INRIA

INSTITUT NATIONAL DE RECHERCHE EN INFORMATIQUE ET EN AUTOMATIQUE

Project-Team Spaces

Solving Problems through Algebraic Computation and Efficient Software

Lorraine - Rocquencourt

THEME 2B

Activity Report

2003

Table of contents

1. Team **1**
2. Overall Objectives **2**
3. Scientific Foundations **2**
3.1. Historical Background 2
3.2. Zero-Dimensional Polynomial Systems 2
3.3. Polynomial Systems of Positive Dimension 3
3.4. Arithmetics 3
3.5. Solving Problems — Applications 4
4. Application Domains **5**
4.1. Introduction 5
4.2. Robotics 5
4.2.1. Parallel Manipulators 5
4.2.2. Serial Manipulators 5
4.3. Geometrical Reasoning 6
4.4. Cryptography 6
5. Software **7**
5.1. Introduction 7
5.2. RS/RealSolving 7
5.3. Gb/FGb 8
5.4. Implicit Curves Drawing 8
5.5. RAGLib 8
5.6. Triangular Decomposition 9
5.7. Epsilon 9
5.8. MPFR 9
5.9. MPFI 10
5.10. MPAI 10
5.11. Exhaustive Tests of the Mathematical Functions 10
5.12. Multiplication by an Integer Constant 10
5.13. Finite Field Arithmetic 10
5.14. UDX 11
5.15. Interfaces 11
5.16. NetTask 11
5.17. TSPR 12
6. New Results **12**
6.1. Polynomial Arithmetic 12
6.2. Floating-Point Arithmetic 13
6.3. Cryptography 13
6.4. Geometric Reasoning 13
6.5. Real Solutions of Systems Depending on Parameters 14
6.6. Computation of one Point in each Connected Component of a Real Algebraic Set 15
6.7. Computation of one Point in each Connected Component of a Semi-Algebraic Set Defined by non-strict Inequalities 16
6.8. Algebraic Identification Problems and Algebraic Systems with Approximated Coefficients 16
6.9. Parallel Manipulators 16
6.10. Cuspidal Robots 17
7. Contracts and Grants with Industry **17**
7.1. MuPAD-Scilab Interface 17

7.2. Machine-Tools 17
8. Other Grants and Activities 18
8.1. National Initiatives 18
8.1.1. CNRS Grant “Robots Cuspidaux” 18
8.1.2. Ministry Grant (ACI) “Cryptologie” 18
8.1.3. Ministry Grant (ACI) “Jeunes” 18
8.2. European Initiatives 18
8.2.1. Real Algebraic and Analytic Geometry Network 18
9. Dissemination 19
9.1. Leadership within Scientific Community 19
9.2. Invited Conferences 19
9.3. Teaching 19
10. Bibliography 20

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.*

Head of project-team
Daniel Lazard [professor, University Pierre et Marie Curie (Paris 6, UPMC), seconded to INRIA]

Vice-head of project team
Paul Zimmermann [research director, INRIA]

Administrative assistant
David Massot [UPMC, part time]
Hélène Zganic [LORIA, part time]

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, from October 2003]
Dongming Wang [research scientist, CNRS]

Staff member (University)
Philippe Aubry [assistant professor, UPMC]
Luc Rolland [teaching assistant, University Henri Poincaré (UHP), until August 2003]
Mohab Safey El Din [assistant professor, UPMC]
Philippe Trébuchet [teaching assistant, UPMC]

Technical staff
Patrick Pélissier [junior technical staff, INRIA, from September 2003]
Étienne Petitjean [project technical staff, INRIA, until May 2003]

Ph. D. student
Gwenolé Ars [DGA, defense planned in 2004]
Magali Bardet [teaching assistant, defense planned in 2004]
Abdolali Basiri [Sfere grant, defended on 21 November 2003]
Jean-Paul Cerri [defense planned in 2005]
Solen Corvez [BDI CNRS, defense planned in 2004]
Laurent Fousse [ENS grant, defense planned in 2006]
Nicolas Gürel [LIX, defended on 15 December 2003]
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]

Post-doctoral fellow
Bill Allombert [ministry grant "Cryptologie", until August 2003]

Visiting scientist
Richard Brent [Prof. at Oxford University, during August 2003]

Student intern
Colas Le Guernic [May-December 2003]
Renaud Lifchitz [July-August 2003]
Jean-René Reinhard [April-June 2003]
Raphaël Rigo [May 2003]

2. Overall Objectives

The objectives of the team are well summarized by the development of its acronym in English and in French: *Solving Problems through Algebraic Computation and Efficient Software* and *Systèmes Polynomiaux, Arithmétiques, Calculs Efficaces et Sûrs*.

The main objective is to solve systems of polynomial equations and inequations (*Systèmes Polynomiaux*). We emphasize on algebraic methods (*Algebraic Computation*) 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 (*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 (*Calculs Efficaces et Sûrs*). 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 Bézout, 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 cryptography, one may prescribe that all the solutions are in K by adding the equation $x^q - q = 0$ for each unknown x.

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

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.

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 cryptography purposes.

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 ($P_i > 0$ or $P_i \geq 0$)[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

[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.

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 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 cryptography 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 of 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 cryptography, 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 [6][5]. The results were often illustrated by calculations of Gröbner bases done with the Gb software. One of the remarkable points of this study is certainly the classification suggested in [3]. The next efforts were related to the real roots and the effective calculation of the solutions [8]. 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 ([44], [46], [45] and [43]) 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 (see Section 6.4).

4.4. Cryptography

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.

A new result is that the weakness of the systems of equations coming from HFE instances can be *explained* by the algebraic properties of the secret key (work presented at Crypto’2003 in collaboration with A. Joux). From this study we are able to predict the maximal degree occurring in the Gröbner basis computation, so that we can establish precisely the complexity of the Gröbner attack and compare it with the theoretical bounds.

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).

We give a definition of weak regular sequences and their associated notion of regularity. The motivation for studying such regular sequences is that "random" sequences are weak regular, and the degree of regularity is closely related to the global cost of the Gröbner basis computation for a graded admissible monomial order. Using the Gröbner approach for solving systems, and in particular the F_5 algorithm, we show that for (weak) regular sequences the computation of the Gröbner basis using F_5 can be followed step by step, and the size of all matrices made explicit. We deduce the regularity of these sequences, and using asymptotic analysis methods we compute the asymptotic expansion of the maximal degree D_{max}.

We show for instance that for n *quadratic equations over GF*(2) we have

$$D_{max} \simeq 0.0900\, n + 1.00\, n^{1/3} - 1.58 + O(n^{-1/3}).$$

The global cost of the Gröbner basis computation is then

$$1.35^{\omega(n+0.77n^{\frac{1}{3}}+O(\log(n)))}$$

with ω the exponent of matrix multiplication complexity.

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 (A. Basiri, N. Gürel, J.-C. Faugère, in collaboration with the TANC team).

As an application of the LLL algorithm we have presented (A. Basiri and J.-C. Faugère) an algorithm for the transformation of a Gröbner basis of an ideal with respect to any given ordering into a Gröbner basis with respect to any other ordering. This algorithm is based on a modified version of the LLL algorithm. The worst case theoretical complexity of this algorithm is not necessarily better than the complexity of the FGLM algorithm (a well known algorithm); but when the output (the final Gröbner basis) is small then algorithm is more efficient.

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 2.1: zero-dimensional systems, systems of positive dimension, arithmetics, solving problems and applications.

5.2. RS/RealSolving

Participant: Fabrice Rouillier.

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

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

Participant: Jean-Charles Faugère.

Key words: *Gröbner basis*, *complex root*.

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 standalone 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

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

Key words: *implicit curve*, *drawing*.

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 2000 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

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

Key words: *polynomial system*, *real solution*.

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

Participant: Philippe Aubry.

Key words: *polynomial system*, *triangular set*, *polynomial gcd*, *unmixed decomposition*.

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. Epsilon

Participant: Dongming Wang.

Key words: *polynomial elimination*, *triangular set*, *zero decomposition*, *polynomial system*, *geometric reasoning*.

Epsilon is a software library for polynomial elimination and decomposition with (geometric) applications. Implemented in Maple and Java, the library has 8 modules and contains more than 70 functions, which allow one to:

- triangularize systems of multivariate (differential) polynomials,
- decompose polynomial systems into triangular systems of various kinds (regular, normal, simple, irreducible, or with projection property),
- decompose algebraic varieties into irreducible or unmixed sub-varieties,
- decompose polynomial ideals into primary components,
- factorize polynomials over algebraic extension fields,
- solve systems of polynomial equations and inequations,
- and handle and prove geometric theorems automatically.

The entire library with documentation, examples, and Maple worksheets has been made publicly available at

http://www-calfor.lip6.fr/~wang/epsilon/

and will also be distributed with a book entitled “Elimination Practice: Software Tools and Applications” [12] (in press by Imperial College Press, London).

5.8. MPFR

Participants: Vincent Lefèvre, Patrick Pélissier, Paul Zimmermann [contact].

Key words: *floating-point number*, *arbitrary precision*, *exact rounding*, *IEEE 754*.

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 exact to the last bit, according to the given rounding mode.

The main objective of year 2003 was to consolidate the current version of MPFR. As a result, a new version, MPFR-2.0.2, was released in November. This is a bug-fix release for MPFR-2.0.1.

In September 2003, Patrick 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 first 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 deal with special values (Not-a-Number, infinities and zeroes). A first implementation of this new scheme in the development version of MPFR already showed a speedup of about 30% with respect to MPFR-2.0.1.

5.9. MPFI

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

MPFI is a library for multiprecision interval arithmetic, written in C (approximately 1000 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.

5.10. 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 1000 lines of code. The algorithms proposed in MPAI include Karatsuba's product and the short product, ...The code is available.

5.11. 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 exact rounding.

5.12. Multiplication by an Integer Constant

Participants: Vincent Lefèvre [contact], Raphaël Rigo.

Raphaël Rigo did a very efficient implementation in ISO C of Lefèvre's algorithm to multiply a number by a (possibly multiple-precision) integer constant. This implementation could be used in compilers, when they need to generate code for an integer multiplication, using only shifts, additions and subtractions. It may also be very useful in the future code of the exhaustive tests of the mathematical functions.

5.13. Finite Field Arithmetic

Participants: Bill Allombert, Guillaume Hanrot [contact], Renaud Lifchitz.

In relation with the work done in cryptology in the project, it has been decided to make a full implementation of finite field arithmetic, with the goal to be at the same time complete and efficient in all situations; namely, if the small and large characteristics are rather well understood, medium characteristic and degree have not been really seriously investigated. In practice, even the question of how to implement basic operations in characteristic three is already the subject of recent work. A recent trend in cryptology is to use that kind of medium characteristic fields.

This package is still at a very experimental stage, and currently involves a set of routines for finite field computations using the Kronecker representation for elements of algebraic extensions, and a low-level set of optimized assembly routines for characteristic two operations.

5.14. UDX

Participant: Fabrice Rouillier.

Key words: *protocol*, *binary data*.

Glossary

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 (9500 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 [42], MuPAD/Scilab interface distributed with MuPAD 2.5.1).

5.15. Interfaces

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

Key words: *interfaces*.

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.

5.16. NetTask

Participants: Jean-Charles Faugère, Étienne Petitjean [SEDRE-LORIA], Fabrice Rouillier [contact].

We benefited during nearly one year of the assistance of Étienne Petitjean (SEDRE-LORIA) for the development of a series of tools to make possible the use of machines and software via a web interface. The developed tool (about 11,000 lines in C++) makes possible the use of software installed on a network which then becomes usable even if it is not ready to be distributed, and thus allows the experimental validation of algorithms proposed in articles, or allows an external user to have a more precise idea of the possibilities offered by recent algorithmic progress.

NetTask is very flexible since it doesn't need any change in the software to be demonstrated. It is compatible with the current safety requirements and contains some powerful tools:

- allocation of the tasks launched by the users according to the availability of the software but also to the load of the nodes on a given network of machines;
- system of queue for the tasks;
- complete control of the launched tasks by the user himself;
- many options of configuration such as for example the declaration of the machines and tasks available on a given network, the maximum number of tasks per user.

This software will be registered at the "*Agence de Protection des Programmes*" (APP).

5.17. TSPR

Participants: Luc Rolland, Fabrice Rouillier [contact].

Key words: *parallel manipulator*, *trajectory*.

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.

6. New Results

6.1. Polynomial Arithmetic

Participants: Richard Brent, Guillaume Hanrot, Paul Zimmermann.

The "short product" algorithm proposed by Mulders in 2000 was revisited by G. Hanrot and P. Zimmermann [21]. The main results are an explicit expression for the optimal cutoff point in Mulders' algorithm used together with Karatsuba's algorithm for multiplication, and a new "even-odd" short product that behaves theoretically as well as Mulders' algorithm with optimal cutoff, and even better in practice.

The search for primitive trinomials of degree 6972593 over GF$(2)[x]$ was completed in 2003. This search started in 2001 with Richard Brent (Oxford) and Samuli Larvala (Helsinki), thanks to the help of the CINES (*Centre Informatique National de l'Enseignement Supérieur*). We found only one primitive trinomial, namely:

$$p = x^{6972593} + x^{3037958} + 1.$$

This result will be published as a short note in *Mathematics of Computation* [16]. As a consequence, the sequence $f \to xf \mod p$ over GF(2) has period $2^{6972593} - 1$.

A program was also written in the NTL library from V. Shoup to double-check the computations for degrees 3021377 and 6972593. This program extends the NTL library, by implementing the Toom-Cook multiplication over GF$(2)[x]$, together with Knuth-Schönhage's algorithm for fast gcd.

6.2. Floating-Point Arithmetic

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

The proposal for a standardization of mathematical function implementation in floating-point arithmetic presented at the SCAN-2002 conference was accepted for publication in *Numerical Algorithms* [19]. This is a joint work with the ARENAIRE team (INRIA Rhône-Alpes).

L. Fousse and P. Zimmermann presented at the *Real Numbers and Computers* conference (RNC'5) a simpler and formal proof of the "accurate summation" algorithm proposed by Demmel and Hida. This new proof was partially checked with the Coq proof assistant by L. Fousse [31].

Concerning the famous Table Maker's Dilemma, D. Stehlé, V. Lefèvre and P. Zimmermann designed a new algorithm based on lattice reduction to find the worst cases for the rounding of an analytic function [36]. The previous best known algorithm for that problem was Lefèvre's algorithm, with a complexity of $2^{2n/3+\varepsilon}$ for n-bit numbers. The new algorithm achieves a complexity of $2^{5n/7+\varepsilon}$. A first prototype implementation was written using the GNU MP library, and some first results were obtained, for example the following nice worst case for the function 2^x with 40 bits of precision (number written in binary):

$$x = -0.00010101010101100101111100110011101010000,$$

where 2^x contains 53 consecutive ones after the round bit!

The technique developed in [36] has been modified by D. Stehlé in order to break a cryptographic scheme proposed by J.E. Littlewood in the 50's [40]. The underlying idea of the scheme was to use the randomness of low order bits of $f(x)$, where f is a known given mathematical function (e.g. $f = \log$ in the initial scheme). The cryptanalysis works in polynomial time under a reasonable heuristic related to Coppersmith's technique to find the small roots of multivariate modular polynomials.

6.3. Cryptography

Participant: Gwenolé Ars, Abdolali Basiri, Magali Bardet, Jean-Charles Faugère, Nicolas Gürel.

A new result is that the weakness of the systems of equations coming from HFE instances can be *explained* by the algebraic properties of the secret key (Crypto'2003 in collaboration with A. Joux). From this study we are able to predict the maximal degree occurring in the Gröbner basis computation. So that we can establish precisely the complexity of the Gröbner attack and we can compare it with the theoretical bounds.

Very recently, we have extended the previous complexity bounds (Macaulay bounds) to some algebraic systems occurring specifically in cryptographic problems: over-determined polynomial systems (with M. Bardet, J.-C. Faugère) and in collaboration with the Algo team. The complexity bounds we give are upper bounds for either the security of the cryptosystem in the first case, or the feasibility of the attack in the second case.

Another application of Gröbner bases to cryptography is the implementation of fast arithmetic of Jacobian groups of general curves: recently we obtained optimized formulas (with only 130 multiplications and 10 squares) for a basic arithmetic operation in the Jacobian group of C_{34} curves (A. Basiri, N. Gürel, J.-C. Faugère) and in collaboration with the TANC team.

We have presented (A. Basiri and J.-C. Faugère, Issac'2003) an algorithm for the transformation of a Gröbner basis of an ideal with respect to any given ordering into a Gröbner basis with respect to any other ordering. This algorithm is based on a modified version of the LLL algorithm.

6.4. Geometric Reasoning

Participant: Dongming Wang.

Key words: *automated drawing*, *geometric diagram*, *implicitization*, *offsetting*.

Automated generation/drawing of diagrams is desirable for the manipulation of geometric theorems and in computer aided geometric design. We have proposed and implemented an approach which permits one to draw diagrams automatically and effectively for theorems in plane Euclidean geometry [27]. This approach uses Maple for algebraic computation and Java for graphic drawing. The drawn diagrams may be modified and animated with a mouse click and dragging. The drawing program has been included as a function in the GEOTHER module of Epsilon [12].

During the completion of GEOTHER (an environment for handling and proving geometric theorems that has been developed since 1990), we have proposed a primitive methodology to design and implement a geometric-object-oriented language for symbolic geometric reasoning and computation [37]. In such a language, basic geometric objects such as line, triangle, circle, curve, plane, and surface will be constructed as general classes. Concrete geometric objects may be instantiated by assigning values to some of the attributes of the classes and most of them will remain symbolic. These objects in the language will interact through standard geometric relations. Making use of techniques from symbolic and numerical computation, the language will provide a basis for implementing advanced methods for geometric reasoning, computation, and visualization.

The implicitization of rational parametric curves and surfaces is a fundamental problem in computer aided geometric design. The existing methods for this problem are based mostly on the computation of Gröbner bases, resultants, and the techniques of moving curves and surfaces. We have introduced a new and simple method of implicitization [41] which works well in general and better in the case where base points are present. Our method uses the principle of undetermined coefficients to reduce the problem of implicitization to the problem of solving sparse, partially triangularized systems of linear equations. An implementation of this method is included in the Epsilon library [12].

The curve or surface defined by the implicit equation computed using the above-mentioned methods may not be the same as that defined by the rational parametric equations. We have shown [28] how to compute the implicit equations and inequations that define exactly the same geometric object as that defined by the given rational parametric equations by means of regular systems. The same technique is applied to the computation of quasi-offsets to algebraic curves and surfaces.

6.5. Real Solutions of Systems Depending on Parameters

Participants: Philippe Aubry, Daniel Lazard, Fabrice Rouillier.

Most of the applications we recently solved (celestial mechanics, cuspidal robots, statistics, etc.) require the study of semi-algebraic systems depending on parameters. Although we covered these subjects in an independent way, it currently releases from these experiments some general algorithms for the resolution of this type of systems.

The general philosophy consists in studying the generic solutions independently from algebraic sub-varieties (which we call from now on discriminating varieties) of dimension lower than the semi-algebraic variety considered. The study of the varieties thus excluded can be done separately to obtain a complete answer to the problem, or is simply neglected if one is interested only in the generic solutions, which is the case in some applications.

We showed that our definition of the discriminating varieties is optimal for the various concepts of “generic solutions” used in practice (compact fibers, continuity of the complex solutions, applicability of the implicit function theorem, etc.). We also showed that we know how to calculate them exactly (and not a sub-variety with the same dimension).

Several computational strategies are proposed. A first decomposition in equidimensional components as zeroes of radical ideals simplifies the computation and the use of the discriminating varieties. This preliminary computation is however sometimes very expensive, so we developed adaptive solutions that compute such decompositions only when needed. The principal progress is that the methods obtained are fast on the easy (generic) problems and slower on the problems with strong geometrical content.

Once discriminant sub-varieties are computed, several possibilities are offered to us. In the case of low-dimension problems, one can continue rather easily without the study of the discriminant varieties by deducing their topology from the study of the generic solutions of the system. In the case of problems of greater dimension (for example the application on cuspidal robots [18]), one can calculate a partial cellular decomposition of the parameter's space according to a fixed property (for example a constant number of real solutions).

Currently, we focus on problems that require a decomposition of the system: this is the hard step in many non generic situations. When inductively decomposing the radical of an ideal given by a lexicographical Gröbner basis into equidimensional components (that may be represented by a triangular set), we showed that some computations can be avoided. We are now studying the choices offered among the strategies that may be deduced from this property for obtaining the better behavior in practice. Moreover, a triangular decomposition of the radical of an ideal is generally redundant in the sense that the ideal represented by a component may be included in another one. A way of making the decomposition non redundant is to compute a Gröbner basis of each component. Since it may be costly, we developed some techniques that avoid this computation as much as possible.

6.6. Computation of one Point in each Connected Component of a Real Algebraic Set

Participants: Philippe Aubry, Fabrice Rouillier, Mohab Safey El Din.

Given an algebraic variety $\mathcal{V} \in \mathbb{C}^n$, the problem of computing at least one point in each connected component of the real counterpart of $\mathcal{V}$ is motivated by the fact that this specification leads to asymptotically optimal algorithms for deciding the emptiness of semi-algebraic sets, or for counting their number of connected components.

Most of the asymptotically optimal algorithms solving that problem are based on computations of critical loci of a polynomial mapping reaching its extrema on each connected component of the real algebraic set considered. This is the case for example for the projection function on a generic line when the studied variety is compact or for the distance function restricted to a generic point.

A justification of our current orientation (positioning with respect to the theoretical algorithms and existing practical solutions) and a short summary of our results concerning the use of the distance function is proposed in [24].

Although it is not stated from the theoretical complexity viewpoint, computing critical points of linear functions, i.e. projections, is better than computing the critical points of distance functions from a computational viewpoint. Nevertheless, as mentioned above, these computations are not sufficient to be sure to reach each connected component of the studied real algebraic sets in non-compact situations.

We proved, with É. Schost, that introducing the study of loci of non-properness of projection functions additionally to the computation of the critical loci of these functions, allows to reach each connected component of a real algebraic set, thus dropping the compactness assumption historically associated to the use of projection functions in this context. This algorithm has been implemented in the RAGLib library (see Section 5.5) and some experiments showed its efficiency compared to the implementation of the algorithm of Aubry, Rouillier and Safey El Din (see [14]). These results will appear in the *Journal of Discrete and Computational Geometry* (see [26]).

A detailed complexity analysis of this algorithm in some *generic* situations showed that it can be far improved. In a work in common with É. Schost, we proved that up to a *generic* linear change of variables some properness properties can be ensured on nested critical loci of projection functions on nested linear subspaces. Taking advantage of these properties and of a more accurate geometrical analysis, we obtain an algorithm whose theoretical complexity, expressed in a model based on Straight Line Programs, is polynomial in some intrinsic geometric degree bounds, and asymptotically optimal. This work has been published in the *Proceedings of the International Symposium on Symbolic and Algebraic Computation* (ISSAC'2003, see [35]). Because of the required linear change of variables, the use of this algorithm in the frame of Gröbner bases cannot lead to better practical results.

In further works, we plan to generalize this last algorithm by avoiding the first step of change of variables, and by keeping on the accurate analysis of the theoretical complexity of our algorithms.

6.7. Computation of one Point in each Connected Component of a Semi-Algebraic Set Defined by non-strict Inequalities

Participants: Colas Le Guernic, Mohab Safey El Din.

Computing at least one point in each connected component of a semi-algebraic set defined by a polynomial system of equations and *non-strict* inequalities has applications in pattern matching, chemistry, and robotics. Curiously, this problem is usually tackled by rewriting polynomial systems of equations and *non-strict* inequalities as boolean combinations of polynomial systems of equations and *strict* inequalities. This induces a combinatorial factor in the complexity of the algorithms derived from this method which is computed over the Puiseux series field with rational coefficients.

During its undergraduate studies (see [39]) supervised by M. Safey El Din, C. Le Guernic had to study properties of semi-algebraic sets defined by polynomial systems of equations and non-strict inequalities and to take advantage of the fact that they are closed subsets for the strong topology like real algebraic sets. These results allow to derive an algorithm taking as input a polynomial system of equations and *non-strict* inequalities and computing at least one point in each connected component of the associated semi-algebraic set, exclusively using algorithms computing at least one point in each connected component of a real algebraic set defined by polynomial systems over $\mathbb{Q}$ and avoiding the afore-mentioned combinatorial factor. The complexity of this algorithm has also been analyzed and under some assumptions of *genericity*, it is asymptotically optimal. The implementation of this algorithm, which will be integrated in the RAGLib library, is now used to solve a pattern matching problem. A paper describing these results is under preparation.

6.8. Algebraic Identification Problems and Algebraic Systems with Approximated Coefficients

Participants: Daniel Lazard, Mohab Safey El Din.

Given two integers n and ℓ, the problem of inverting a rational mapping $\Phi : \mathbb{R}^n \to \mathbb{R}^\ell$ given an approximation of a point in the image of $\mathbb{R}^n$ by Φ has many applications in particular in statistics. We have shown that existing computing tools allow to prove exact inversibility [22]. Relating *numerical stability* with geometrical properties of the graph of Φ, we show how to compute a subset of the Zariski closure of the image of Φ in the vicinity of which there can be no numerical stability. We also give a symbolic-numeric procedure to solve algebraic identification problems. These results have been applied to the identification of a mixture of two normal distributions. The corresponding paper is under preparation.

6.9. Parallel Manipulators

Participants: Jean-Charles Faugère, Luc Rolland, Fabrice Rouillier.

In terms of papers, the principal news of the year is the defense of Luc Rolland's Ph. D. thesis in December 2003. The last phase of drafting having required a strong share of experimentation, that was the occasion for us to update the algorithms (and their implementations) present in the simulator on which we have worked for several years.

The main algorithmic tool present in this simulator (partially presented in [34]) is an hybrid method (mixing computer algebra, numerical calculation and interval analysis) for the resolution of the direct geometrical model.

The main work was to develop a semi-numerical method (based on Newton's method), powerful in terms of computation time (4000 calculations per minute) and certified, i.e. always returning a correct result: the answer is either a confidence interval on the provided numerical approximation, or a message of failure. We ensured that failure remains exceptional (less than 10 percent of the practical problems) and, when it occurs, efficiency is obtained by a special version of the F4 algorithm for Gröbner bases computation and a very efficient version (adapted to that particular case of systems) of the Rational Univariate Representation algorithm.

The undertaken experiment, in particular within the framework proposed in the thesis of Luc Rolland, shows that this tool is of a great effectiveness both in terms of computing time and in terms of quality of the provided results, for the complete diagnostic of a control device for high speed machining (performance of the CAD software, computational strategies and their implementation, real performances, in terms of precision, of these robots, etc.).

For obvious reasons of confidentiality (work with CMW), the related software is not available and even less downloadable. On the other hand, it should be accessible at the end of 2003 via the NetTask server for our partners.

6.10. Cuspidal Robots

Participants: Solen Corvez, Fabrice Rouillier.

An extended version of paper presented at the ADG'2002 conference last year has been accepted for publication [18].

From the computer algebra point of view, in order to solve the stated problem (see Section 4.2), it remains to solve a system of equalities and inequalities depending on three parameters which correspond to the design parameters of this kind of robots. The method basically used is ad hoc, and no known automatic computer algebra methods were able to solve completely the problem.

From a robotic point of view, the result obtained is a full classification of a class of serial robots with three degrees of freedom according to their cuspidal character. Since then, towards the end of the Math-Stic project "*Robots Cuspidaux*" (see Section 8.1), these results were simplified and analyzed to allow a better description of the workspace of such mechanisms.

The computations done for this application were critical also for the development of the general and systematic methods mentioned in Section 6.5. We have shown that these general methods can now be used in place of ad hoc computations to calculate the same classification and a recent experiment even shows that they allow to relax one more parameter and thus to solve a more general problem.

Currently, we work at simplifying the results obtained in the general case (decrease the size of the result in particular which is, for the moment, not exploitable in practice). The work with IRCCyN, one of the teams of the Math-Stic project "*Robots Cuspidaux*", will follow this year on the characterization of families of parallel robots.

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.

7.2. Machine-Tools

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

Since 1997, we work with the SME CMW (*Constructions Mécaniques des Vosges Marioni*) on the design of a parallel robot for high speed machining. A first agreement was signed in 1999 and CMW partially sponsored the Ph. D. of Luc Rolland.

The year 2002 was devoted to the search of partners to constitute a complete chain (software and electronic components) for proposing a new control device for this type of manipulators (work done in cooperation with the INRIA project COPRIN). Since all the partners agreed, the year 2003 was devoted to raise funds. An official project should start in 2004.

8. Other Grants and Activities

8.1. National Initiatives

8.1.1. CNRS Grant "Robots Cuspidaux"

Participants: Fabrice Rouillier [contact], Solen Corvez, Jean-Charles Faugère, Mohab Safey El Din.

The project "*Robots Cuspidaux et Racines Triples*" (Cuspidal Robots and Triple Roots) was held within the framework of a specific action CNRS Math-Stic. The project started in June 2002 and ended in June 2003.

Two CNRS teams ("*CMAO et Productique*" from IRCCyN, Nantes and "*Géométrie Réelle et Calcul Formel*" from IRMAR, Rennes) and two INRIA projects (COPRIN, SPACES) took part in it.

8.1.2. 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 will end in 2004. Results for 2003 are described in Sections 4.4 and 5.13.

The main goal of this project is to study the interactions between cryptography 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.

8.1.3. Ministry Grant (ACI) "Jeunes"

Participants: Jean-Charles Faugère [contact], Fabrice Rouillier, Daniel Lazard, Mohab Safey El Din, Philippe Aubry, Solen Corvez, Magali Bardet.

The main goal of this "ACI Jeunes" (Young Investigators Ministry Grant) is to study polynomial systems with parameters (that is to say with infinitely many solutions).

This projects ends in December 2003. The results obtained in 2003 are described in Section 6.5.

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 team 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. Leadership within Scientific Community

F. Rouillier was in the scientific committee of the “Summer School on Computer Tools for Real Algebraic Geometry” (July 2003, Rennes, France).

G. Hanrot and F. Rouillier took part in the organization of the “*Journées Nationales de Calcul Formel*” which took place at Luminy (France) in January 2003.

D. Wang is co-editor of two books [11][13] and member of the editorial board of the Journal of Symbolic Computation (published by Elsevier/Academic Press, London) and for the Book Series on Mathematics Mechanization (published by Science Press, Beijing). He served as chair of the Scientific Committee for the Summer School in Symbolic Computation (SSSC 2003, Huangshan, China, July 13–26, 2003; postponed to Summer 2004 due to SARS) and is member of the program committees for the:

- 6th Asian Symposium on Computer Mathematics (ASCM 2003) (Beijing, China, April 17–19, 2003);
- 2004 International Symposium on Symbolic and Algebraic Computation (ISSAC 2004) (Santander, Spain, July 4–7, 2004);
- 5th International Workshop on Automated Deduction in Geometry (Gainesville, USA, September 16–18, 2004);
- 7th International Conference on Artificial Intelligence and Symbolic Computation (AISC 2004) (Linz, Austria, September 20–22, 2004).

P. Zimmermann was member of the program committee of the Arith’16 conference (June 2003, Spain), and is 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/).

9.2. Invited Conferences

Jean-Charles Faugère gave invited conferences at the following events:

- NESSIE final conference, (Lund, Sweden),
- WCC’03 “International Workshop on Coding and Cryptography”.

9.3. Teaching

F. Rouillier gave 4 courses (2 hours each) during the “Summer School and Conference on Real Algebraic Geometry and its Applications” (August 2003, Trieste, Italy), 2 courses (1.5 hour each) during the “Summer School on Computer Tools for Real Algebraic Geometry” (July 2003, Rennes, France) and (in cooperation with Renaud Rioboo) one lecture (2 hours) during the “*École Adéquation Algorithmes Arithmétique*” (March 2003, Dijon, France).

M. Safey El Din gave a lecture at the “Summer School on Computer Tools for Real Algebraic Geometry” (July 2003, Rennes, France).

G. Hanrot gave the annual lecture of the *École Doctorale IAE+M* day (Computer Science, Control, Electronics, Mathematics) in Nancy.

V. Lefèvre and P. Zimmermann gave lectures (20 hours) about floating-point arithmetic with exact rounding 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. MORENO MAZA. *On the Theories of Triangular Sets.* in « Journal of Symbolic Computation, Special Issue on Polynomial Elimination », volume 28, 1999, pages 105–124.

[3] J. FAUGÈRE, D. LAZARD. *The Combinatorial Classes of Parallel Manipulators.* in « Mechanism and Machine Theory », volume 30, 1995, pages 765–776.

[4] G. HANROT, F. MORAIN. *Solvability by Radicals From an Algorithmic Point of View.* in « International Symposium on Symbolic and Algebraic Computation - ISSAC'2001, London, Ontario, Canada », ACM, B. MOURRAIN, editor, 2001.

[5] D. LAZARD. *Stewart Platforms and Gröbner Basis.* in « Proceedings of Advances in Robotics Kinematics », pages 136–142, 1992.

[6] D. LAZARD, J.-P. MERLET. *The (True) Stewart Platform has 12 Configurations.* in « Proceedings of IEEE International Conference on Robotics and Automation, San Diego », pages 2160–2165, May, 1994.

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

[8] F. ROUILLIER. *Real Root Counting For Some Robotics Problems.* in « Solid Mechanics and its Applications, Kluwer Academic Publishers », volume 40, 1995, pages 73–82.

[9] F. ROUILLIER. *Solving Zero-Dimensional Systems Through the Rational Univariate Representation.* in « Journal of Applicable Algebra in Engineering, Communication and Computing », number 5, volume 9, 1999, pages 433–461.

[10] J.-C. FAUGÈRE.. *A New Efficient Algorithm for Computing Gröbner Bases (F4).* in « Journal of Pure and Applied Algebra », number 1–3, volume 139, 1999, pages 61–88.

Books and Monographs

[11] F. CHEN, D. WANG, editors, *Geometric Computation.* World Scientific Publishing Co., Singapore New Jersey, December, 2003.

[12] D. WANG. *Elimination Practice : Software Tools and Applications.* Imperial College Press, London, December, 2003.

[13] D. WANG, editor, *Selected Lectures in Symbolic Computation.* Tsinghua University Press, Beijing, 2003, in Chinese.

Articles in referred journals and book chapters

[14] P. AUBRY, F. ROUILLIER, M. SAFEY EL DIN. *Real Solving for Positive Dimensional Systems.* in « Journal of Symbolic Computation », number 6, volume 34, 2002, pages 543-560.

[15] Y. BERTOT, N. MAGAUD, P. ZIMMERMANN. *A Proof of GMP Square Root.* in « Journal of Automated Reasoning », number 3-4, volume 29, December, 2002, pages 225–252, Special Issue on Automating and Mechanising Mathematics : In honour of N.G. de Bruijn.

[16] R. BRENT, S. LARVALA, P. ZIMMERMANN. *A Primitive Trinomial of Degree 6972593.* in « Mathematics of Computation », to appear.

[17] R. P. BRENT, S. LARVALA, P. ZIMMERMANN. *A Fast Algorithm for Testing Reducibility of Trinomials mod 2 and Some New Primitive Trinomials of Degree 3021377.* in « Mathematics of Computation », number 243, volume 72, July, 2003, pages 1443-1452.

[18] S. CORVEZ, F. ROUILLIER. *Using computer algebra tools to classify serial manipulators.* F. WINKLER, editor, in « Automated Deduction in Geometry », series Lecture Notes in Artificial Intelligence, Springer, to appear.

[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 », to appear.

[20] G. HANROT, J. RIVAT, G. TENENBAUM, P. ZIMMERMANN. *Density Results on Floating-point Invertible Numbers.* in « Theoretical Computer Science », number 2, volume 291, January, 2003, pages 135-141.

[21] G. HANROT, P. ZIMMERMANN. *A long note on Mulders' short product.* in « Journal of Symbolic Computation », to appear.

[22] D. LAZARD. *Injectivity of real rational mappings: The case of a mixture of two Gaussian laws.* in « Mathematics of Computation », to appear.

[23] V. LEFÈVRE, J.-M. MULLER. *On-the-Fly Range Reduction.* in « Journal of VLSI Signal Processing-Systems for Signal, Image, and Video Technology », number 1-2, volume 33, January, 2003, pages 31-35.

[24] F. ROUILLIER. *Efficient Algorithms based on Critical Points Method.* S. BASU, L. GONZALEZ-VEGA, editors, in « Algorithmic and Quantitative Real Algebraic Geometry », series DIMACS Series in Discrete Mathematics and Theoretical Computer Science, volume 60, American Mathematical Society, 2003, pages 123–138.

[25] F. ROUILLIER, P. ZIMMERMANN. *Efficient Isolation of a Polynomial Real Roots.* in « Journal of Computational and Applied Mathematics », number 1, volume 162, 2003, pages 33-50.

[26] M. SAFEY EL DIN, É. 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 », to appear.

[27] D. WANG. *Automated Generation of Diagrams with Maple and Java.* M. JOSWIG, N. TAKAYAMA, editors, in « Algebra, Geometry, and Software Systems », Springer-Verlag, Berlin Heidelberg, 2003, pages 277–287.

[28] D. WANG. *Implicitization and Offsetting via Regular Systems.* F. CHEN, D. WANG, editors, in « Geometric Computation », World Scientific, Singapore New Jersey, December, 2003, chapter 5, pages 156-176.

Publications in Conferences and Workshops

[29] R. BRENT, P. ZIMMERMANN. *Algorithms for finding almost irreducible and almost primitive trinomials.* in « Primes and Misdemeanours: Lectures in Honour of the Sixtieth Birthday of Hugh Cowie Williams », The Fields Institute, Toronto, A. VAN DER POORTEN, A. STEIN, editors, Banff, Canada, May, 2003, 12 pages. Invited paper. To be published by the AMS..

[30] R. BRENT, P. ZIMMERMANN. *Random number generators with period divisible by a Mersenne prime.* in « Proceedings of Computational Science and its Applications (ICCSA) », series Lecture Notes in Computer Science, number 2667, Springer-Verlag, pages 1–10, 2003, invited paper.

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

[32] D. LAZARD. *Central Configurations of Four Gravitational Masses With an Axis of Symmetry.* in « 8th International Conference on Applications of Computer Algebra (ACA'2002), Volos, Greece », 2002.

[33] V. LEFÈVRE. *Multiplication by an Integer Constant: Lower Bounds on the Code Length.* in « 5th Conference on Real Numbers and Computers 2003 - RNC5, Lyon, France », pages 131-146, September, 2003.

[34] F. ROUILLIER. *Efficient real solutions and robotics.* in « First EMS-SMAI-SMF Joint Conference Applied Mathematics and Applications of Mathematics », 2003.

[35] M. SAFEY EL DIN, É. 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 (ISSAC'2003), Philadelphie, USA », ACM Press, J. SENDRA, editor, pages 224-231, August, 2003.

[36] 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 », pages 142-147, June, 2003.

[37] D. WANG. *GEOTHER 1.1: Handling and Proving Geometric Theorems Automatically.* in « Proceedings of Automated Deduction in Geometry (ADG'2002) », Springer-Verlag, Berlin Heidelberg, F. WINKLER, editor, 2003.

Internal Reports

[38] Y. GÉRARD, I. DEBLED-RENNESSON, P. ZIMMERMANN. *A fast and elementary algorithm for digital plane recognition.* Rapport de recherche, Institut National de Recherche en Informatique et en Automatique, November, 2003.

[39] C. LE GUERNIC. *Résolution de systèmes d'égalités et d'inégalités polynomiales.* Stage ENS, ENS, September, 2003.

[40] D. STEHLÉ. *Breaking Littlewood's cipher.* Rapport de recherche, number 4988, Institut National de Recherche en Informatique et en Automatique, November, 2003, http://www.inria.fr/rrrt/rr-4988.html.

Miscellaneous

[41] D. WANG. *A Simple Method for Implicitizing Rational Curves and Surfaces.* 2003, submitted for publication.

Bibliography in notes

[42] 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.

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

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

[45] P. WENGER. *Some Guidelines for the Kinematic Design of New Manipulators.* in « Mechanism and Machine Theory », volume 35, 2000, pages 437–449.

[46] P. WENGER. *Classification of 3R Positioning Manipulators.* in « Journal of Mechanical Design », volume 120, June, 1998, pages 327–332.