INRIA

INSTITUT NATIONAL DE RECHERCHE EN INFORMATIQUE ET EN AUTOMATIQUE

Project-Team SPACES

Solving Problems through Algebraic Computation and Efficient Software

Lorraine - Rocquencourt

THEME SYM

Activity Report

2004

Table of contents

1. Team

SPACES *is a project-team with people working on two sites: at LORIA in Nancy and at LIP6 in Paris. The Nancy subgroup depends from INRIA Lorraine, the Paris subgroup depends from INRIA Rocquencourt.*

Team Leader

Daniel Lazard [professor, University Pierre et Marie Curie (Paris 6, UPMC), seconded to INRIA, until August]

Team Vice-Leader

Paul Zimmermann [research director, INRIA]

Administrative Assistant

David Massot [UPMC, part time]
Hélène Zganic [LORIA, part time, until May 2nd]
Céline Simon [LORIA, part time, from May 3rd to August 6th, and from September 14th]

Staff member (CNRS or INRIA)

Jean-Charles Faugère [research scientist, CNRS]
Guillaume Hanrot [research scientist, INRIA]
Vincent Lefèvre [research scientist, INRIA]
Fabrice Rouillier [research scientist, INRIA]
Emmanuel Thomé [research scientist, INRIA]
Dongming Wang [research scientist, CNRS]

Staff member (University)

Philippe Aubry [assistant professor, UPMC]
Mohab Safey El Din [assistant professor, UPMC]
Philippe Trébuchet [teaching assistant, UPMC]

Technical staff

Patrick Pélissier [junior technical staff, INRIA]

Ph. D. student

Gwenolé Ars [DGA, defense planned in December 2004 or 2005]
Magali Bardet [teaching assistant, defended 2004/12/08]
Jean-Paul Cerri [defense planned in 2005]
Solen Corvez [BDI CNRS, defense planned in 2005]
Laurent Fousse [ENS grant, defense planned in 2006]
Amir Hashemi [Sfere grant, defense planned in 2005]
Sylvain Lacharte [CIFRE grant with Thalès, defense planned in 2006]
Damien Stehlé [ENS grant, defense planned in 2006]

Visiting scientist

Richard Brent [Prof. at Oxford University, during September]

Student intern

Marc Helbling [July-September]
Dahab Hakim [April-September]
Colas Le Guernic [April-September]

2. Overall Objectives

The objectives of the team are well summarized by the development of its acronym: *Solving Problems through Algebraic Computation and Efficient Software*.

The main objective is to solve systems of polynomial equations and inequations. We emphasize on algebraic methods which are more robust and frequently more efficient than purely numerical tools.

The high complexity of the problems which are studied implies that efficient software may only be obtained by associating good theoretical algorithms with carefully designed implementations. Especially we need efficient and well suited arithmetics for integers and rational numbers of arbitrary precision and high precision floating-point numbers. But we need also more exotic arithmetics, like modular and p-adic ones, or even infinitesimal numbers.

Polynomial systems have many applications in various scientific — academic as well as industrial — domains. However much work is yet needed in order to define specifications for the output of the algorithms which are well adapted to the problems. In addition we have frequently to translate the problems into certain forms which are well suited for the resolution. Thus solving problem is an essential part of our research and software development.

The variety of these applications implies that our software needs to be robust. In fact, almost all problems we are dealing with are highly numerically unstable, and therefore, the correctness of the result needs to be guaranteed.

3. Scientific Foundations

3.1. Historical Background

Solving polynomial equations and systems is a longstanding fundamental problem. Let us mention as witness the attempts to solve univariate equations by radicals until Abel and Galois proved (around 1830) that it is impossible, or the fact that the "fundamental theorem of algebra" deals with polynomial equations.

It is only at the end of the 19th century that the first general algorithms for polynomial systems appeared, with the works of Bezout, Sylvester, Kronecker, and Macaulay. However the computations were intractable, due to the complexity of the problem. Therefore the mathematicians gave up this effective approach to put the emphasis on qualitative theoretical results.

With electronic computers and computer algebra systems, the problem of multivariate systems came back to the attention of the researchers around 1970, when polynomial gcd and factorization problems got a satisfactory answer. However, because of the difficulty of the problem, it is only in the second half of the 80's that computers began to solve problems which are really intractable by hand.

3.2. Zero-Dimensional Polynomial Systems

Strictly speaking, a system of polynomial equations is a formula

$$P_1 = 0 \text{ and } P_2 = 0 \text{ and ... and } P_k = 0$$

where the P_i are multivariate polynomials with coefficients in a field K. Such a system is usually represented by the set of polynomials P_i. Solving it consists in finding the values of the indeterminates which satisfy this formula. These values are searched in an algebraically closed field containing K or in the field of the reals if K is a real field[1]. When K is the finite field with q elements, which is usually the case in cryptology, one may prescribe that all the solutions are in K by adding the equation $x^q - x = 0$ for each unknown x.

A system is zero-dimensional if the set of the solutions in an algebraically closed field is finite. In this case, the set of solutions does not depend on the algebraically closed field which is chosen. This case is the only one which is accessible to numerical solvers, and only when the number of equations equals the number of unknowns.

Note that a single univariate polynomial is a very special case of zero-dimensional system, which is especially important because most algorithms express the solutions as functions of the roots of one univariate equation.

[1] The problem of finding the integer or the rational solutions is known to be undecidable.

With the algorithms and software of SPACES, it is now a routine task to solve large zero-dimensional systems, by computing first a Gröbner basis, then deducing a RUR (*Rational Univariate Representation*) and finally getting numerical approximations of the solutions, when needed.

This efficiency allows to use zero-dimensional solving as a sub-task in other algorithms. This has the consequence that further improvements are yet needed and that complexity issues take a new importance, especially for applications to cryptology.

3.3. Polynomial Systems of Positive Dimension

When a system is not zero-dimensional, the set of solutions is infinite, and it is no more possible to enumerate them. Therefore, the solving process reduces to decompose the set of the solutions into subsets which have a well-defined geometry. One may do such a decomposition from an algebraic point of view or from a geometrical one, the latter meaning not taking the multiplicities into account. Although there exist algorithms for both approaches, the algebraic point of view is presently out of the possibilities of practical computations, and we restrict ourselves to geometrical decompositions.

When one studies the solutions in an algebraically closed field, the decompositions which are useful are the equi-dimensional decomposition (which consists in considering separately the isolated solutions, the curves, the surfaces, ...) and the prime decomposition in irreducible components. In practice, the team works on algorithms for decomposing in *regular separable triangular sets*, which corresponds to a decomposition in equi-dimensional but not necessarily irreducible components. These irreducible components may be obtained at the end by using polynomial factorization.

However, in many situations one is looking only for real solutions satisfying some inequalities[2]. In this case, there are various kinds of decompositions besides the above ones: connected components, cellular or simplicial decompositions, ...

There are general algorithms for such tasks, which rely on Tarski's quantifier elimination. Unfortunately, these problems have a very high complexity, usually doubly exponential in the number of variables or the number of blocks of quantifiers, and these general algorithms are intractable. It follows that the output of a solver should be restricted to a partial description of the topology or of the geometry of the set of solutions, and our research consists in looking for more specific problems, which are interesting for the applications, and which may be solved with a reasonable complexity.

SPACES got results recently or has work in progress about such specific problems, for example: testing if a semi-algebraic set is empty; counting the number of its connected components or bounding their number by providing (at least) a point in each component; counting the number of solutions of a system depending on parameters, as a function of the parameters; globally optimizing an algebraic cost function with algebraic constraints; solving over-determined zero-dimensional systems involving approximate coefficients coming from experimental measurements (this may be viewed as an optimization problem); drawing in a robust way plane curves defined by an implicit equation; proving automatically theorems of elementary geometry.

3.4. Arithmetics

For solving polynomial systems, it is not enough to have algorithms of good arithmetical complexity[3], if the operations on the coefficients are not extremely fast. In fact, we very frequently encounter coefficients with several thousands of digits. It is therefore of prime necessity to optimize any factor of the real cost of the algorithms (bit complexity).

There are mainly two kinds of basic algorithms which may have a dramatic effect on the efficiency of a solver, the arithmetic on the coefficients of the equations to solve and the data management (garbage collector, protocol for transferring data, ...). Although we obtained in the past some important results on data management, especially the patented protocol UDX for transferring binary data between processes and

[2] In the zero-dimensional case, inequations and inequalities are usually taken into account only at the end of the computation, to eliminate irrelevant solutions.

[3] The arithmetical complexity is the number of operations on the coefficients.

computers, we will not give details here on this aspect and we will emphasize on our work on arithmetics. In fact, while our initial purpose on this subject was mainly the needs of polynomial system solvers, an increasing part of the team members is working on arithmetics for themselves and not only for the needs of the solvers. The aspects of arithmetics on which SPACES is working are the following:

Middle product. The team has introduced a new arithmetic operation, the middle product, which consists in computing only the middle bits of a product. This operation allows to improve the efficiency of the division and of the square root computations. It has also been used by another team, for improving the efficiency of polynomial evaluation and interpolation.

Floating-point arithmetic. The team is mainly interested in controlling the rounding errors in floating-point computations. Inside this subject we are studying the worst cases for arithmetic operations and multi-precision floating-point arithmetic with correct rounding. The latter is the object of our library MPFR and of its derived library MPFI for interval computations.

Finite field arithmetic. For cryptographic purposes, we are designing a library for finite field arithmetics which should work efficiently whatever the size of the field (small, medium or large).

Infinitesimal numbers. The arithmetic of infinitesimal (or non standard) numbers may appear as rather exotic, but it is needed in real geometry computations, where many algorithms use infinitesimal deformations.

Other arithmetics. The following are the various arithmetics the team is interested in but has not recently worked on: the arithmetics of polynomials, the arithmetics of algebraic numbers and the hybrid arithmetics which consist in representing a rational number by a pair of a floating-point number and a modular number.

3.5. Solving Problems — Applications

Applications are fundamental for our research for several reasons.

The first one is that they are the only source of fair tests for the algorithms. In fact, the complexity of the solving process depends very irregularly on the problem itself. Therefore, random tests do not give a right idea of the behavior of a program, and the complexity analysis, when possible, does not provide realistic information.

A second reason is that, as quoted above, we need real world problems to determine which specifications of algorithms are really useful. Conversely, it is frequently by solving specific problems through ad hoc methods that we found new algorithms of general impact.

Finally, obtaining successes with problems which are intractable by the other known approaches is the best proof of the importance of polynomial system solving and of the value of our work.

On the other hand, there is a specific difficulty. The problems which may be solved with our methods may be formulated in many different ways, and their usual formulation is rarely well suited for polynomial system solving. Frequently, it is not even clear that the problem is purely algebraic, because researchers and engineers are used to formulate them in a differential way or to linearize them before asking questions on it. Therefore, our softwares may not be used as black boxes, and we have to understand the origin of the problem in order to translate it in a form which is well suited for our solvers.

It follows that many of our papers, published or in preparation, are classified in scientific domains which are different from ours, like cryptology, error correcting codes, robotics, signal processing, statistics or biophysics.

4. Application Domains

4.1. Introduction

In this section, we describe the application domains in which the team is doing a significant work.

4.2. Robotics

4.2.1. Parallel Manipulators

The manipulators which we study are general parallel robots: the hexapodes are complex mechanisms made up of six (often identical) kinematic chains, of a base (fixed rigid body including six joints or articulations) and of a platform (mobile rigid body containing six other joints).

The design and the study of parallel robots require the setting up and the resolution of direct geometrical models (calculation of the absolute coordinates of the joints of the platform knowing the position and the geometry of the base, the geometry of the platform as well as the distances between the joints of the kinematic chains at the base and the platform) and inverse geometrical models (distances between the joints of the kinematic chains at the base and the platform knowing the absolute positions of the base and the platform).

Since the inverse geometrical models can be easily solved, we focus on the resolution of the direct geometrical models.

The study of the direct geometrical model is a recurrent activity for several members of the project. One can say that the progress carried out in this field illustrates perfectly the evolution of the methods of resolution of algebraic systems. The interest carried on this subject is old. The first work the members of the project took part in primarily concerned the study of the number of (complex) solutions of the problem. The results were often illustrated by calculations of Gröbner bases done with the GB software. The next efforts were related to the real roots and the effective calculation of the solutions. The studies then continued following the various algorithmic progresses, until the developed tools made possible to solve non-academic problems. In 1999, the various efforts were concretized by an industrial contract with the SME CMW (*Constructions Mécaniques des Vosges-Marioni*) for making a robot dedicated to machine tools.

New problems have appeared and are related to two of our research directions:

- the calculation in real time of the direct geometrical model. This could be done in particular by generating numerical calculation programs starting from a sufficiently generic exact calculation (Gröbner bases).
- the study of systems of positive dimension, for example, in order to be able to process data depending on time (dimension 1) or to allow to slacken some parameters of the problem for studying them.

4.2.2. Serial Manipulators

Industrial robotic manipulators with 3 degrees of freedom are currently designed with very simple geometric rules on the designed parameters, the ratios between them are always of the same kind. In order to enlarge the possibilities of such manipulators, it may be interesting to relax the constraints on the parameters.

However, the diversity of the tasks to be done carries out to the study of other types of robots whose parameters of design differ from what is usual and which may have new properties, like stability or existence of new kinds of trajectories.

An important difficulty slows down the industrial use of such new robots. Recent studies ([58], [60], [59] and [50]) showed that they may present a behavior which is qualitatively different from that of the robots currently used in industry and allows new changes of posture. These robots, called cuspidal, cannot be controlled like the others. The majority of the robots are in fact cuspidal: the industrial robots currently on the market form a very restricted subclass of all the possible robots.

A systematic characterization of all the cuspidal robots would be of a great interest for the designer and the user. Such a project forms part of a current tendency in robotics which consists in designing a robot in order that its performances are optimal for a given application while preserving the possibility of using it for another task, that is to say to specialize it to the maximum for an application in order to reduce its cost and to increase its operational safety.

The study of the behavior at a change of posture is identical, from a mathematical point of view, to the study of the existence of a triple root of some polynomials of degree 4 depending on the parameters.

4.3. Geometrical Reasoning

Geometric reasoning is an active area of research in which algebraic methods (such as Gröbner bases, triangular sets, and quantifier elimination) for solving polynomial systems have successful applications. These applications cross over several important areas of modern engineering geometry such as computer aided geometric design and modeling (geometric constraint solving, implicitization of rational curves and surfaces, surface blending and offsetting, etc.), computer vision (adjustment of models, derivation of geometric properties, etc.), and robotics (see Section 4.2). One fundamental problem of geometric reasoning is to study and establish relations among geometric objects, for example, to prove known geometric theorems and to derive unknown geometric relations. This problem can be attacked effectively by translating geometric relations into algebraic expressions and then using algebraic methods.

In the framework of the SPACES project, we have developed and applied algebraic methods and software tools to deal with several geometric reasoning problems including proving and discovering theorems in elementary and differential geometry, solving geometric problems for which real solutions of the corresponding algebraic systems with parameters need be studied, generating geometric diagrams automatically, implicitizing rational curves and surfaces, and computing the offsets of algebraic curves and surfaces.

4.4. Cryptology

The idea of using multivariate (quadratic) equations as a basis for building public key cryptosystems appeared with the Matsumoto-Imai cryptosystem. This system was first broken by Patarin and, shortly after, Patarin proposed to repair it and thus devised the hidden field equation (HFE) cryptosystem.

The basic idea of HFE is simple: build the secret key as a univariate polynomial $S(x)$ over some (big) finite field (often GF(2^n)). Clearly, such a polynomial can be easily evaluated; moreover, under reasonable hypotheses, it can also be "inverted" quite efficiently. By inverting, we mean finding any solution to the equation $S(x) = y$, when such a solution exists. The secret transformations (decryption and/or signature) are based on this efficient inversion. Of course, in order to build a cryptosystem, the polynomial S must be presented as a public transformation which hides the original structure and prevents inversion. This is done by viewing the finite field GF(2^n) as a vector space over GF(2) and by choosing two linear transformations of this vector space L_1 and L_2. Then the public transformation is the composition of L_1, S and L_2. Moreover, if all the terms in the polynomial $S(x)$ have Hamming weight 2, then it is obvious that all the (multivariate) polynomials of the public key are of degree two.

By using fast algorithms for computing Gröbner bases, it was possible to break the first HFE challenge (real cryptographic size 80 bits and a symbolic prize of 500 US$) in only two days of CPU time. More precisely we have used the $F_5/2$ version of the fast F_5 algorithm for computing Gröbner bases (implemented in C). The algorithms available up to now (Buchberger) were extremely slow and could not have been used to break the code (they should have needed at least a few centuries of computation). The new algorithm is thousands of times faster than previous algorithms. Several matrices have to be reduced (Echelon Form) during the computation: the biggest one has no less than 1.6 million columns, and requires 8 gigabytes of memory. Implementing the algorithm thus required significant programming work and especially efficient memory management.

Since it is easy to transform many cryptographic problems into polynomial equations, SPACES is in a position to apply this general method to other cryptosystems. Thus we have a new general cryptanalysis approach, called algebraic cryptanalysis. G. Ars and J.-C. Faugère are currently testing the robustness of cryptosystems based on filtered LFSRs (Linear Feedback Shift Registers), in collaboration with the CODES team and the DGA (Celar).

Another relevant tool in the study of cryptographic problems is the LLL algorithm which is able to compute in polynomial time a "good" approximation for the shortest vector problem. Since a Gröbner basis can be seen as the set of smallest polynomials in an ideal with respect to the divisibility of leading terms, it is natural to compare both algorithms: an interesting link between LLL (polynomial version) and Gröbner bases was suggested by a member of SPACES.

A standard algorithm for implementing the arithmetic of Jacobian groups of curves is LLL. By replacing LLL by the FGLM algorithm we establish a new structure theorem for Gröbner bases; consequently, on a generic input we were able to establish explicit and optimized formulas for a basic arithmetic operation in the Jacobian groups of C_{34} curves (in collaboration with the TANC team).

5. Software

5.1. Introduction

An important part of the research done in the SPACES project is published within software. We present here our software following the same order as in Section 2.: zero-dimensional systems, systems of positive dimension, arithmetics, solving problems and applications.

5.2. RS/RealSolving

Keywords: *real root*, *univariate polynomial*, *zero-dimensional system*.

Participant: Fabrice Rouillier.

RS is a software dedicated to the study of real zeroes of algebraic systems. It is written in C (100,000 lines approximately) and is the successor to RealSolving developed at the time of the European projects PoSSo and FRISCO. RS mainly contains functions for counting and isolating real zeroes of algebraic systems with a finite number of complex roots. The user interfaces of RS are entirely compatible with those of GB/FGB (ASCII, MuPAD, Maple). RS has been used in the project and by various teams for several months.

5.3. Gb/FGb

Keywords: *Gröbner basis*, *complex root*.

Participant: Jean-Charles Faugère.

GB is a C++ program for efficiently computing Gröbner bases. From a theoretical and practical point of view GB is now outperformed by the new FGB program.

GB is a stand-alone program but it can be used from general computer algebra systems (for instance Maple).

GB is completely *free* for academic people and non-commercial use.

5.4. Implicit Curves Drawing

Keywords: *drawing*, *implicit curve*.

Participants: Jean-Charles Faugère, Fabrice Rouillier [contact].

As soon as they come from real applications, the polynomials resulting from elimination processes (Gröbner bases, triangular sets) are very often too large to be studied by general computer algebra systems.

In the case of polynomials in two variables, a certified layout is enough in much cases to solve the studied problem (it is the case in particular for certain applications in celestial mechanics). This type of layout is now possible thanks to the various tools developed in the project.

Two components are currently under development: the calculation routine (taking as input the polynomial function and returning a set of points) is stable (about 2,000 lines in the C language, using the internal libraries of RS) and can be used as a black box in stand-alone mode or through Maple; the layout routine is under study.

5.5. RAGLib

Keywords: *polynomial system*, *real solution*.

Participants: Colas Le Guernic [MMFAI internship], Mohab Safey El Din [contact].

RAGLib (**R**eal **A**lgebraic **G**eometry **Lib**rary) is a Maple library of symbolic algorithms devoted to some problems of effective real algebraic geometry, and more particularly, to the study of real solutions of polynomial systems of equations and inequalities. It contains algorithms performing:

- the *equi-dimensional decomposition* of an ideal generated by a polynomial family;

- the *emptiness test* of a real algebraic variety defined by a polynomial system of equations;
- the *computation of at least one point in each connected component* of a real algebraic variety defined by a polynomial system of equations;
- the *emptiness test* of a semi-algebraic set defined by a polynomial system of equations and non strict inequalities;
- the *computation of at least one point in each connected component* of a semi-algebraic set defined by a polynomial system of equations and non strict inequalities.

5.6. Triangular Decomposition

Keywords: *polynomial gcd*, *polynomial system*, *triangular set*, *unmixed decomposition*.

Participant: Philippe Aubry.

Triangular Decomposition is a library devoted to the decomposition of systems of polynomial equations and inequations and provides some tools for working with triangular sets. It decomposes the radical of the ideal generated by a family of polynomials into regular triangular sets that represent radical equidimensional ideals. It also performs the computation of polynomial gcd over an extension field or a product of such fields given by a triangular set.

A first version of this library was implemented in the Axiom computer algebra system. It is now developed in Magma.

5.7. MPFR

Keywords: *IEEE 754*, *arbitrary precision*, *correct rounding*, *floating-point number*.

Participants: Laurent Fousse, Guillaume Hanrot, Vincent Lefèvre, Patrick Pélissier, Paul Zimmermann [contact].

MPFR is one of the main software developed by the SPACES team. MPFR is a library for computing with arbitrary precision floating-point numbers, together with well-defined semantics. In particular, all arithmetic operations are performed according to a rounding mode provided by the user, and all results are guaranteed correct to the last bit, according to the given rounding mode.

The main objective of year 2004 was to consolidate the current version of MPFR, and to release a first version containing the improvements made by P. Pélissier for small precision (see below). As a result, MPFR 2.0.3, was released in February, and MPFR 2.1.0 in November.

In September 2003, P. Pélissier joined the MPFR team, as a Junior technical staff, to help improve the efficiency of MPFR for small precision (up to 200 bits, in particular in double, double extended and quadruple precision). P. Pélissier designed a program called MPFR-BENCH to precisely measure the number of cycles spent by each of the basic MPFR routines. Then he proposed a new architecture to reduce the overhead of dealing with special values (Not-a-Number, infinities and zeroes). In addition, special-purpose code has been written in the common case where both the arguments and the destination have the same precision, for addition and subtraction. These optimizations are included in MPFR 2.1.0, released in November 2004. The following table compares the number of cycles of the five basic operations between MPFR 2.0.1 and MPFR 2.1.0, on a Pentium 4 (Northwood model) and an Athlon, for a precision of 53 bits, corresponding to the double precision from IEEE 754.[4]

version	machine	add	sub	mul	div	sqrt
2.0.1	Pentium 4	298	398	331	1024	1211
2.1.0	Pentium 4	211	213	268	549	1084
2.0.1	Athlon	222	323	270	886	975

[4] MPFR was compiled with GMP 4.1.4, with the same compilation flags than those determined by GMP's configure.

2.1.0	Athlon	132	151	183	477	919

Following the theoretical results from [10], a new function `mpfr_sum` was added to MPFR. This function lets the user compute the sum of an arbitrary number of floating-point numbers with correct rounding, and is considered especially useful when performing scalar products (a very common usage of floating-point numbers).

5.8. MPFI

Participants: Nathalie Revol [ARENAIRE project], Fabrice Rouillier [contact].

MPFI is a library for multiprecision interval arithmetic, written in C (approximately 1,000 lines), based on MPFR. Initially, MPFI was developed for the needs of a new hybrid algorithm for the isolation of real roots of polynomials with rational coefficients. MPFI contains the same number of operations and functions as MPFR, the code is available and documented.

5.9. MPAI

Participant: Philippe Trébuchet.

MPAI is a library for computing with algebraic infinitesimals. The infinitesimals are represented as truncated series. The library provides all the arithmetic functions needed to perform computations with infinitesimals. The interface is both GMP and RS compliant. It is implemented in the C language and represents approximatively 1,000 lines of code. The algorithms proposed in MPAI include Karatsuba's product and the short product, ...The code is available.

5.10. Exhaustive Tests of the Mathematical Functions

Participant: Vincent Lefèvre.

The tests of the mathematical functions ($\exp$, $\log$, $\sin$, $\cos$, etc.) have been partially rewritten. In particular, all the low-level routines were rewritten in ISO C in order to be portable, using the MPN layer of GMP for speed reasons. The code was also improved from the algorithmic point of view, in particular to reduce the number of iterations from several thousands to two or three in some intervals; one third of the intervals that were too long to test previously could now be tested very quickly.

The results are used:

- by us, to detect bugs in MPFR and in the GNU C library (glibc);
- by the ARENAIRE team, for their implementation of the mathematical functions with correct rounding.

5.11. Approximations of a Function by Polynomials

Participants: Vincent Lefèvre, Marc Helbling.

Marc Helbling worked on the hierarchical approximations of a function (approximated and represented by a high-degree polynomial) by polynomials of low degrees, as first described in [13]. One of the goals is to be able to quickly evaluate a function at regularly spaced points; this is needed for the exhaustive tests of the mathematical functions. Both suggested methods have been studied and implemented using MPFR.

5.12. UDX

Keywords: *binary data*, *protocol*.

Participant: Fabrice Rouillier.

XDR eXternal Data Representation

UDX Universal Data Representation

UDX is a software for binary data exchange. It was initially developed to show the power of a new protocol, object of a patent by INRIA and UPMC (deposit number 99 08172, 1999). The resulting code, written in ANSI C (9,500 lines), is very portable and very efficient, even when the patented protocol is not used. UDX is composed of five independent modules:

- base: optimized system of buffers and synchronization of the input and output channels;
- supports: read/write operations on various supports (sockets, files, shared memory, etc.);
- protocols: various exchange protocols (patented protocol, XDR, etc.);
- exchange of composite types: floating-point numbers (simple and double precision), multiprecision integers, rational numbers;
- interfaces: user interfaces implementing high level callings to the four other modules.

UDX is used in some interfaces developed in the project (Gb, RS, MuPAD) but also in software from other projects (SYNAPS and ROXANE, in collaboration with the INRIA project GALAAD [48], MuPAD/Scilab interface distributed with MuPAD 2.5.1).

5.13. Interfaces

Keywords: *interfaces*.

Participants: Jean-Charles Faugère [contact], Fabrice Rouillier.

In order to ease the use of the various softwares developed in the project, some conventions for the exchange of ASCII and binary files were developed and allow a flexible use of the servers Gb, FGb or RS.

To make transparent the use of our servers from general computer algebra systems such as Maple or MuPAD, we consolidated and homogenized the prototypes of existing interfaces and we currently propose a common distribution for Gb, FGb and RS including the servers as well as the interfaces for Maple and MuPAD. The instructions are illustrated by concrete examples and a simple installation process.

5.14. TSPR

Keywords: *parallel manipulator*, *trajectory*.

Participant: Fabrice Rouillier [contact].

The purpose of the TSPR project (Trajectory Simulator for Parallel Robots) is to homogenize and export the tools developed within the framework of our applications in parallel robotics. The software components of TSPR (about 1500 lines) are primarily written in C following the standards of RS and FGb and algorithms implemented in Maple using the prototypes of interfaces for FGb and RS. The encapsulation of all these components in a single distribution is available but not downloadable.

Some prototypes of components for real-time resolution of certain types of systems now use the ALIAS library developed in the INRIA project COPRIN.

5.15. Function Field Sieve

Keywords: *discrete logarithm*, *finite fields*, *function field sieve*.

Participants: Dahab Hakim, Emmanuel Thomé [contact].

Dahab Hakim (DEA Limoges) studied and implemented the function field sieve algorithm for computing discrete logarithms in fields of characteristic three. The resulting code is mixed C and Magma.

6. New Results

6.1. Interval Arithmetic

Participants: Nathalie Revol [ARENAIRE project], Fabrice Rouillier [contact].

In [25], we justify why an arbitrary precision interval arithmetic is needed. To provide accurate results, interval computations require small input intervals; this explains why bisection is so often employed in interval algorithms. The MPFI library has been built in order to fulfill this need. Indeed, no existing library met the required specifications. The main features of this library are briefly given and a comparison with a fixed-precision interval arithmetic, on a specific problem, is presented. It shows that the overhead due to the multiple precision is completely acceptable. Eventually, some applications based on MPFI are given: robotics, isolation of polynomial real roots (by an algorithm combining symbolic and numerical computations) and approximation of real roots with arbitrary accuracy.

6.2. Arithmetic of $C_{3,4}$ Curves

Participant: Jean-Charles Faugère.

In [30], we provide explicit formulas for realising the group law in Jacobians of superelliptic curves of genus 3 and $C_{3,4}$ curves. It is shown that two distinct elements in the Jacobian of a $C_{3,4}$ curve can be added with 150 multiplications and 2 inversions in the field of definition of the curve, while an element can be doubled with 174 multiplications and 2 inversions. In superelliptic curves, 10 multiplications are saved.

6.3. Univariate Polynomials

Participant: Fabrice Rouillier.

In [43] and [24], we explain how Bernstein's basis, widely used in Computer Aided Geometric Design, provides an efficient method for real root isolation, using De Casteljau's algorithm. We explain also the link between this approach and more classical methods for real root isolation [56]. Most of the content of the paper can be found in [49][45]. However, we present a new improved method for isolating real roots in Bernstein's basis inspired by [52].

6.4. Generalized Normal Forms

Participant: Philippe Trébuchet.

Let $K[x_1, ..., x_n]$ be the ring of n-variate polynomials over the field K, and I an ideal of $K[x_1, ..., x_n]$. Finding an effective representation of the quotient algebra $K[x_1, ..., x_n]/I$ is a key point for studying the variety defined by I. Efficient ways to reach this goal exist, and the most used is to first compute a Gröbner basis, and next to derive an exact parameterization of the solution set. Though the work of Faugère made the Gröbner step rather efficient, the computation of Gröbner bases still suffers from very unstable behavior: the time and space needed for computing a Gröbner basis can vary greatly from a system to one another. Such changes obviously must appear when the structure of the variety defined by the system changes. However the behavior of the computation of Gröbner bases changes in a way uncorrelated to geometric changes of the variety. This is rather problematic as it implies that the actual cost of the solving process of a given system via Gröbner bases may be dramatically greater than what ought to be paid.

Our common work with Mourrain showed that in fact the methods of Macaulay and Buchberger share the same directing idea. Remark that Macaulay's method is known for its good stability properties, hence the idea to mix Macaulay's and Buchberger's approaches seems natural. The algorithm we provided so far was only shown to work for zero dimensional ideals. In the article [40] we provide a stopping criterion for this algorithm when the input system is positive dimensional. First we show that detecting that the ideal generated by the homogeneous parts of maximal degree of the polynomial sets constructed by the algorithm is k-regular (i.e. the Castelnuovo-Mumford regularity of this ideal is less than k) is enough to ensure that these sets allow to solve the *ideal membership* problem. Next, we show that detecting that this ideal is k-regular is simple using Quillen's theorem.

6.5. Complexity of Gröbner Bases

Participants: Magali Bardet, Jean-Charles Faugère.

In [29], we extend the notion of regular sequence (Macaulay) to overdetermined systems of algebraic equations. We study generic properties of Gröbner bases and analyze precisely the behavior of the F_5 algorithm. We extend the previous known results by defining the semi regular sequences and the degree of regularity for which we give a precise asymptotic analysis. For example, as soon as the number m of polynomials is strictly greater than the number n of variables, we gain a factor of 2 on the Macaulay bound and a factor 11.65 when $m = 2n$.

6.6. Gröbner Bases over Finite Fields

Participants: Gwenolé Ars, Jean-Charles Faugère.

In [28], we compare the XL algorithm with the Gröbner basis algorithm. We explain the link between the XL result and the Gröbner basis result with the well-known notion of D-Gröbner basis. Then we compare these algorithms in two cases: in the fields F_2 and F_q with $q < n$. For the field F_2, we have proved that if XL needs to compute polynomials with degree D to terminate, the whole Gröbner basis is computed without exceeding that degree.

We have studied the XL algorithm and the F_5 algorithm on semi-regular sequences. We show that the size of matrices constructed by XL is huge compared to the ones of F_5.

So the complexity of XL is worse than that of the F_5 algorithm on these systems. For the field F_q, we introduce an emulated algorithm using Gröbner basis computation to have a comparison between XL and Gröbner basis. We have proved that this algorithm will always reach a lower degree for intermediate polynomials than the XL algorithm. A study on semi-regular sequences shows that F_5 always has a better behavior than the XL algorithm especially when m is near n.

6.7. Zero-Dimensional Systems

Participant: Fabrice Rouillier.

In [37], we propose some new results about the computation and use of the Rational Univariate Representation. The first result is an algorithm that computes a RUR-candidate (no guarantee for the choice of a separating element), performing $O(\sharp\mathcal{T}D^{3/2} + nD^2)$ operations in $\mathbb{Q}$, taking as input $\mathcal{T}$, a multiplication table of $\mathbb{Q}[X_1, ..., X_n]/I$ which is slightly better than [46]. The proposed algorithm is mainly the same as in [46] (based on the baby-step/giant-step algorithm [55]) but computes differently the so called "transposed product" (main operation). In addition, the output preserves the multiplicities. Basically, [46] generalizes the formulas from [51]: the coefficients of the RUR-candidate can directly be expressed with respect to $l(X_j t^i)$ where l is a "sufficiently generic" linear form. Taking for l the trace map (trace of the multiplication in $\mathbb{Q}[X_1, ..., X_n]/I$), one recovers the formulas from [51]. The second result shows that one can check that the obtained RUR-candidate is a RUR performing $O(n\sharp\mathcal{T}D^{3/2})$ arithmetic operations, so that it produces the same output as [51] with the same computing time as the probabilistic computation of the RUR-candidate from [46]. We also propose an algorithm that computes all the sign conditions realized by a set of polynomials at the zeroes of a zero-dimensional system. Its computing time complexity is D times less than the one of Algorithm 11.18 p. 375 in [45] for the same input ($\mathcal{T}$). Given any polynomial f, the formulas from [51] or [46] can be used to compute a rational function $g_{t,f}/g_t$ which takes the same values as f at the zeroes of I. According to the results above, given a set of polynomials $\mathcal{F} = \{f_1, ..., f_s\}$, one can compute a RUR and the rational functions g_{t,f_i} in $O(t_1 + sD^2)$ if t_1 is the computing time of the RUR. Thus, computing the sign conditions realized by the f_i leads to computing the sign conditions realized by the g_{t,f_i} at the roots of f_t. We can then substitute the "Sturm query" algorithm based on Hermite's quadratic form in [45] (computing time in $O(D^3)$ by its equivalent for univariate polynomials (see also for example [45]) - computing time in $O(D^2)$ for the naive version).

6.8. Parametric Polynomial Systems

Participants: Daniel Lazard, Fabrice Rouillier.

In [41] and [34], we present a new algorithm for solving basic parametric constructible or semi-algebraic systems like $\mathcal{C} = \{x \in \mathbb{C}^n, p_1(x) = 0, ..., p_s(x) = 0, f_1(x) \neq 0, ..., f_l(x) \neq 0\}$ or $\mathcal{S} = \{x \in \mathbb{C}^n, p_1(x) = 0, ..., p_s(x) = 0, f_1(x) > 0, ..., f_l(x) > 0\}$, where $p_i, f_i \in \mathbb{Q}[U, X]$, $U = [U_1, ..., U_d]$ is the set of parameters and $X = [X_{d+1}, ..., X_n]$ the set of unknowns.

If Π_U denotes the canonical projection on the parameter's space, solving $\mathcal{C}$ leads to compute sub-manifolds $\mathcal{U} \subset \overline{\Pi_U(\mathcal{C})}$ such that $(\Pi_U^{-1}(\mathcal{U}) \cap \mathcal{C}, \Pi_U)$ is an analytic covering of $\mathcal{U}$ (we say that $\mathcal{U}$ has the $(\Pi_U, \mathcal{C})$-*covering property*). This guarantees that the cardinal of $\Pi_U^{-1}(\mathcal{U}) \cap \mathcal{C}$ is locally constant on $\mathcal{U}$ and that $\Pi_U^{-1}(\mathcal{U}) \cap \mathcal{C}$ is a finite collection of sheets which are all locally homeomorphic to $\mathcal{U}$. In the case where $\Pi_U(\mathcal{C})$ is dense in $\mathbb{C}^d$, known algorithms for solving $\mathcal{C}$ or $\mathcal{S}$ ([47], [54], [57]) compute implicitly or explicitly a Zariski closed subset W such that any sub-manifold of $\overline{\Pi_U(\mathcal{C})} \setminus W$ has the $(\Pi_U, \mathcal{C})$-covering property.

We introduce the *discriminant varieties of* $\mathcal{C}$ *with respect to* Π_U which are algebraic sets with the above property. We then show that the set of points of $\overline{\Pi_U(\mathcal{C})}$ which do not have any neighborhood with the $(\Pi_U, \mathcal{C})$-covering property is a Zariski closed set and thus defines the *minimal discriminant variety of* $\mathcal{C}$ *with respect to* Π_U, and we propose an algorithm to compute it efficiently. Thus, solving $\mathcal{C}$ (resp. $\mathcal{S}$) remains to describing $\overline{\Pi_U(\mathcal{C})} \setminus W_D$ (resp. $(\overline{\Pi_U(\mathcal{C})} \setminus W_D) \cap \mathbb{R}^d$) which can be done using critical point methods such as in [45] or partial CAD based strategies [47].

6.9. Parametric Varieties and Rational Maps

Participant: Daniel Lazard.

Our work on parametric varieties and rational maps began with an application of our methods to statistics which leads us to a complete study of the topology of the image of a rational mapping [23]. This application became our first significative example of a parametric polynomial system with a non dense projection on the space of the parameters, and was therefore a fundamental tool for extending correctly our theory of discriminant varieties to this case.

Rational maps appear in many application domains, whenever some measurable quantities are rational functions of some hidden state variables. They appear also in the definition of the parametric varieties, which are widely used in geometrical design. On the other hand, their linearity with respect to their image leads to shortcuts in many computations, especially the one of the discriminant variety.

Therefore their study deserves a special effort which started with [23], has been continued with [33],[41] and will be submitted to the book following the AGGM conference.

6.10. Positive Dimensional Systems of Equations

Participant: Mohab Safey el Din.

The results of [26] have been published. In this article, we provide an algorithm computing at least one point in each connected component of a real algebraic set defined by a polynomial system of equations. The used strategy is based on the compution of critical loci of some projections and their respective sets of non-properness. The paper ends with intensive experimental results. These results have been improved in [53].

In [27], we have provided some sharp bounds on the number of critical points of a polynomial mapping restricted to a smooth algebraic variety. These bounds have been obtained by exploiting the bi-homogeneous structure of the Lagrange system and proving a Bezout bound to such systems on the sum of the degree of the isolated primary components of the ideal generated by the studied polynomial system. Then, we have used these results to improve the Thom-Milnor bound which majorates the maximal number of connected components of a real algebraic set. This is done by generalizing the algorithm of [53] to non equidimensional situations for which some optimal bounds on the degree of the output and the number of arithmetic operations. These results have been submitted to Journal of Complexity.

6.11. Generalized Critical Values and Semi-Algebraic Systems

Participants: Mohab Safey el Din [contact], Fabrice Rouillier.

In [35], we provide an algorithm computing at least one point in each connected component of a semi-algebraic set defined by a system of equations and non-strict inequalities. We prove improved complexity bounds to tackle this problem, provide an implementation, and study an application to pattern-matching problems.

In [38], we focus on the case of semi-algebraic sets defined by a single polynomial inequality or a single polynomial inequation. In this case, a classical strategy consists in introducing an infinitesimal and performing computations on a Puiseux series field. We substitute this strategy by finding an *a priori* good specialization value for this infinitesimal. This is done by computing the set of generalized critical values of a polynomial mapping. The complexity of the obtained algorithm is better than the previous ones and an implementation is provided. The extended abstract presenting these results has been accepted for an oral communication at ICPSS'04 (International Conference on Polynomial System Solving) which has been held at Pierre et Marie Curie University, Paris. The full paper is in preparation and will be submitted to the Special Issue of Journal of Symbolic Computation in honour of Daniel Lazard.

6.12. Integer and Polynomial Arithmetic

Participants: Richard Brent, Guillaume Hanrot, Paul Zimmermann.

Several articles describing fundamental work done in the last years finally appeared in 2004. The article describing the "middle product" appeared in *AAECC* [21]. The article describing the optimal strategy for Mulders' short product when used with Karatsuba multiplication appeared in the *Journal of Symbolic Computation* [22]. The 2-adic gcd algorithm was presented at the ANTS-VI conference [39]. The article describing the search for primitive trinomials of degree 6972593 over GF(2) appeared in *Mathematics of Computation* [18].

6.13. Floating-Point Arithmetic

Participants: Laurent Fousse, Guillaume Hanrot, Vincent Lefèvre, Damien Stehlé, Paul Zimmermann.

Following the effort towards a standardization of mathematical functions in the IEEE 754 standard [19], we propose an extension of Gal's "accurate table" method [44]. This paper contains two main ideas. For one function, we use "worst-case" inputs as distinguished points, which provide the arguably best set of values for Gal's method. For two functions, for example $\sin$ and $\cos$, we present a new algorithm, using a variant of that from [17], which finds all simultaneous n-bit worst cases in $O(2^{n/2})$, instead of $O(2^n)$ for the naive method.

Laurent Fousse and Susanne Schmitt (MPI für Informatik, Saarbrücken) analyzed two polynomial evaluation schemes, providing proven error bounds on the result of the evaluation together with fast implementation of the algorithms [32]. This work confirms the common knowledge that Horner's scheme is considered numerically more stable but gives a rigourous error analysis and benchmarks. The idea to truncate partial results was used to speed up the computation where possible; this might give an improvement in the calculation of elementary series.

The generic multiple-precision floating-point addition of two positive floating-point numbers in base 2 with correct rounding, as specified in MPFR, is studied and a bit-based version of the algorithm is described in [36]. This study shows that the MPFR implementation is asymptotically optimal. Moreover a future mechanically-checked proof could be based on this work.

Vincent Lefèvre started a work on the integration of numerical computations into the OMSCS framework, in Jacques Calmet's team at Karlsruhe University, Germany [31]. This work aims at showing what problems can occur when dealing with numerical computations (with a floating-point or any other arithmetic) and what can be done to solve them or at least to provide a clear meaning of a numerical result; it has been partially supported by the Calculemus Research Training Network HPRN_CT-2000-00102.

6.14. Cryptology

Participant: Emmanuel Thomé.

Emmanuel Thomé and Pierrick Gaudry (LIX, École polytechnique) designed and analyzed a new algorithm suited for computing discrete logarithms in hyperelliptic curves in genus 3. This algorithm improves on the previously best known algorithms, since the complexity drops from $O(q^{2-\frac{2}{g+1/2}})$ to $O(q^{2-\frac{2}{g}})$, for computing discrete logarithms in hyperelliptic curves of genus 3 defined over $\mathrm{GF}(q)$. The algorithm has been implemented, and we have been able to compute discrete logarithms in a group of cardinality $\approx 2^{81}$ (genus 3 over $\mathrm{GF}(2^{27})$). The main originality of this work lies in the analysis of the algorithm. Indeed, the technique used is known as the double large prime variation, and has been commonplace for a long time in neighboring contexts (integer factorization, index calculus algorithms in general...). Only heuristic or experimental arguments have been raised so far to justify the use of the double large prime variation, but our work gives a proof, in our particular contexts, that an improvement in complexity is obtained. This work has been published as a preprint on the IACR web server, and an extended version written in collaboration with Nicolas Thériault (University of Waterloo, Canada) will be submitted by the end of 2004.

7. Contracts and Grants with Industry

7.1. MuPAD-Scilab Interface

Participants: Fabrice Rouillier, Paul Zimmermann.

In November 2001, a cooperation agreement around the UDXF program for binary data exchange was signed between the SciFace company (which distributes the computer algebra system MuPAD) and INRIA Lorraine (representing the University Pierre and Marie Curie). UDXF takes as parameter a communication protocol, for example that of patent UDX, but it can also be another protocol. Within the framework of this agreement, SciFace has the right to use UDXF for the MuPAD-Scilab interface. In exchange, SciFace takes part in the development and the improvement of UDX. Version 2.5.x of MuPAD, distributed since September 2002, integrates this interface.

8. Other Grants and Activities

8.1. National Initiatives

8.1.1. Ministry Grant (ACI) "Cryptologie"

Participants: Guillaume Hanrot [contact], Jean-Charles Faugère, Magali Bardet, Gwenolé Ars, Bill Allombert, Renaud Lifchitz.

This project started in 2002 and ended in 2004. The main goal of this project is to study the interactions between cryptology and computer algebra and more specifically the impact of fast polynomial system algorithms on public key cryptosystems based on multivariate (quadratic) equations (such as HFE). For instance a member of this project was able to "break" the first HFE challenge by computing a huge Gröbner basis.

8.2. European Initiatives

8.2.1. Real Algebraic and Analytic Geometry Network

Participants: Fabrice Rouillier [contact], Mohab Safey El Din.

The SPACES project takes part in the European project RAAG (Real Algebraic and Analytic Geometry), F. Rouillier being the deputy chief for the applications and connection with industry item.

RAAG is a Research Training Network of the “Human Potential” program of the European Commission, sponsored for 48 months starting from March 2002. The network is managed by the University of Passau (Germany), the coordination of the French side being ensured by the team of real geometry and computer algebra of the University of Rennes I.

The main goal of this project is to increase the links between the various fields of research listed in the topics of the project (real algebraic geometry, analytical geometry, complexity, formal calculation, applications, etc.) by means of conferences, schools and exchanges of young researchers. It brings together a great number of European teams and in particular the majority of the French teams working in the scientific fields falling under the topics of the project.

9. Dissemination

9.1. Scientific Animation

9.1.1. ICPSS Conference

The members of the project organized jointly with the CALFOR team the first International Conference on Polynomial System Solving. This first edition was in honor of Daniel Lazard. About 100 participants from various countries were present and prestigious researchers gave invited lectures (B. Buchberger, J. Calmet, J. Davenport, H. Hong and M.-F. Roy). A special issue of Journal of Symbolic Computation is planned for 2005.

9.2. Leadership within Scientific Community

P. Zimmermann is member of the program committee of the Arith’17 conference (June 2005, Cape Cod, USA), and was member of the editorial board of a special issue of the *Journal of Logic and Algebraic Programming*, on the practical development of exact real number computation (http://www.informatik.uni-trier.de/~mueller/JLAP/).

9.3. Teaching

G. Hanrot and P. Zimmermann gave lectures (20 hours) about links between arithmetics and cryptology at the *DEA Informatique* of University Nancy 1.

10. Bibliography

Major publications by the team in recent years

[1] P. Aubry. *Ensembles triangulaires de polynômes et résolution de systèmes algébriques*, Ph. D. Thesis, Université Paris 6, 1999.

[2] P. Aubry, D. Lazard, M. M. Maza. *On the theories of triangular sets*, in "Journal of Symbilic Computation", vol. 28, 1999, p. 105-124.

[3] P. Aubry, D. Lazard, M. Moreno Maza. *On the Theories of Triangular Sets*, in "Journal of Symbolic Computation, Special Issue on Polynomial Elimination", vol. 28, 1999, p. 105–124.

[4] P. Aubry, F. Rouillier, M. Safey. *Real Solving for Positive Dimensional Systems*, in "Journal of Symbolic Computation", vol. 34, n^{o} 6, 2002, p. 543–560.

[5] Y. BILU, G. HANROT, P. VOUTIER. *Existence of primitive divisors of Lucas and Lehmer sequences*, in "J. Reine Angew. Math.", vol. 539, 2001, p. 75–122.

[6] Y. BUGEAUD, G. HANROT. *Un nouveau critère pour l'équation de Catalan*, in "Mathematika", vol. 47, 2000, p. 63–73.

[7] J.-C. FAUGÈRE. *A new efficient algorithm for computing Gröbner bases without reduction to zero F_5*, in "International Symposium on Symbolic and Algebraic Computation Symposium - ISSAC 2002, Villeneuve d'Ascq, France", July 2002.

[8] J.-C. FAUGÈRE. *A New Efficient Algorithm for Computing Gröbner bases (F_4)*, in "Journal of Pure and Applied Algebra", vol. 139, n^o 1-3, June 1999, p. 61-88.

[9] J.-C. FAUGÈRE, F. M. DE SAINT MARTIN, F. ROUILLIER. *Design of regular nonseparable bidimensional wavelets using Groebner basis techniques*, in "IEEE SP Transactions Special Issue on Theory and Applications of Filter Banks and Wavelets", vol. 46, n^o 4, April 1998, p. 845-856.

[10] L. FOUSSE, P. ZIMMERMANN. *Accurate Summation : Towards a Simpler and Formal Proof*, in "5th Conference on Real Numbers and Computers 2003 - RNC5, Lyon, France", September 2003, p. 97–108.

[11] A. J. J.-C. FAUGÈRE. *Algebraic Cryptanalysis of Hidden Field Equation (HFE) Cryptosystems Using Gröbner Bases*, in "CRYPTO 2003", 2003, p. 44-60.

[12] J.-C. FAUGÈRE.. *A New Efficient Algorithm for Computing Gröbner Bases (F4)*, in "Journal of Pure and Applied Algebra", vol. 139, n^o 1–3, 1999, p. 61–88.

[13] V. LEFÈVRE. *Moyens arithmétiques pour un calcul fiable*, Thèse de doctorat, École Normale Supérieure de Lyon, 2000.

[14] F. ROUILLIER. *Solving Zero-Dimensional Systems Through the Rational Univariate Representation*, in "Journal of Applicable Algebra in Engineering, Communication and Computing", vol. 9, n^o 5, 1999, p. 433–461.

[15] F. ROUILLIER, P. ZIMMERMANN. *Efficient Isolation of Polynomial Real Roots*, in "Journal of Computational and Applied Mathematics", vol. 162, n^o 1, 2003, p. 33-50.

[16] M. SAFEY EL DIN, E. SCHOST. *Polar varieties and computation of one point in each connected component of a smooth real algebraic set*, in "International Symposium on Symbolic and Algebraic Computation 2003 - ISSAC'2003, Philadelphie, USA", J. SENDRA (editor)., ACM Press, August 2003, p. 224-231.

[17] D. STEHLÉ, V. LEFÈVRE, P. ZIMMERMANN. *Worst Cases and Lattice Reduction*, in "16th IEEE Symposium on Computer Arithmetic 2003 - ARITH-16'03, Santiago de Compostela, Spain", June 2003, p. 142-147.

Articles in referred journals and book chapters

[18] R. Brent, S. Larvala, P. Zimmermann. *A Primitive Trinomial of Degree 6972593*, in "Mathematics of Computation", to appear, 2004.

[19] D. Defour, G. Hanrot, V. Lefèvre, J.-M. Muller, N. Revol, P. Zimmermann. *Proposal for a Standardization of Mathematical Function Implementation in Floating-Point Arithmetic*, in "Numerical Algorithms", vol. 37, n° 1-2, 2004, p. 367–375.

[20] Y. Gerard, I. Debled-Rennesson, P. Zimmermann. *An elementary digital plane recognition algorithm*, in "Discrete Appl. Math.", special issue devoted to IWCIA 2003, to appear, 2004.

[21] G. Hanrot, M. Quercia, P. Zimmermann. *The Middle Product Algorithm I. Speeding up the division and square root of power series*, in "Applicable Algebra in Engineering, Communication and Computing - AAECC", vol. 14, n° 6, Jan 2004, p. 415-438.

[22] G. Hanrot, P. Zimmermann. *A long note on Mulders' short product*, in "Journal of Symbolic Computation", vol. 37, 2004, p. 391–401.

[23] D. Lazard. *Injectivity of real rational mappings: the case of a mixture of two Gaussian laws*, in "Math. Comput. Simulation", vol. 67, n° 1-2, 2004, p. 67–84.

[24] B. Mourrain, F. Rouillier, M.-F. Roy. *Bernstein's basis and real root isolation*, in "Current trends in combinatorial and computational geometry", to appear, vol. Special volume devoted to special program. “Discrete and Computational Geometry” at MSRI in its series “Mathematical Sciences Research Institute Publications”, Cambridge University Press, 2004.

[25] N. Revol, F. Rouillier. *Motivations for an arbitrary precision interval arithmetic and the MPFI library*, in "Reliable Computing", Accepted for publication, 2004.

[26] M. Safey El Din, E. Schost. *Properness defects of projection functions and computation of at least one point in each connected component of a real algebraic set*, in "Journal of Discrete and Computational Geometry", September 2004.

[27] M. Safey El Din, P. Trébuchet. *Strong bihomogeneous Bézout theorem and degree bounds for algebraic optimization*, in "submitted to Journal of Pure and Applied Algebra", 2004.

Publications in Conferences and Workshops

[28] G. Ars, J. Faugère. *Comparison of XL and Gröbner basis algorithms over Finite Fields*, in "AsiaCrypt 2004", 2004.

[29] M. Bardet, J. Faugère, B. Salvy. *On the complexity of Gröbner basis computation of semi-regular overdetermined algebraic equations*, in "Proceedings of the International Conference on Polynomial System Solving", 2004, p. 71–74.

[30] A. Basiri, A. Enge, J.-C. Faugère, N. Gürel. *Implementing the arithmetic of $C_{3,4}$ curves*, in "Sixth Algorithmic Number Theory Symposium, ANTS VI proceedings (Vermont, USA, June 13–18, 2004)", D. Buell (editor)., LNCS, vol. 3076, Springer-Verlag, 2004, p. 57–71.

[31] J. Calmet, V. Lefèvre. *Toward the Integration of Numerical Computations into the OMSCS Framework*, in "7th International Workshop on Computer Algebra in Scientific Computing - CASC'2004, Saint Petersburg, Russia", Jul 2004, p. 71-79, http://www.loria.fr/publications/2004/A04-R-453/A04-R-453.ps.

[32] L. Fousse, S. Schmitt. *A comparison of polynomial evaluation schemes*, in "6th Conference on Real Numbers and Computers (RNC6), Dagstuhl, Germany", Nov 2004, http://www.loria.fr/publications/2004/A04-R-452/A04-R-452.ps.

[33] D. Lazard. *Gröbner bases for implicitisation, local inversibility and related questions*, in "AGGM (Algebraic Geometry and Geometric Modeling)", 2004.

[34] D. Lazard, F. Rouillier. *Solving Parametric Polynomial Systems*, in "International Conference on Polynomial System Solving", 2004, p. 53–55.

[35] C. Le Guernic, F. Rouillier, M. Safey El Din. *Computing sampling points in a semi-algebraic set defined by non-strict inequalities, Application to Pattern-Matching Problems*, in "EACA, Santander", ACM Press, 2004.

[36] V. Lefèvre. *The Generic Multiple-Precision Floating-Point Addition With Exact Rounding (as in the MPFR Library)*, in "6th Conference on Real Numbers and Computers (RNC6), Dagstuhl, Germany", Nov 2004, p. 135-145, http://www.loria.fr/publications/2004/A04-R-450/A04-R-450.ps.

[37] F. Rouillier. *On the Rational Univariate Representation*, in "International Conference on Polynomial System Solving", 2004, p. 75–79.

[38] M. Safey El Din. *Generalized critical values and solving polynomial inequalities*, in "Proceedings of the International Conference on Polynomial System Solving", Nov 2004, p. 40–42.

[39] D. Stehlé, P. Zimmermann. *A Binary Recursive Gcd Algorithm*, in "ANTS-VI, Burlington (Vt), USA", LNCS, Springer, May 2004, p. 411-425.

[40] P. Trébuchet. *Generalized normal forms for positive dimensional ideals*, in "ICPSS 2004", 2004.

Internal Reports

[41] D. Lazard, F. Rouillier. *Solving Parametric Polynomial Systems*, Technical report, n° RR-5322, INRIA, October 2004, http://www.inria.fr/rrrt/rr-5322.html.

[42] V. Lefèvre, P. Zimmermann. *Arithmétique flottante*, Rapport de recherche, n° RR-5105, INRIA, Feb 2004, http://www.inria.fr/rrrt/rr-5105.html.

[43] B. MOURRAIN, F. ROUILLIER, M.-F. ROY. *Bernstein's basis and real root isolation*, Technical report, n^o RR-5149, INRIA, 2004, http://www.inria.fr/rrrt/rr-5149.html.

[44] D. STEHLÉ, P. ZIMMERMANN. *Gal's Accurate Tables Method Revisited*, 23 pages, Rapport de Recherche, n^o RR-5359, INRIA, October 2004, http://www.inria.fr/rrrt/rr-5359.html.

Bibliography in notes

[45] S. BASU, R. POLLACK, M.-F. ROY. *Algorithms in real algebraic geometry*, Algorithms and Computations in Mathematics, vol. 10, Springer-Verlag, 2003.

[46] A. BOSTAN, B. SALVY, É. SCHOST. *Fast Algorithms for Zero-Dimensional Polynomial Systems Using Duality*, in "Applicable Algebra in Engineering, Communication and Computing", vol. 14, n^o 4, 2003, p. 239–272.

[47] G. E. COLLINS, H. HONG. *Partial Cylindrical Algebraic Decomposition*, in "Journal of Symbolic Computation", vol. 12, n^o 3, 1991, p. 299–328.

[48] G. DOS REIS, B. MOURRAIN, P. TRÉBUCHET, F. ROUILLIER. *An Environment for Symbolic and Numeric Computation*, in "International Congress of Mathematical Software (ICMS'2002), Beijing, China", August 2002.

[49] B. MOURRAIN, M. VRAHATIS, J. YAKOUBSOHN. *On the Complexity of Isolating Real Roots and Computing with Certainty the Topological Degree*, in "Journal of Complexity", vol. 18, n^o 2, 2002, p. 612–640.

[50] J. E. OMRI. *Analyse géométrique et cinématique des mécanismes de type manipulateur*, PhD Thesis, Université de Nantes, February 1996.

[51] F. ROUILLIER. *Solving zero-dimensional systems through the rational univariate representation*, in "Journal of Applicable Algebra in Engineering, Communication and Computing", vol. 9, n^o 5, 1999, p. 433–461.

[52] F. ROUILLIER, P. ZIMMERMANN. *Efficient Isolation of Polynomial Real Roots*, in "Journal of Computational and Applied Mathematics", vol. 162, n^o 1, 2003, p. 33–50.

[53] M. SAFEY EL DIN, E. SCHOST. *Polar varieties and computation of one point in each connected component of a smooth real algebraic set*, in "International Symposium on Symbolic and Algebraic Computation 2003 - ISSAC'2003, Philadelphie, USA", J. SENDRA (editor)., ACM Press, August 2003, p. 224–231.

[54] É. SCHOST. *Sur la résolution des systèmes polynomiaux à paramètres*, Ph. D. Thesis, École polytechnique, 2000.

[55] V. SHOUP. *Efficient Computation of Minimal Polynomials in Algebraic Extensions of Finite Fields*, in "ISSAC'99", S. DOOLEY (editor)., ACM Press, 1999, p. 53–58.

[56] J. USPENSKY. *Theory of equations*, MacGraw Hill, 1948.

[57] V. WEISPFENNING. *Solving parametric polynomial equations and inequalities by symbolic algorithms*, World Scientific, 1995.

[58] P. WENGER, J. E. OMRI. *Changing Posture for Cuspidal Robot Manipulators*, in "Proceedings of the 1996 IEEE Int. Conf on Robotics and Automation", 1996, p. 3173–3178.

[59] P. WENGER. *Some Guidelines for the Kinematic Design of New Manipulators*, in "Mechanism and Machine Theory", vol. 35, 2000, p. 437–449.

[60] P. WENGER. *Classification of 3R Positioning Manipulators*, in "Journal of Mechanical Design", vol. 120, June 1998, p. 327–332.