Computer Algebra and Computer Algebra Systems
L'algèbre computationnelle et systèmes d'algèbre computationnelle

(Org: **Michael Monagan** (SFU))

OLEG GOLUBITSKY, Ontario Research Centre for Computer Algebra, London, Ontario, Canada, N6A 5B7

Implementation of Arithmetics in Aldor

ALDOR is a programming language designed for implementing symbolic computation algorithms. On the one hand, it allows to encode mathematical structures and algorithms in a natural way, exhibiting their essence and hiding various lower-level programming issues. On the other hand, the ALDOR compiler allows to use machine resources much more efficiently than computer algebra systems.

The efficiency of implementation of most symbolic methods significantly depends on the efficiency of basic arithmetic operations. The latter include computations with fixed size integers, residue classes modulo an integer, arbitrary precision integers, single and double floating point numbers, and arbitrary precision floating point numbers.

We will discuss a new implementation of arithmetic types in Aldor, which combines and extends two existing ALDOR libraries, AXLLIB and LIBALDOR. In this implementation, we refine the hierarchy of arithmetic types in an attempt to reflect structural relationships between them more naturally. We also solve the issue of compatibility with multiple platforms, on which the machine arithmetic types may have different sizes. In particular, we develop a general framework for doubling the size of a given machine integer type or restricting it to an integer of smaller size (including a new algorithm for dividing double integers). We show how the ALDOR optimizer translates this generic object-oriented code into highly efficient machine code by in-lining small subroutines and unfolding records.

This work is part of a larger project on unifying existing ALDOR libraries and providing one efficient general-purpose library for implementing computer algebra algorithms.

This is joint work with Stephen Watt.

HOWARD CHENG, University of Lethbridge, 4401 University Drive, Lethbridge, Alberta

Time- and Space-Efficient Evaluation of Some Hypergeometric Constants

The current best practical algorithms for the numerical evaluation of hypergeometric constants such as $\zeta(3)$ to d decimal digits have time complexity $O\big(M(d)\log^2 d\big)$ and space complexity of $O(d\log d)$ or $O(d)$. Following work from Cheng, Gergel, Kim and Zima, we present a new algorithm with the same asymptotic complexity, but more efficient in practice. Our implementation of this algorithm improves over existing programs for the computation of π, and we announce a new record of 2 billion digits for $\zeta(3)$.

This work was done jointly with Eugene Zima (Wilfrid Laurier University), Guillaume Hanrot, Emmanuel Thomé, and Paul Zimmermann (INRIA, France).

VAHID DABBAGHIAN-ABDOLY, Simon Fraser University, Burnaby, BC, V5A 1S6

Implementation of Cellular Automata Models on Maple

Cellular Automata is a discrete dynamical model providing an excellent platform for performing behaviour of complex systems with the help of only local information. In this talk I will present a cellular automata model for describing the dynamics of urban transformations such as spread of infectious disease and crime. In particular I will show implementation of some cellular automata models on Maple.

RON FERGUSON, Simon Fraser University
Search Algorithms for Low Autocorrelation Sequences

Low autocorrelation sequences have been studied both for their number theoretic properties and for applications in communications engineering. Such problems as the merit factor problem, the peak sidelobe problem and the existence of barker sequences are both computationally challenging and have deep theoretical implications. The problems may be discrete, as in the cases of binary or n-phase sequences, or continuous, where the sequences are unimodular. We will describe algorithms used to obtain computational results, combining both continuous and discrete approaches as well as both exhaustive and stochastic methods.

MARK GIESBRECHT, University of Waterloo, School of Computing Science
Complexity and Practicality in Sparse Matrix Computation

We present new algorithms for operations related to sparse matrices which are asymptotically faster than those known previously and quite practical in some cases. Sparsity is designated by requiring a fast matrix-vector product—typically quasi-linear time—which captures many traditional families of sparse or structured matrices. We exhibit a probabilistic algorithm which finds the (dense) inverse of such a sparse matrix with $O(n^{2.27})$ field operations. This is surprising in that it is less than the cost of dense matrix multiplication and inversion, which was the previously best known approach to sparse matrix inversion. For sparse integer matrices (with constant sized entries), we show how to solve such systems with $O(n^{2.5})$ machine operations using standard matrix arithmetic. These techniques are shown to be practical at least on some classes of large sparse matrices.

This is joint work with Wayne Eberly, Pascal Giorgi, Arne Storjohann and Gilles Villard.

OLEG GOLUBITSKY, University of Western Ontario
Comprehensive Triangular Decomposition

We introduce the concept of comprehensive triangular decomposition (CTD) for a parametric polynomial system F with coefficients in a field. In broad words, this is a finite partition of the the parameter space into regions, so that within each region the "geometry" (number of irreducible components together with their dimensions and degrees) of the algebraic variety of the specialized system $F(u)$ is the same for all values u of the parameters. We propose an algorithm for computing the CTD of F. It relies on a procedure for solving the following set theoretical instance of the coprime actorization problem.

Given a family of constructible sets, $A_1, \ldots, A_s$ compute a family $B_1, \ldots, B_t$ of pairwise disjoint constructible sets such that, for all index i, the set A_i writes as a union of some of the $B_1, \ldots, B_t$. We report on an implementation of our algorithm computing CTDs, based on the RegularChains library in Maple. We provide comparative benchmarks with Maple implementations of related methods for solving parametric polynomial systems. Our results illustrate the good performances of our CTD code.

This is joint work with Changbo Chen, Francois Lemaire, Marc Moreno Maza and Wei Pan.

BRYAN KRAWETZ, MapleSoft, 615 Kumpf Drive, Waterloo, Ontario
Faster polynomial arithmetic over algebraic number and function fields in Maple 11

In Maple, the command *evala* is used to evaluate expressions in an algebraic number (or function) field. When conditions are appropriate, evala relies on *recden*, a Maple library designed to work with dense polynomials efficiently. The recden library was developed by M. Monagan and his group at Simon Fraser University.

For Maple 11, the core routines of the recden library were converted to internal kernel routines. In this talk, we will give an overview of how these new routines are implemented, as well, we will discuss the performance benefits they provide over the pure library implementation.

SCOTT MACLEAN, University of Waterloo
Mathematical Symbol Recognition in the MathBrush System

The MathBrush project at the University of Waterloo is a vehicle for investigating issues which arise in the construction of pen-based interfaces for computer algebra systems. Our prototype application currently comprises several components including an interface module, a pretty-printer, a symbol recognizer, and a structural analyzer. Here, we outline the MathBrush system in its entirety but focus on the symbol recognition module, demonstrating particular difficulties associated with recognizing symbols in the context of mathematical expressions as opposed to English text and describing our techniques for addressing these difficulties. We detail our experiments with various recognition algorithms, our approach to stroke grouping in a mathematical context, and aspects of our recognizer intended to improve performance and usability. Ongoing work incorporating feedback between the symbol recognition and structural analysis modules to improve recognition accuracy is also described.

MATT MALENFANT, University of Western Ontario
Better Evaluation Points for the Interpolation of Sparse Symbolic Polynomials

Symbolic polynomials, whose exponents themselves are integer-valued multivariate polynomials, arise often in algorithm analysis. Unfortunately, modern computer algebra systems do not provide ample support for said algebraic structures. Basic operations involving symbolic polynomials are indeed trivial (addition, multiplication, derivatives); however, other crucial operations remain much more difficult, such as factorization, GCD, Gröbner Bases.

The exponent variables can be evaluated, producing Laurent Polynomials which can then be interpolated back to their original form. For a t-sparse exponent polynomials of p variables and degree d, sparse interpolation can be used to reduce the required number of images from $O\big((d+1)^p\big)$ to $O(pdt)$.

In practice, selecting random, or small evaluations will often result in polynomials of very large degree. In this talk, we will describe a method of selecting evaluation points, that will minimize the maximum degree of the input symbolic exponents.

MICHAEL MONAGAN, University of Western Ontario
Solving Linear Systems over Cyclotomic Fields

We present three algorithms for solving a linear system $Ax = b$ over a cyclotomic field. If $m(z)$ is the minimal polynomial for the field, a cyclotomic polynomial, then what makes this problem of special interest is that it is relatively easy to find primes which split $m(z)$ into linear factors. This means we can solve $Ax = b$ modulo a prime at each root of $m(z)$, potentially in parallel.

Our first algorithm uses Chinese remaindering and rational reconstruction. Our second algorithm uses linear p-adic lifting and rational reconstruction. A third approach is to express the solutions as ratios of determinants. This can be a factor of $d = \deg m(z)$ more compact.

In the talk we will present the algorithms and improvements made to improve the complexity of the reconstruction, and, for the p-adic lifting approach, computation of the error.

We have implemented the three algorithms in Maple. We present timings comparing the three algorithms on two sets of benchmarks, firstly, a set of real problems arising from computational group theory. These problems have the property that the size of the rationals in the solution vector x is much smaller than they can be in general. The second set is for problems where the integers in the input are generated at uniformly at random.

This is joint work with Liang Chen at SFU.

ROMAN PEARCE, Simon Fraser University
Sparse Polynomial Arithmetic using Heaps

We present some old and seemingly forgotten algorithms for multiplying and dividing sparse polynomials using heaps. The algorithms do an n-ary merge of all partial products using only a heap of pointers into the input, constructing exactly the terms that appear in the result. The amount of memory required is linear in the size of the smaller input for multiplication or in the quotient or the divisor for division. We also constructed a variant of the division algorithm that is linear time in the size of the quotient, something which is not possible for algorithms based on merging. As a result, some common cases of exact division can be done an order of magnitude faster using an order of magnitude less memory than with a divide and conquer strategy, such as geobuckets. We plan to integrate our C library into the next release of Maple.

This is joint work with Dr. Michael Monagan at Simon Fraser University.

GREG REID, Department of Applied Mathematics, University of Western Ontario, Middlesex College, London, ON, N6A 5B7

Introduction to Symbolic-Numeric Completion Methods for PDE

Differential elimination methods apply a finite sequence of differentiations and eliminations to general systems of PDE to extract potent information about their solutions. Much recent progress has been made in the design and implementation of exact algorithms, applying to exact input sytems, by researchers such as Boulier, Hubert, Mansfied, Seiler, Wittkopf and others. Though powerful, such methods cannot be applied to approximate systems, since the strong underlying use of rankings of partial derivatives, often induces instability, by forcing such methods to pivot on small quantities.

The talk will be an introduction to the new area of symbolic-numeric methods for completion of PDE. Main features include the focus on geometric methods and the use of homotopy-continuation methods for the detection of new constraints by slicing varieties in jet space with random hyperplanes. Our most recent work on this topic will be presented in this talk.

ALLAN WITTKOPF, MapleSoft, 615 Kumpf Drive, Waterloo, Ontario, N2V 1K8

What's new in Maple 11

I will give an overview of the new features in Maple 11, including:

- interface features, such as drawing, and new array plots;
- new packages (Graph Theory, Physics, etc.);
- symbolic computation improvements (Groebner, summation, inequality solving);
- numeric computation improvements (fsolve, numeric integration/summation, ODE solutions);

and several other features.

WENYUAN WU, Department of Applied Mathematics, University of Western Ontario, Middlesex College, London, ON, N6A 5B7

New Progress on Implicit Riquier Bases for PDE

Riquier Bases for systems of analytic PDE are, loosely speaking, a differential analogue of Grobner Bases for polynomial equations. They are determined in the exact case by applying a sequence of prolongations and eliminations to an input system of PDE.

We present a symbolic-numeric method to determine Riquier Bases in implicit form for systems which are dominated by pure derivatives in one of the independent variables and have the same number of PDE and unknowns. The method is successful provided the prolongations with respect to the dominant independent variable have a block structure which is uncovered by Linear Programming and certain Jacobians are non-singular when evaluated at points on the zero sets defined by the functions of the PDE. For polynomially nonlinear PDE, homotopy continuation methods from Numerical Algebraic Geometry can be used to compute approximations of the points.

WEI ZHOU, University of Waterloo

Fraction-free Computation of LCM and GCD by values

Given the values of two univariate polynomials at a set of interpolation points, we examine the problem of computing the values of their least common multiple (LCM) and their greatest common divisor (GCD) at these points. We show that the values of an LCM of two univariate polynomials can be computed directly from the values of the polynomials and the interpolation points without first converting the polynomials to the standard power form. The result is the interpolation data for the LCM of the input polynomials. The values of a GCD can then be computed from the values of the LCM.