Forms for General Polynomial Matrices, *Journal of Symbolic Computation*, **41**(6) (2006) 708–737.

[9] A. Bostan, B. Salvy and É. Schost, *Fast conversion algorithms for orthogonal polynomials*. Accepted in *Linear Algebra and its Applications*.

[10] F. Boulier, F. Lemaire *Differential Algebra and System Modeling in Cellular Biology* In *Algebraic Biology*, LNCS, 5147 Springer 2008.

[11] A. Bostan, C.-P. Jeannerod and É. Schost, *Solving structured linear systems with large displacement rank*. *Theoretical Computer Science*, in press.

[12] A. Bostan, B. Salvy and É. Schost, *Power series composition and change of basis*. Proceedings of ISSAC'08 (ACM International Symposium on Symbolic and Algebraic Computation), ACM Press, 2008, pp. 269–276.

[13] E. Cartan. *Les problèmes d'équivalence*, volume 2 of *oeuvres complètes*, pages 1311–1334. Gauthiers–Villars, Paris, 1953.

[14] M. Bousquet-Melou and M. Mishna, Walks with small steps in the quarter plane arXiv:0810.4387 (October 2008).

[15] M. Bodirsky, É. Fusy, M. Kang and S. Vigerske, An unbiased pointing operator for unlabeled structures, with applications to counting and sampling. *Proc. SODA 2007* 356–365.

[16] O.A. Carvajal, F.W. Chapman and K.O. Geddes, *Hybrid Symbolic-Numeric Integration in Multiple Dimensions via Tensor-Product Series*. Proceedings of ISSAC'05 Manuel Kauers (ed.), ACM Press, pp. 84–91, 2005.

[17] O.A. Carvajal, *A Hybrid Symbolic-Numeric Method for Multiple Integration Based on Tensor-Product Series Approximations*. Master's Thesis, Department of Computer Science, University of Waterloo, 2004.

[18] F. W. Chapman, *Generalized Orthogonal Series for Natural Tensor Product Interpolation* Ph.D Thesis, Department of Applied Mathematics, University of Waterloo, 2003.

[19] G. Chapuy, E. Fusy, M. Kang and B. Shoilekova, A Complete Grammar for Decomposing a Family of Graphs into 3-connected Components, submitted Aug 2008.

[20] C. Chen, O. Golubitsky, F. Lemaire, M. Moreno Maza, and W. Pan. *Comprehensive Triangular Decomposition*, volume 4770 of *LNCS*, pages 73–101. Springer Verlag, 2007.

[21] C. Chen, F. Lemaire, L. Liyun, M. Moreno Maza, W. Pan and Y. Xie. The ConstructibleSetTools and ParametricSystemsTools modules of the `RegularChains` library in MAPLE, In *Proc. of the International Conference on Computational Science and Applications*, IEEE Computer Society, pp 342–352, 2008.

[22] C. Chen, M. Moreno Maza, W. Pan, and Y. Xie. On the verification of polynomial system solvers. In *Frontiers of Computer Science in China*, 2(1): 55–66 (2008).

[23] L. Chen and M. Monagan. Algorithms for Solving Linear Systems over Cyclotomic Fields. Submitted to the Journal of Symbolic Computation, February 2008.

[24] H. Cheng and G. Labahn, Output-sensitive Modular Algorithms for Polynomial Matrix Normal Forms, *Journal of Symbolic Computation*, **42**(7) (2007) 733–750.

[25] H. Cheng and G. Labahn, Modular Computation for Matrices of Ore Polynomials. *Proc. of WWCA: Conference in honour of the 60th birthday of Sergei Abramov* , World Scientific, (2007) 43–66.

[26] R.M. Corless, **S. Ilie** & G. Reid (2006). *Computational complexity of numerical solution of polynomial systems*. Proc. of Transgressive Computing 2006, Granada, 405–408.

[27] X. Dahan, M. Moreno Maza, É. Schost, and Y. Xie. On the complexity of the D5 principle. In *Proc. of* Transgressive Computing, Univ. of Granada, 2006.

[28] X. Dahan, X. Jin, M. Moreno Maza, and É Schost. Change of ordering for regular chains in positive dimension. In *Theor. Comput. Sci.* 392(1–3): 37–65, 2008.

[29] X. Dahan, M. Moreno Maza, É. Schost, W. Wu, and Y. Xie. Lifting techniques for triangular decompositions. In *ISSAC'05*, pages 108–115. ACM Press, 2005.

[30] P. Davies, H. Cheng and G. Labahn, Computing Popov Form of General Ore Polynomial Matrices. *Proc. of Milestones in Computer Algebra (MICA)*, (2008) 149–156.

[31] J.-G. Dumas, C. Pernet and W. Zhou, Memory efficient scheduling of the Strassen-Winograd matrix multiplication algorithm, Submitted to SIAM Journal of Matrix Analysis (Sept 2007)

[32] A. Filatei, X. Li, M. Moreno Maza, and É. Schost. Implementation techniques for fast polynomial arithmetic in a high-level programming environment. In *Proc. ISSAC'06*, ACM Press, pp. 93–100, 2006.

[33] M. V. Foursov and M. Moreno Maza. *On Computer-Assisted Classification of Coupled Integrable Equations*. J. Symb. Comput., 33: 647–660, 2002.

[34] K.O. Geddes, *Generating Efficient Numerical Evaluation Routines for Bivariate Functions via Tensor Product Series*. Invited presentation, SNSC '08 (4th International Conference on Symbolic and Numerical Scientific Computing), Hagenberg, Austria, Jul 2008.

[35] O. Golubitsky, M. Kondratieva, M. Moreno Maza and A. Ovchinnikov. A bound for the Rosenfeld-Gröbner algorithm, 2006. *J. of Symbolic Computation*. Volume 43 , Issue 8 (August 2008) Pages 582–610.

[36] I.G. Lisle and G.J. Reid (2006). *Symmetry classification using noncommutative invariant differential operators*. Foundations of Computational Mathematics **6**(3), 353–386.

[37] S. Ilie, R.M. Corless and G.J. Reid (2006). *The numerical solution of index-1 differential algebraic equations is of polynomial cost*. Numerical Algorithms **41**, 161–171.

[38] M. Javadi and M. Monagan. A Sparse Modular GCD Algorithm for Polynomial GCD Computation over Algebraic Function Fields. *Proceedings of ISSAC '07*, ACM Press, pp. 187–194, July 2007.

[39] J. de Kleine, M. Monagan and A. Wittkopf. Algorithms for the Non-monic case of the Sparse Modular GCD Algorithm. *Proc. of ISSAC '05*, ACM Press, pp. 124–131, 2005.

[40] E. Kaltofen and Z. Zheng. On exact and approximate interpolation of sparse rational functions. *Proc. ISSAC '07*, ACM Press, pp. 203–210, 2007.

[41] F. Lemaire, M. Moreno Maza, and Y. Xie. The `RegularChains` library. In Ilias S. Kotsireas, editor, Maple Conference 2005, pages 355–368, 2005.

[42] F. Lemaire, M. Moreno Maza, W. Pan and Y. Xie. When does (T) equal Sat(T)? *Proc. ISSAC '08*, ACM Press, pages 207–214, 2008.

[43] X. Li and M. Moreno Maza. Efficient implementation of polynomial arithmetic in a multiple-level programming environment. In A. Iglesias and N. Takayama, editors, *Proc.* International Congress of Mathematical Software - ICMS 2006, pages 12–23. Springer, 2006.

[44] X. Li and M. Moreno Maza. Multithreaded parallel implementation of arithmetic operations modulo a triangular set. In *Proc. Parallel Symbolic Computation'07 (PASCO'07)*, ACM Press, p. 53-59, 2007.

[45] X. Li, M. Moreno Maza, and É. Schost. Fast Arithmetic for triangular sets: from theory to practice, *J. Symbolic Comp.*, in press, 2008.

[46] X. Li, M. Moreno Maza, and É. Schost. On the virtues of generic programming for symbolic computation. In *ICCS'07*, volume 4488 of *Lecture Notes in Computer Science*, pages 251–258. Springer, 2007.

[47] X. Li, M. Moreno Maza, R. Rasheed and É. Schost. The `modpn` library: bringing fast polynomial arithmetic into MAPLE. In *Proc. of Milestones in Computer Algebra*, pages 72–60, 2008.

[48] X. Li, M. Moreno Maza, R. Rasheed and É. Schost. High-Performance Symbolic Computation in a Hybrid Compiled-Interpreted Programming Environment In *Proc. 2008 International Conference on Computational Sciences and Its Applications*, pages 331–341. IEEE Computer Society, 2008.

[49] S. Liang, D.J. Jeffrey, M. Moreno Maza, *The complete root classification of a parametric polynomial on an interval. Proceedings of ISSAC '08*, D.J. Jeffrey (ed.), ACM Press, New York, 2008, pp 189–196.

[50] S. Liang, D.J. Jeffrey, Unconstrained parametric minimization of a polynomial: approximate and exact. *Proceedings ASCM 2007*. LNAI 5081, Springer 2008, pp22–31.

[51] M. Mishna, Classifying lattice walks in the quarter plane, to appear in JCTA (preliminary version presented at FPSAC 2007)

[52] M. Mishna and A. Rechnitzer, Two non-holonomic lattice walks in the quarter plane, to appear in TCS-A, 2008.

[53] M. Mishna and M. Zabrocki, Analytic aspects of the shuffle product, STACS 2008.

[54] M. Monagan and R. Pearce. Sparse Polynomial Pseudo Division Using a Heap. *J. Symb. Comput.*, submitted, September 2008.

[55] M. Monagan and R. Pearce. Polynomial Division using Dynamic Arrays, Heaps, and Packed Exponent Vectors. *Proceedings of CASC '07* Springer-Verlag LNCS **4770**, pp. 295–315, 2007.

[56] M. Moreno Maza, G.J. Reid, R. Scott & W. Wu (2007). *On Approximate Triangular Decompositions in Dimension Zero*. Journal of Symbolic Computation **42**(7), 693–716.

[57] M. Moreno Maza, G.J. Reid, R. Scott & W. Wu (2007). *On Approximate Linearized Triangular Decompositions*. In Symbolic-Numeric Computation, edited by Dongming Wang and Lihong Zhi, Birkhauser, 279–298.

[58] M. Moreno Maza and Y. Xie. Component-level parallelization of triangular decompositions. In *Proc. PASCO'07*, ACM Press, p. 69–77, 2007.

[59] M. Moreno Maza, B. Stephenson, S. M. Watt and Y. Xie. Multiprocessed parallelism support in ALDOR on SMPs and Multi-cores. *Proc. PASCO'07*, ACM Press, p. 60–68, 2007.

[60] T. Mulders and A. Storjohann, On lattice reduction for polynomial matrices. *Journal of Symbolic Computation*, (2003) **35**(4), 377–401.

[61] S. Neut and M. Petitot. La géométrie de l'équation $y''' = f(x, y, y', y'')$. *C. R. Acad. Sci. Paris*, (335):515–518, 2002.

[62] Z. Olesh and A. Storjohann, The vector rational function reconstruction problems. *Proc. of WWCA: Conference in honour of the 60th birthday of Sergei Abramov* , World Scientific, (2007) 137–149.

[63] G.J. Reid, J. Verschelde, A.D. Wittkopf & W. Wu (2005). *Symbolic-Numeric Completion of Differential Systems by Homotopy Continuation*. Proc. ISSAC 2005, ACM, 269–276.

[64] G.J. Reid & L. Zhi (2008). *Solving polynomial systems via symbolic-numeric reduction to geometric involutive form*. To appear in J. of Symbolic Computation (accepted 14-10-07).

[65] A. Sexton and V. Sorge, Abstract matrices in symbolic computation, *Proc. of ISSAC '06*, ACM Press, (2006) 318–325.

[66] A. P. Sexton, V. Sorge and S. M. Watt, *Abstract Matrix Arithmetic*. Proc. 10th International Symposium on Symbolic and Numeric Algorithms for Scientific Computing, (SYNASC 2008), Sept 26–29 2008, Timisoara Romania, (accepted).

[67] E. Smirnova and S.M. Watt, An Pen-Based Mathematical Environment: Mathink, Ontario Research Centre for Computer Algebra (ORCCA), University of Western Ontario, Research Report TR-06-05 (16 pages), 2006.

[68] A. Storjohann, On the complexity of inverting integer and polynomial matrices, July 22, 2008.

[69] D. Tausky, G. Labahn, E. Lank and M. Marzouk, Managing Ambiguity in Mathematical Matrices, *Proceedings of the 4th Eurographics workshop on Sketch-based interfaces and modeling (SBIM 2007)*, (2007) 115–122.

[70] X. Wang, *Automated Generation of Numerical Evaluation Routines for Bivariate Functions via Tensor Product Series*. Master's Thesis, David R. Cheriton School of Computer Science, University of Waterloo, 2008.

[71] W. Wu & G.J. Reid (2006). Application of Numerical Algebraic Geometry and Numerical Linear Algebra to PDE. *Proc. ISSAC '06*, ACM Press, 345–352, 2006.

[72] W. Wu & G.J. Reid (2007). Symbolic-numeric Computation of Implicit Riquier Bases for PDE. *Proc. ISSAC '07*, ACM Press, 377–386, 2007.

[73] W. Wu, G.J. Reid & S. Ilie (2008). *Implicit Riquier bases for PDAE and their semi-discretizations*. To appear Journal of Symbolic Computation (accepted 25-04-2008).

[74] S. M. Watt, *Functional Decomposition of Symbolic Polynomials*. Proc. International Conference on Computatioanl Sciences and its Applications, (ICCSA 2008), June 30-July 3 2008, Perugia, Italy, IEEE Computer Society, pp. 353-362.

[75] S. M. Watt, *Symbolic Polynomials with Sparse Exponents*. Proc. Milestones in Computer Algebra: a Conference in Honour of Keith Geddes' 60th Birthday, (MICA 2008), May 1-3 2008, Stonehaven Bay, Trinidad and Tobago, University of Western Ontario ISBN 978-0-7714-2682-7, pp. 91–97.

[76] S. M. Watt, *Functional Decomposition of Symbolic and Multivariate Laurent Polynomials* **(Invited)**. Proc. 10th International Symposium on Symbolic and Numeric Algorithms for Scientific Computing, September 26-29 2008, Timisoara, Romania, (to appear).

[77] Lu Yang, Xiaorong Hou, and Bican Xia. A complete algorithm for automated discovering of a class of inequality-type theorems. *Science in China, Series* **F**, 44(6):33–49, 2001.

[78] W. Zhou, J. Carette, D.J. Jeffrey, M.B. Monagan, Hierarchical representations with signatures for large expression management. *Proc. AISC '06*, LNCS **4120**, Eds. J. Calmet, T.Ida, D. Wang. pp. 254–268, Springer 2006.

[79] W. Zhou, D.J. Jeffrey, *Fraction-free matrix factors: new forms for LU and QR factors.* Frontiers in Computer Science in China, Vol 2, no. 1, pp. 67–80, 2008.

[80] W. Zhou, J. Carette, D.J. Jeffrey, M.B. Monagan, *Hierarchical representations with signatures for large expression +management.* Proceedings AISC 2006. LNCS 4120, pp. 254–268, Springer 2006.

4. Development of Highly Qualified Personnel

a) Describe HQP involvement with partner organizations.

The personnel trained through the MOCAA consortium are in general PDFs, graduate students at the Masters and PhD level, and senior undergraduate students. They interact with faculty and industry personnel through

- professional conferences (see Section 5 Networking),
- MITACS internships – which means 2 months on site at the company,
- the three group meetings at Simon Fraser, Waterloo, and Western,
- the monthly seminar organized at Waterloo and Western,
- software training provided by company personnel, and our
- MOCAA project workshops (June 2007 at Winnipeg, December 2007 at SFU and May 2008 at UWO)

Students and PDFs are encouraged to give talks and demo software at the biweekly group meetings at SFU, UW and UWO, present posters and demo software at the ACA, MITACS, CECM, ECCAD, ISSAC, PASCO, SNC, and MAPLE conferences/meetings. Also, this year we held a project workshop May 6-9 at UWO which provided an opportunity for HQP from Vancouver to meet and interact with Maplesoft company personnel.

b) Describe training activities initiated (summer schools, tutorials, curriculum development, etc.)

A first course in computer algebra is offered regularly at Simon Fraser, Waterloo, and Western. Faculty also regularly provide additional *graduate* courses in computer algebra and related topics for HQP to take.

- Algebraic Geometry and Gröbner Bases. SFU, spring 2007, Monagan.
- Topics in Symbolic Computation, UW, winter 2007, Giesbrecht and Labahn.
- Topics in Computer Algebra. SFU, summer 2007, Monagan.
- Algorithms for recurrences differential equations, and the automatic proof of identities. UWO, spring 2008, Schost.
- Cryptography and computational number theory. SFU, fall 2008, Monagan.
- Foundations of computational algebra. UWO, Schost (fall 2008) and Moreno Maza (winters 2006 and 2007).
- Parallel scientific computing: models, algorithms and implementation. UWO, winters 2006 and 2007, Moreno Maza.

Each summer at SFU, Borwein, Mishna, and Monagan have summer NSERC fellows (senior undergraduate students) working on various computer algebra/Maple related projects to attract them to graduate school in this area. The students in 2007 and 2008 were:

Andrew Arnold – computational number theory (cyclotomic polynomials)
Jamie Lutley – computational combinatorics (enumeration of lattice paths).
Rebecca Nie – computational combinatorics (enumeration of lattice paths).
Natasha Richardson – health modelling.
Amy Wiebe – health modelling.
Asif Zaman – computational group theory (algorithms and visuals)

5. Networking

In this section we provide some details of future planned networking events including those with industry partners (involving diffusion of results, technology transfer, collaborative research or industry linkage), both within the project and within MITACS themes.

Regular Meetings and Seminars.

The Computer Algebra Group at SFU, the Symbolic Computation Group at UW, and the Symbolic Computation Lab at UWO hold weekly or biweekly meetings/seminars. The two groups at Waterloo and Western collectively form the Ontario Research Center for Computer Algebra (ORCCA).

ORCCA meets on the second Friday of every month either in Waterloo or London with Jacques Carette from McMaster in Hamilton and members from the Math group at Maplesoft for a seminar, followed by a poster session and networking over lunch. Individuals from the two groups at UW and UWO also meet regularly with people at Maplesoft.

Conference Involvement.

Faculty and students on the project, and personnel from the Math group at Maplesoft attend the following annual scientific meetings.

- The *International Symposium on Symbolic and Algebraic Manipulation* (ISSAC). The 2007 meeting was organized at Waterloo, Canada by Keith Geddes, George Labahn, Mark Giesbrecht and Arne Storjohann. The 2008 meeting was held in July in Hagenberg, Austria. Michael Monagan organized the Software Systems session. The 2009 meeting will be in Seoul, South Korea.

- Joint MITACS/CMS Conference. The 2008 meeting was held in June at Dalhousie university in Halifax, Canada. The 2009 meeting will be held in Fredericton, New Brunswick. We plan to organize a computer algebra session for our project.

- *Applications of Computer Algebra* (ACA). The 2008 meeting was held at RISC Linz in Hagenberg, Austria. Stephen Watt co-organized the "Compact Computer Algebra" session and Illias Kotsireas co-orgaznized the "Gröbner Bases and Applications" session. The 2009 meeting will be held in Montreal, Canada. Many project members are expected to participate in this meeting given its location.

- *Conferences in Intelligent Computer Mathematics (CICM)* The 2008 meeting was held in Birmingham, England. The 2009 meeting will be held in Grand Bend, Ontario. Stephen Watt is chairing the 8th International Conference on Mathematical Knowledge Management. Stephen Watt will co-chair Compact Computer Algebra 2009. George Labahn and Stephen Watt will co-chair Pen-Based Mathematical Computation 2009.

6. Knowledge Exchange and Technology Transfer

"Describe the Intellectual Property (IP) generated, including software, since the last Project CV (November 2006). Describe the state of the IP and its readiness for use by other institutions or industry, if applicable. Indicate whether the IP has or will be licensed. Are patents or other protection being sought?"

The IP generated by our MOCAA project consists mainly of MAPLE software which includes MAPLE programs, C programs, documentation, MAPLE demo worksheets, and test files.

Proprietary software packages.

The contributions listed here have been contributed in the period November 1, 2006 through November 1, 2008 under contracts with Maplesoft. Contributions that have already been integrated into MAPLE 12 or MAPLE 13 are identified.

1. M. Javadi and M. Monagan (2008), MAPLE 13. MAPLE code for computing the GCD of two multivariate polynomials over an algebraic function field with zero or more parameters (hence includes number fields) where the field is presented with one or more field extensions. The algorithm uses a sparse interpolation.

2. C. Percival, P. Borwein and A. Wittkopf (2007), MAPLE 12. An implementation of a self initializing quadratic sieve algorithm for factoring integers. Increases the size of integers MAPLE can factor in three hours from 60 digits to 90 digits.

3. M. Monagan (2007 & 2008), MAPLE 13. A modular algorithm for solving linear systems involving roots of unity.

4. A. Erickson, M. Ghebleh, M. Monagan, A. Wittkopf (2007, 2008), MAPLE 12 & 13. Addition of tools for generating random regular graphs, non-isomorphic graphs with m edges and n vertices, a graph isomorphism test, etc., were integrated into MAPLE's `GraphTheory` package.

5. G. Fee, V. Dhabaghian, M. Monagan (2007). MAPLE 13 (submitted). A data base of finite groups including all groups of order up to 200 and special groups from the Atlas of Finite Groups.

6. Alan Meichsner and Peter Borwein (2008), MAPLE 13. A complex version for Bailey and Ferguson's PSLQ algorithm for searching for integer relations.

7. Roman Pearce and Michael Monagan (2008), MAPLE 13. New heap based algorithms for multivariate polynomial multiplication and division over finite fields. We are awaiting a contract before we integrate our code for multiplication and division over the integers – which will have a greater impact than the finite field case.

8. George Labahn (2008), MAPLE 13 (submitted). MAPLE code for the integration of expressions of the form $x^m exp(x^k)^n f(x)$ with f either Ci, Si, erf, Fresnel C etc, or products of pairs of these functions, for some m, n (arbitrary) and k (usually 1 or 2).

9. C. Chen, F. Lemaire, L. Li, M. Moreno Maza, W. Pan and Y. Xie. The `ConstructibleSetTools` module of the `RegularChains` library in MAPLE 12. **5,000** lines of MAPLE code. This package provides an extensive set of commands to compute with constructible sets, parametric or not, in characteristic zero or not.

10. C. Chen, F. Lemaire, L. Li, M. Moreno Maza, W. Pan and Y. Xie. The `ParametricSystemTools` module of the `RegularChains` library in MAPLE 12. **3,000** lines of MAPLE code. This package provides an implementation of the *Comprehensive Triangular Decomposition* (CTD) of a parametric constructible set. As an application, the command `ComplexRootClassification` determines the possible numbers of (complex) solutions of an input parametric system together with the corresponding conditions on the parameters.

11. C. Chen, F. Lemaire, M. Moreno Maza, W. Pan, B. Xia, and Y. Xie. The `SemiAlgebraicSetTools` module of the `RegularChains` library. MAPLE 13. Together with the command `ParametricSystemTools:-RealRootClassification`, it amounts to **25,000** lines of MAPLE code. This provides a variety of tools for studying the real solutions of polynomial systems, including real root isolation and counting, partial cylindrical decomposition, real root classification.

12. X. Li and M. Moreno Maza (developers). É. Schost and W. Pan (contributors). The `modpn` library. MAPLE 13. `modpn` is a C and MAPLE library dedicated to fast arithmetic for multivariate polynomials. The main objective of `modpn` is to provide highly efficient routines implemented in C for supporting the implementation of modular methods in MAPLE. `modpn` amounts to **35,000** lines of C code and **5,000** lines of MAPLE code.

13. X. Li and M. Moreno Maza. `The FastArithmeticTools` module of the `RegularChains` library. MAPLE 13. **5,000** lines of MAPLE code. This package provides modular methods, with fast arithmetic support from the `modpn` library, for the core operations of the `RegularChains` library. The current version works in prime characteristic only. The characteristic zero case is work in progress for the next release.

Other: publically available software.

1. Jon Borwein (2008). Updated version of the inverse symbolic calculator.
Available on-line at `http://glooscap.cs.dal.ca:8087/standard`

2. Roman Pearce and Michael Monagan (2008). `sdmp` is a C library of high performance software for computing with multivariate polynomials over $\mathbb{Z}$ and $\mathbb{Z}_n$. The algorithms for multiplication and division use binary heaps with chaining to obtain good locality.

3. *fflas-ffpack*: A new in-place reduction to matrix multiplication for the computation of an inverse of a square matrix over a finite field. Implemented and included in the LinBox library by C. Pernet (2007)

7. Additional Information.

Project Management

The current management team is, and will continue to be

Marc Moreno Maza, Western,

George Labahn, Waterloo (Eastern Group Leader) and,

Michael Monagan, Simon Fraser (Western Group Leader).

with Labahn and Monagan co-leading the project. Michael Monagan was promoted to full professor in 2008 and Marc Moreno Maza was promoted to associate professor in 2008.

Relationship to other Research Support

All subprojects except subprojects for 3.2.4, 3.4.1, 3.4.2, and 3.4.3 are new projects not appearing on any other grant applications. Subprojects 3.4.1 (zeros and poles of Padé approximants) and 3.4.3 (efficient tools for arbitrary precision numeric computation) are continuations of our existing mitacs grant projects. Subprojects 3.2.4 (numerical algebraic geometry) has an overlap with the NSERC grant of Reid. The first project on subproject 3.4.2 (automatic combinatorics) is new but the second one has an overlap with a CNRS grant of Mishna. For these latter two subprojects the mitacs support will focus entirely on software issues (computer implementations and experimentation).