INRIA

INSTITUT NATIONAL DE RECHERCHE EN INFORMATIQUE ET EN AUTOMATIQUE

Action SPACES

Systèmes Polynomiaux, Arithmétiques, Calculs Efficaces et Sûrs

Lorraine

THÈME 2B

Table des matières

1. Composition de l'équipe

Spaces est un projet qui sera vraisemblablement créé au début 2002, bilocalisé entre le LORIA (Nancy) et le LIP6 (Laboratoire d'Informatique de Paris 6). Spaces a été créé comme action en janvier 2001.

Responsable scientifique
Daniel Lazard [professeur, Université Pierre et Marie Curie (Paris 6) (UPMC)]

Responsable permanent
Paul Zimmermann [directeur de recherche, INRIA]

Assistants de projet
David Massot [UPMC, temps partiel]
Geneviève Pierrelée [LORIA, temps partiel 25%, jusqu'à mars 2001]
Franck Girault [LORIA, temps partiel 25%, avril à octobre 2001]
Hélène Zganic [LORIA, temps partiel 17%, depuis octobre 2001]

Personnel CNRS
Jean-Charles Faugère [CR]
Dongming Wang [CR]

Personnel INRIA
Guillaume Hanrot [CR]
Vincent Lefèvre [CR]
Fabrice Rouillier [CR]

Enseignants-Chercheurs
Philippe Aubry [MC UPMC]
Luc Rolland [ATER Université Nancy 1 depuis septembre 2001, doctorant auparavant]
Mohab Safey El Din [MC UPMC depuis août 2001, ATER auparavant]

Chercheurs doctorants
Gwenolé Ars [DGA, co-encadré avec Marie-Françoise Roy, soutenance prévue en 2004]
Magali Bardet [allocataire moniteur, co-encadrée avec Daniel Augot, soutenance prévue en 2004]
Abdolali Basiri [bourse Sfere, soutenance prévue en 2003]
Solen Corvez [BDI CNRS, co-encadrée avec Marie-Françoise Roy, soutenance prévue en 2004]
Nicolas Gurel [LIX, co-encadré avec François Morain, soutenance prévue en 2003]
Philippe Trébuchet [allocataire moniteur, co-encadré avec Bernard Mourrain, soutenance prévue en 2003]

Chercheur post-doctorant
David Daney [Région Lorraine et INRIA, depuis mars 2001]

Chercheurs invités
Richard Brent [août et septembre 2001]
Yves Pétermann [octobre 2001]

Membres extérieurs
François Boulier [MC, Université de Lille 1]
Marie-Françoise Roy [professeur, Université de Rennes 1]

2. Présentation et objectifs généraux du projet

2.1. (Sans titre)

Le thème de notre projet est parfaitement résumé par le développement de l'acronyme Spaces en français et en anglais :

L'objectif principal est la résolution des systèmes d'équations et inégalités polynomiales (**Systèmes Polynomiaux**). Nous privilégions les méthodes algébriques (**Algebraic Computation**), qui sont plus robustes et souvent plus efficaces que les méthodes purement numériques.

En raison de la complexité élevée des problèmes abordés, pour obtenir des logiciels efficaces (**Efficient Software**), il faut associer aux algorithmes théoriques des méthodes d'implantation fines, et en particulier des **Arithmétiques** adaptées et efficaces, notamment les entiers et les rationnels de précision arbitraire, les flottants de grande précision, l'arithmétique d'intervalles, mais aussi les arithmétiques modulaires et p-adiques ou celle des nombres infinitésimaux.

Les systèmes polynomiaux ont des applications dans des domaines variés, aussi bien industriels qu'académiques. Cependant, un travail important est nécessaire pour définir des spécifications de sortie des logiciels qui soient adaptées aux problèmes posés, et aussi pour transcrire les problèmes du langage des applications vers une forme adaptée à la résolution. Ainsi, la résolution de problèmes (**Solving Problems**) est une part essentielle de notre activité de développement logiciel.

La variété même de ces applications rend nécessaire que les logiciels soient robustes (**Calculs Efficaces et Sûrs**) : d'une part, l'instabilité numérique est presque la règle dans les problèmes abordés, et une garantie sur les erreurs d'arrondis est indispensable ; d'autre part, il est certain que les hypothèses simplificatrices courantes nécessitées par certains algorithmes, telle la régularité, ne sont pas toujours satisfaites, et doivent donc pouvoir être évitées ou à tout le moins testées.

2.2. Aperçu historique

La résolution des équations et systèmes polynomiaux constitue depuis longtemps un problème fondamental. En sont témoins, par exemple, les efforts consacrés à la résolution par radicaux, jusqu'à ce que Galois résolve le problème vers 1830, mais aussi le fait que le théorème de d'Alembert, qui affirme l'existence de solutions complexes, soit appelé « **fundamental theorem of algebra** » par les anglo-saxons.

Ce n'est qu'à la fin du XIX^{e} siècle que sont apparus les premiers algorithmes généraux pour résoudre les systèmes polynomiaux, dans les travaux de Bézout, Sylvester, Kronecker et surtout McCaulay. Mais la complexité du problème rendait les calculs totalement impraticables, ce qui a conduit les mathématiciens à abandonner cette approche effective pour mettre l'accent sur des résultats qualitatifs et non effectifs. C'est ainsi qu'est apparue la géométrie algébrique moderne, avec son niveau d'abstraction qui la rend souvent presque ésotérique. Cette évolution est illustrée par le célèbre traité d'algèbre de van der Waerden [dW55] dans lequel les méthodes effectives ont été supprimées à partir de la quatrième édition ; en particulier le chapitre sur la théorie de l'élimination, qui traitait de la résolution effective des systèmes polynomiaux, a totalement disparu.

Avec l'invention des ordinateurs et l'apparition des systèmes de calcul formel, le problème de la résolution effective des systèmes polynomiaux est redevenu d'actualité, dès que, dans les années 70, les calculs de PGCD et de factorisation des polynômes ont reçu des solutions satisfaisantes. Mais la complexité et la difficulté du problème font qu'il a fallu attendre la deuxième moitié des années 80 pour commencer à pouvoir résoudre par logiciel des problèmes inaccessibles au calcul manuel.

2.3. Systèmes de dimension zéro

Stricto sensu, un système d'équations polynomiales est une formule

$$P_1 = 0 \text{ et } P_2 = 0 \text{ et } \ldots \text{ et } P_k = 0, \tag{1}$$

où les P_i sont des polynômes à plusieurs variables et à coefficients dans un corps K. Un tel système est généralement représenté par l'ensemble des polynômes P_i. Résoudre un tel système consiste à déterminer les valeurs des variables qui satisfont cette formule. Ces valeurs sont recherchées dans un corps algébriquement clos contenant K, ou dans le corps des réels si K est un corps réel[1].

Un système est dit « de dimension zéro » si l'ensemble de ses solutions dans un corps algébriquement clos est fini. Dans ce cas, les solutions ne dépendent pas du corps algébriquement clos choisi. C'est la seule situation la plus favorable au calcul purement numérique. Encore faut-il que le nombre d'équations soit égal

[1] Si $\mathbb{Z}$ est l'anneau des nombres entiers, l'existence de solutions dans $\mathbb{Z}$ est un problème indécidable.

au nombre de variables, et même dans ce cas les performances des solveurs numériques sont le plus souvent médiocres en raison des instabilités numériques.

Les techniques algébriques subdivisent le problème de la résolution en deux étapes : la première consiste à transformer le système en un ou plusieurs systèmes équivalents mais mieux adaptés et qui constituent ce que l'on peut appeler une **solution algébrique**. La deuxième étape consiste, dans le cas où K est un sous-corps des complexes, à calculer les valeurs numériques des solutions à partir de la solution algébrique.

Le calcul de la solution algébrique se décompose elle-même généralement en plusieurs étapes. La première est le plus souvent le calcul d'une base de Gröbner ; aussi de nombreux chercheurs ont assimilé, pendant plusieurs années, le calcul des bases de Gröbner à la résolution des systèmes polynomiaux.

La solution algébrique peut prendre diverses formes ayant chacune ses avantages et ses inconvénients. La forme la mieux adaptée au calcul des solutions numériques est la RUR (représentation univariée rationnelle - [Rou99]) qui consiste en une équation en une variable $f(t) = 0$ (où t est souvent une variable auxiliaire) et en l'expression des autres variables comme fractions rationnelles en t (quotients de deux polynômes).

À partir de la RUR, le calcul des valeurs numériques des solutions revient à calculer les racines d'un polynôme univarié. Ce n'est toutefois pas aussi simple qu'il n'y paraît car c'est généralement un polynôme de degré élevé ayant de très grands coefficients. En outre, dans le cas très fréquent où l'on s'intéresse aux solutions réelles satisfaisant certaines conditions de signe, il faut garantir une précision suffisante pour pouvoir déterminer le signe d'un certain nombre de polynômes en t.

Les algorithmes et les implantations développés par J.-C. Faugère et F. Rouillier sont très largement les plus performants existant actuellement. Ils permettent couramment de résoudre des systèmes ayant de l'ordre d'un millier de solutions. Néanmoins, un travail important est encore nécessaire pour optimiser encore ces algorithmes, mais surtout, à très court terme, pour les rendre vraiment accessibles à la communauté sous forme de logiciels robustes, utilisables par des non spécialistes.

Il faut noter qu'en raison de la taille des systèmes dont la résolution est maintenant accessible, le facteur limitant provient souvent de l'efficacité de l'arithmétique utilisée.

2.4. Systèmes de dimension positive

Pour les systèmes ayant une infinité de solutions, on ne peut pas se limiter aux systèmes de la forme (1) : il faut souvent considérer des inéquations ($P_i \neq 0$) et, dans le cas réel, des inégalités ($P_i > 0$, ou encore $P_i \geq 0$)[2]. En outre, beaucoup de problèmes ne se limitent pas à une conjonction d'équations, inéquations et inégalités, mais se présentent sous la forme d'une expression avec quantificateurs (formule de la logique du premier ordre dont les objets atomiques sont des équations, inéquations ou inégalités).

Pour de tels systèmes généraux, la résolution se décompose a priori en trois étapes : en premier lieu, on connaît des algorithmes permettant d'associer à un tel système général un système sans quantificateur (formule du calcul propositionnel) ayant les mêmes solutions dans un corps algébriquement clos ou, s'il y a des inégalités, dans le corps des réels. Étant donné un tel système sans quantificateur, on connaît également des algorithmes permettant de décomposer l'ensemble des solutions en composantes irréductibles et/ou en composantes connexes (définies par de nouveaux systèmes). Enfin, étant donnée une telle composante, se pose le problème de l'étudier (la dessiner, déterminer ses singularités, sa topologie, etc.).

Un tel programme se heurte à des difficultés qui sont très loin d'être résolues. En premier lieu, les algorithmes résolvant les deux premières étapes ont une complexité élevée, généralement exponentielle en le nombre de variables et doublement exponentielle en le nombre d'alternances de quantificateurs. Cette complexité est asymptotiquement optimale, à une constante près **située en exposant**. Mais ces algorithmes ont une efficacité pratique si mauvaise qu'il n'y a aucun intérêt à essayer de les implanter.

Maintenant que nous disposons d'algorithmes efficaces en dimension zéro, un des objectifs principaux du projet est de mettre au point et d'implanter des algorithmes ayant une efficacité pratique acceptable en dimension positive, quitte à restreindre la classe des problèmes considérés.

[2] Il en est de même en dimension zéro, mais, dans ce cas, les inéquations et inégalités peuvent n'être considérées qu'à la fin du calcul, pour éliminer certaines solutions.

Une autre catégorie de difficultés vient de ce que l'« étude » mentionnée plus haut n'a pas de spécification générale. Il y a donc un vaste travail nécessaire pour déterminer quelles spécifications de sortie sont utiles pour les applications tout en étant accessibles au calcul.

Comme exemples de telles spécifications partielles, on peut citer deux travaux récents dans le projet : un algorithme efficace pour produire au moins un point par composante connexe de l'ensemble des solutions réelles d'un système de la forme (1), et un algorithme pour discuter le nombre de solutions réelles satisfaisant certaines inégalités pour un système de dimension zéro dépendant de un ou deux paramètres. Ce dernier algorithme a été développé dans le cadre de la résolution d'un problème de mécanique céleste.

2.5. Calculs efficaces et arithmétique

Pour résoudre un système polynomial, il est vain d'utiliser un algorithme de bonne complexité arithmétique, c'est-à-dire effectuant peu d'opérations sur les coefficients des polynômes, si les opérations sur ces coefficients sont elles-mêmes lentes. En effet, ces coefficients peuvent couramment atteindre des tailles de l'ordre du millier de chiffres. Il est donc primordial pour obtenir des logiciels efficaces (**Efficient Software**), ce qui constitue notre but final, de diminuer tous les facteurs intervenant dans le coût réel des algorithmes (**bit complexity**). En langage courant, on dirait « tous les coups sont permis ». On peut distinguer **grosso modo** deux classes d'améliorations possibles : celles concernant les coefficients des polynômes considérés, et toutes les autres (protocoles, ramasse-miette, ...). Les algorithmes rapides de manipulation de polynômes ont pour leur part déjà été considérés dans la section 2.3.

D'autre part, les algorithmes de calcul formel nécessitent des développements spécifiques pour leur implantation. En effet, une des particularités du calcul exact est de manipuler un nombre important d'objets de grande taille variant fortement et de façon souvent mal contrôlée dans le temps. Il en découle toute une série de problèmes informatiques délicats (gestionnaire de mémoire, échanges de données, etc.) ne trouvant pas de solution satisfaisante parmi les outils standards.

2.5.1. Arithmétiques

On s'intéresse ici aux algorithmes et méthodes permettant de rendre plus efficaces les calculs sur les coefficients des polynômes du système à résoudre. Les coefficients de départ sont le plus souvent des entiers ou rationnels ($\mathbb{Z}$ ou $\mathbb{Q}$) ou des entiers modulaires ($\mathbb{F}_p$). Cependant, plusieurs autres arithmétiques auxiliaires peuvent être utilisées efficacement : les nombres flottants à précision fixée, tout en bornant la taille des calculs, permettent dans nombre de cas d'obtenir des résultats suffisants ; les nombres p-adiques sont souvent un intermédiaire très utile entre nombres entiers et modulaires (par exemple dans l'algorithme de factorisation de polynômes de Berlekamp-Zassenhaus) ; les nombres algébriques sont indispensables pour la résolution de certains systèmes avec paramètres ; enfin les infinitésimaux permettent l'étude de systèmes légèrement « déformés ».

On peut répartir ces arithmétiques en deux classes : les arithmétiques « exactes » (entiers, nombres algébriques) et les arithmétiques « approchées » (flottants, nombres p-adiques, infinitésimaux). La plupart des algorithmes sur les polynômes (resp. séries) s'appliquent aux objets de la première (resp. deuxième) classe. Si les meilleurs algorithmes asymptotiques demeurent les mêmes depuis longtemps pour les arithmétiques exactes, curieusement il n'en est pas de même pour les arithmétiques approchées. Citons pour preuve les travaux récents de Mulders [Mul00] qui a montré qu'un gain d'environ 20% pouvait être obtenu sur un produit flottant sur n bits par rapport au produit entier[3], lorsque ce dernier est calculé via l'algorithme en $O(n^{1.59})$ de Karatsuba.

2.5.2. Infrastructures logicielles

2.5.2.1. L'environnement de programmation.

En calcul formel, l'utilisateur est, pour l'heure, obligé de jongler avec différents logiciels. Les logiciels généraux tels Maple, MuPAD ou Mathematica possèdent de nombreuses fonctionnalités mais sont très limités

[3] Rappelons que le produit flottant sur n bits est le produit d'entiers tronqué aux n bits de poids fort.

lorsqu'il s'agit de faire du calcul intensif. D'un autre côté, il y a des logiciels spécialisés et langages de base (C/C++), rustiques d'utilisation mais permettant de faire certaines parties de calcul beaucoup plus efficacement. Pour utiliser de façon optimale les ressources dont on dispose, il est essentiel de recourir à un environnement de développement efficace. La diversité des structures de données utilisées en calcul formel rend impérative la définition de protocoles d'échanges assez complexes mais surtout souples et efficaces comme le protocole UDX (**Universal Data Exchange**) développé par les membres du projet et objet d'un dépôt de brevet.

2.5.2.2. La gestion de mémoire.

L'implantation des algorithmes est une tâche primordiale à laquelle les membres du projet prennent une part active : en effet une mauvaise implantation peut rendre totalement inefficace un algorithme potentiellement bon. Un objectif du projet est d'implanter dans des langages de bas niveau (C/C++) toutes les parties critiques des nouveaux algorithmes. Il est à noter que la vitesse d'exécution d'un algorithme n'est pas le seul critère à prendre en considération ; bien souvent c'est la taille des données en mémoire qui limite l'utilisation d'un programme de calcul formel. Les membres du projet ont développé un algorithme de gestion de la mémoire original particulièrement adapté à ce type de données (des données dont la taille varie beaucoup en cours de calcul) et facile à utiliser dans un langage de bas niveau. Il faut que ce mécanisme de gestion de la mémoire reste efficace même lorsqu'on utilise plus de mémoire que la mémoire physiquement disponible (**swap**) ; de plus ce mécanisme est bien adapté au calcul distribué.

2.6. Applications

Les applications sont fondamentales pour l'activité du projet pour plusieurs raisons, et, de ce fait, font pleinement partie de notre thématique de recherche.

La première de ces raisons est que les applications constituent des tests indispensables pour vérifier et valider l'efficacité de nos algorithmes. En effet, la complexité de la résolution d'un système polynomial dépend très irrégulièrement du problème posé, et la difficulté d'un problème tiré au hasard ne donne aucune indication sur celle d'un problème réel de taille comparable.

Une deuxième raison a été mentionnée plus haut : les applications sont nécessaires pour déterminer quels algorithmes sont à développer ou à optimiser en priorité. Inversement, c'est souvent en essayant de comprendre pourquoi une application particulière résistait à nos logiciels que de nouveaux algorithmes, beaucoup plus efficaces, ont été découverts.

Enfin, la résolution de problèmes inaccessibles aux autres approches existantes est la meilleure justification de l'importance de la résolution des systèmes polynomiaux et de la qualité de nos travaux.

Mais il y a là une difficulté spécifique : les problèmes susceptibles d'être résolus avec nos méthodes peuvent être modélisés de manières très variées. Avec l'idée reçue que les systèmes polynomiaux sont numériquement insolubles, la plupart des chercheurs et ingénieurs présentent leur problèmes sous une forme telle, qu'il n'est pas immédiat qu'il s'agisse de systèmes polynomiaux. Même quand c'est apparent, ces systèmes sont présentés sous une forme « simplifiée » voire linéarisée qui augmente la difficulté du problème au point de le rendre insoluble.

Ainsi, il n'est pas possible d'utiliser nos logiciels comme une boîte noire, et il faut souvent remonter aux sources des problèmes (et donc comprendre le langage lié à l'application), afin de trouver une formulation susceptible d'être résolue.

Ceci aboutit à ce qu'une part notable de nos publications relève de domaines scientifiques extérieurs au nôtre.

Ainsi, nous avons déjà publié ou avons des articles en préparation en robotique (robots parallèles), traitement du signal (brevet en compression d'image), mécanique céleste (configurations stables de n corps) et biophysique (forme de molécules minimisant l'énergie). Il faut aussi mentionner les applications à la cryptologie, qui deviennent une part notable de notre activité, avec quatre thèses en préparation sur le sujet.

3. Fondements scientifiques

3.1. Résolution algébrique

Le problème de la résolution des systèmes polynomiaux (non linéaires) ne représente qu'une partie des questions abordées par le calcul formel ; cependant ce problème fondamental pourrait bien être le meilleur exemple d'application où le calcul formel apporte le plus par rapport aux méthodes numériques. En ce domaine les avantages du calcul formel sont nombreux : exactitude des résultats puisque tous les calculs intermédiaires sont exacts; garantie de trouver **toutes** les solutions (même si le nombre de solutions est infini) ; efficacité des algorithmes et des implantations de ces méthodes y compris pour des problèmes provenant d'applications industrielles.

Les membres du projet et les logiciels qu'ils ont écrits (FGb/RS représentant environ 200 000 lignes de code C/C++) sont déjà reconnus comme les leaders mondiaux pour le calcul des racines réelles et complexes des systèmes d'équations algébriques dans le cas particulier (important) où le nombre de solutions est fini. Ces logiciels ont déjà permis de résoudre plusieurs applications provenant du milieu académique ou industriel (traitement du signal, robotique, mécanique céleste, biophysique, cryptologie, codes correcteurs).

Le cas particulier de la résolution des systèmes algébriques où le nombre de solutions est fini est maintenant bien maîtrisé en théorie et en pratique et ceci en grande partie grâce aux travaux des membres du projet. Dans ce cas particulier, l'outil de base pour résoudre les systèmes est le calcul de bases de Gröbner. Plusieurs algorithmes parmi les plus efficaces ont été proposés par J.-C. Faugère et sont maintenant des algorithmes standards implantés dans tous les systèmes de calcul formel (Maple, Mathematica, Axiom, Magma, ...). Cette étape permet de calculer toutes les racines complexes du problème. Dans la pratique les solutions cherchées vérifient d'autres contraintes : les racines doivent être réelles, réelles positives ou vérifier un certain nombre d'inégalités. Pour extraire ces racines réelles, la méthode de [Rou99] consiste à ramener l'étude d'un problème à plusieurs variables à l'étude d'**un** polynôme en une variable, en transformant le résultat de l'étape précédente (base de Gröbner) en une liste de fractions rationnelles en une variable, exprimant les solutions en fonction des racines d'un polynôme (forme RUR). Des algorithmes [RZ01] permettent ensuite de calculer très rapidement les racines réelles d'un polynôme univarié de façon exacte et avec une précision aussi grande que voulue. Tous ces algorithmes sont implantés dans les logiciels FGb et RS et constituent un solveur complet permettant de résoudre des problèmes ayant plusieurs milliers de solutions. Ces logiciels sont de plusieurs ordres de grandeur plus efficaces ou fiables que leurs concurrents.

Notre but est donc de faire avancer significativement en théorie et en pratique le problème de la résolution des systèmes d'équations algébriques **avec paramètres**. Lorsque le nombre de solutions est infini, les systèmes d'équations en nombres réels constituent des problèmes très mal résolus, en raison à la fois de la complexité sur-exponentielle des problèmes et de l'inefficacité des algorithmes actuellement implantés. Pourtant les applications potentielles de ces méthodes sont nombreuses : c'est la raison pour laquelle ce problème constitue un défi actuel du calcul formel.

Certains algorithmes pour calculer les racines réelles ont une très bonne complexité théorique mais sont inefficaces en pratique et très difficiles à implanter en machine. De nouveaux algorithmes (dont certains développés par des membres du projet) radicalement différents, visant à réduire le coût des calculs par rapport à la CAD[Col75] (**Cylindric Algebraic Decomposition**) utilisent fortement la possibilité de décomposer les idéaux (en particulier la décomposition en idéaux premiers) : plusieurs membres du projet ont proposé des algorithmes efficaces pour le calcul de ces décompositions. Un objectif du projet est d'implanter dans des langages de bas niveau (C/C++) et de façon très efficace (vitesse et gestion mémoire) ce type d'algorithmes sophistiqués et d'en faire une évaluation précise.

La spécification mathématique du résultat d'un calcul lorsque le nombre de solutions est infini est déjà un problème difficile [Aub99][ALM99][Laz01]. Dans ce cas, plusieurs expressions formelles des solutions sont envisageables mais il apparaît que les outils algébriques nécessaires pour calculer les zéros réels ou complexes puissent se résumer à :

(Z) la résolution des systèmes sans paramètre (problème bien maîtrisé) ;

(Rad) le calcul d'un système de générateurs du radical d'un idéal ;
(Dec) la décomposition équidimensionnelle d'un idéal (c'est-à-dire séparer les points isolés des courbes et des surfaces dans l'ensemble des solutions) ;
(Prim) la décomposition en idéaux premiers d'un idéal, c'est-à-dire l'action de (Dec) plus séparer entre eux les points isolés, séparer entre elles les courbes, ...

À noter que le radical d'un idéal se déduit facilement d'une décomposition équidimensionnelle qui à son tour dérive très facilement d'une décomposition en premiers. On pourrait donc s'attendre à ce que le calcul du seul radical soit moins coûteux que la décomposition en idéaux premiers, pourtant l'expérience montre que l'algorithme F_7 [Fau01] réalise ces tâches de façon efficace et à peu près dans le même temps. De plus, même si le temps pour calculer la décomposition en premiers est légèrement supérieur au calcul du radical, comme le calcul a permis de casser le problème initial en sous-problèmes indépendants de plus petites tailles, le calcul de toutes les racines peut se poursuivre sur chacune des composantes (éventuellement en parallèle).

L'outil des bases de Gröbner, qui était le coeur de l'algorithmique pour les systèmes sans paramètre, n'est plus adapté pour répondre calculer des décompositions puisqu'une base de Gröbner conserve toute l'information qui était présente initialement (multiplicités, ...). De plus le nombre d'éléments dans une base de Gröbner est généralement exponentiel en le nombre de variables. Deux voies sont explorées par les membres de l'équipe pour contourner les bases de Gröbner.

Plusieurs membres du projet ont proposé des algorithmes pour calculer une décomposition équidimensionnelle : ces algorithmes sont basés sur les ensembles triangulaires. Ces méthodes sont potentiellement meilleures que les bases de Gröbner puisqu'elles se placent d'emblée dans le radical de l'idéal (méthode ensembliste) et parce qu'elles effectuent des scindages dès le début de l'algorithme. Un autre avantage des ensembles triangulaires est que le nombre de générateurs est borné par le nombre de variables. L'étude des idéaux représentés par un ensemble triangulaire nous a permis de dégager des propriétés plus fines, trouvant des applications dans différents domaines. Nous avons ainsi nettement amélioré l'efficacité d'algorithmes que nous avons proposés en géométrie réelle effective. Notons que les ensembles triangulaires sont un outil essentiel en algèbre différentielle et que le travail mené sur ce sujet par les membres du projet bénéficie directement à la résolution des systèmes algébriques différentiels. Par ailleurs, les décompositions triangulaires sont bien adaptées au problèmes de raisonnement automatique en géométrie où on a besoin de considérer les variétés plutôt que les idéaux et de prendre en compte les cas dégénérescents qui sont en général représentés séparément du cas générique dans les sorties obtenues. Des progrès spectaculaires ont déjà été accomplis dans le domaine de l'efficacité de ces méthodes. Toutefois il reste encore un travail important d'optimisation et d'implantation fine de ces méthodes pour traiter les problèmes industriels.

Faugère a proposé un nouvel algorithme (F_7) qui calcule une décomposition en idéaux premiers, l'algorithme F_7 étant lui-même basé sur l'algorithme F_5. L'algorithme F_5 est un algorithme récent et très efficace qui permet de calculer des bases de Gröbner sans calculs intermédiaires inutiles en utilisant de façon intensive l'algèbre linéaire creuse. L'algorithme F_7 essaye de calculer une base de Gröbner traditionnelle mais tente de détecter les matrices singulières qui apparaissent dans F_5 ; lorsque c'est le cas un scindage est effectué permettant de casser prématurément le problème. L'algorithme F_7 utilise également dans sa phase terminale les travaux récents sur la factorisation rapide des polynômes [ASZ00]. Un premier prototype d'implantation de cet algorithme a été réalisé et montre que bien souvent non seulement le calcul est plus rapide qu'un simple calcul de base de Gröbner mais, de plus, la qualité du résultat est fortement améliorée : le résultat est unique et mathématiquement bien spécifié ; la taille du résultat plus petite (parfois d'un facteur 1000), pour chaque composante la structure du résultat est plus simple, pour chaque composante on peut aussi extraire un ensemble triangulaire [ALM99] qui est un système de générateurs particulièrement simple (en particulier le nombre d'éléments est inférieur au nombre de variables).

Toutefois la meilleure façon de représenter mathématiquement et informatiquement une composante première est encore un problème ouvert : nous comptons sur le retour d'expériences provenant des applications afin de déterminer la meilleure façon d'exprimer les solutions en fonction des besoins de l'algorithme sur les zéros réels.

Le couplage des algorithmes décrits dans ce paragraphe et des algorithmes issus de la partie réelle constituera, à terme, le noyau informatique d'un solveur complet pour les systèmes paramétrés.

3.2. Solutions réelles

Mots clés : *zéros réels, isolation, systèmes algébriques, ensembles semi-algébriques.*

Glossaire

CAD Cylindrical Algebraic Decomposition

L'étude des zéros réels des systèmes algébriques est un sujet plutôt mal cerné d'un point de vue algorithmique. Si l'on sait désormais résoudre correctement les systèmes de dimension zéro (**i.e.**, admettant un nombre fini de solutions complexes), il n'en est pas de même pour les systèmes de dimension positive.

En géométrie réelle effective, l'objectif à long terme de beaucoup d'équipes est la résolution du problème général d'élimination des quantificateurs. La résolution de ce problème en géométrie réelle s'appuie principalement, d'un point de vue algorithmique, sur la résolution de systèmes généraux d'égalités et d'inégalités polynomiales (par résolution, on entend décider du vide et/ou fournir un point par composante connexe).

L'état de l'art sur l'implantation de méthodes résolvant ce problème est facile à établir.

D'un côté, nous avons l'algorithme de décomposition cylindrique algébrique (CAD)[Col75] qui est la seule méthode viable en pratique mais dont la complexité est désastreuse tant en ce qui concerne les calculs qu'en ce qui concerne la taille des résultats (beaucoup trop de points calculés, codage complexe des résultats intermédiaires : polynômes à coefficients dans des tours d'extensions). Cette méthode est basée sur un traitement récursif en le nombre de variables des polynômes considérés.

D'un autre côté, nous avons un certain nombre de méthodes théoriques de bonne complexité (au moins en ce qui concerne la taille de la sortie) essentiellement basées sur l'étude des points critiques de fonctions bien choisies (projection selon un axe, fonction distance à un point etc.). Grigoriev et Vorobjov puis différents auteurs comme en particulier Bashu, Pollak et Roy ou encore Traverso travaillent sur des méthodes basées sur l'étude des points critiques de fonctions bien choisies. Pour illustrer la philosophie générale, on sait par exemple que l'ensemble des points critiques de la fonction de projection par rapport à une coordonnée intersecte chaque composante connexe de la variété étudiée lorsqu'elle est lisse et bornée. De plus, on sait facilement modéliser ces points lorsque la variété est définie par une unique équation.

Deux avantages importants sont à mettre au crédit des méthodes basées sur ce type de résultats :

- on se ramène en une étape (et non récursivement) à la résolution de problèmes *simples* (résolution de systèmes zéro-dimensionnels),
- la taille de la sortie est bien contrôlée.

Les différents auteurs ramènent le cas général des hypersurfaces définies par une unique équation à celui d'une hypersurface lisse et bornée, au moyen de déformations infinitésimales (une ou deux selon les versions).

Pour généraliser ensuite ces techniques au cas des systèmes d'équations quelconques, des méthodes standards sont employées (prendre la somme des carrés des équations par exemple) et le passage au cas de systèmes d'inégalités se fait par une nouvelle série de déformations et/ou de mises en position générale.

Un rapport détaillé dû à Hoon-Hong montre que l'implantation naïve de ces **méthodes théoriquement bonnes** est elle aussi désastreuse (en estimant simplement le temps d'affichage de certains résultats intermédiaires, il constate que sur des problèmes où la CAD prend moins d'une seconde, il faudrait plusieurs milliers d'années aux algorithmes dits théoriquement bons pour effectuer le calcul). Les mêmes observations peuvent être faites sur les propositions d'algorithmes les plus récentes.

En analysant la situation précisément, on peut voir que les méthodes **de bonne complexité théorique** proposées souffrent d'une mauvaise conception au sens informatique du terme. En effet, la façon dont elles ont été pensées est le reflet d'une démarche fréquente en mathématiques : on résout le problème dans une situation particulière où l'on peut soigneusement éviter tous les ennuis (par exemple en supposant les variétés

lisses ou bornées). On se ramène ultérieurement à cette situation par une suite de transformations ayant peu d'influence sur la complexité théorique, mais catastrophiques en pratique : sommes de carrés (l'élévation au carré ne multipliant les degrés que par un facteur $O(1)$), déformations infinitésimales (cela ne change que le coût effectif des opérations de base bien souvent ignoré dans les calculs de complexité théorique), utilisation de changements de variables dits **génériques** qui tuent en général la structure des équations initiales si bien que l'on se retrouve systématiquement dans le **pire des cas**.

Notre objectif principal à court et moyen terme est de fournir des algorithmes efficaces pour le calcul d'au moins un point par composante connexe pour les variétés algébriques réelles. Ce type d'outil permettra alors de résoudre ponctuellement bon nombre de problèmes et, à plus long terme, servira de base pour la résolution des systèmes d'égalités puis, plus généralement, pour l'élaboration de méthodes générales pour la résolution du problème d'élimination des quantificateurs.

Notre idée première est de préserver la philosophie des méthodes théoriquement bonnes (étude de points critiques de fonctions bien choisies), assurant une bonne complexité pour la taille de la sortie, notre but étant d'en rendre possible le calcul exact.

Notre point de vue est d'étudier les variétés dans leur globalité et non par projections successives pour limiter dans un premier temps la taille de la sortie (beaucoup de discussions de cas inutiles sont introduites par l'utilisation de projections successives). L'idée est de se ramener autant que faire se peut à la résolution de problèmes **simples** comme la résolution de systèmes zéro-dimensionnels. De nombreuses études ont déjà été effectuées dans ce sens et quelques résultats théoriques intéressants sont disponibles.

3.3. Arithmétiques

Dans la plupart des arithmétiques étudiées, l'essentiel des problèmes se rencontre lors de l'étude des algorithmes de multiplication, division et racine carrée, ces deux dernières opérations pouvant elles-mêmes se ramener à la multiplication. Dans le cas des arithmétiques exactes, selon la taille des objets manipulés (notée n par la suite), on utilise généralement soit une méthode naïve de complexité $O(n^2)$, soit une méthode de Karatsuba de complexité $O(n^{1.59})$, pour laquelle les variantes de type Toom-Cook donnent une complexité $O(n^{\log(2r+1)/\log(r+1)})$, soit des méthodes basées sur la transformée de Fourier rapide de complexité $O_\varepsilon(n^{1+\varepsilon})$. Le cas des arithmétiques approchées est similaire, excepté que la prise en compte des spécificités du problème conduit usuellement à de meilleures constantes multiplicatives.

Si relativement peu de résultats nouveaux ont été obtenus ces dix dernières années pour les calculs entiers[4], plusieurs algorithmes originaux ont été inventés pour les calculs flottants (astuce de Karp et Markstein pour la division et la racine carrée via l'itération de Newton en 1997 [ePM97] produit et division « courts » de Mulders[Mul00] et algorithme RMP de Hanrot-Quercia-Zimmermann en 2000 [HQZ00]), ce qui laisse penser que de nombreux autres résultats sont à attendre.

Le domaine d'application de ces algorithmes est très vaste ; par exemple, l'itération de Newton est utilisée jusque dans les microprocesseurs actuels pour effectuer les divisions et racines carrées, même en simple précision !

Sur le plan des implantations, d'excellents outils d'arithmétique exacte sont largement répandus. Citons par exemple la bibliothèque GNU MP, développée par T. Granlund, qui semble à l'heure actuelle offrir les meilleures performances sur une large palette d'architectures courantes.

La situation des arithmétiques approchées est bien plus complexe. Le premier problème qui se pose pour l'utilisateur (mais auquel le développeur n'a hélas pas toujours pensé) est celui de la sémantique des opérations (une définition précise de cette sémantique est indispensable pour garantir un comportement déterministe des programmes). Prenons par exemple le cas de la multiplication de deux nombres flottants de n bits ; seuls les n premiers bits du résultat auront un sens, et les méthodes de calcul tenteront souvent d'exploiter le fait que seuls n bits seront renvoyés ; toutefois, certaines bibliothèques ignorent par là-même complètement toute retenue pouvant surgir de la partie « non significative », entachant ainsi d'erreur les derniers bits du résultat retourné.

[4] Citons cependant l'algorithme de division rapide à la Karatsuba inventé par Jebelean et clarifié par Burnikel et Ziegler (rapport 98-1-022, MPI Saarbrücken, 1998).

La définition d'une sémantique précise est particulièrement critique dans le cas d'algorithmes hybrides, où l'utilisation d'algorithmes approchés doit quand même rester dans un cadre où la rigueur est suffisante pour s'apercevoir le cas échéant que les calculs ont perdu toute signification.

Historiquement, ce problème a déjà été rencontré pour les nombres flottants double précision, et a donné lieu, à l'issue de longues tergiversations[5], à la définition de la norme IEEE-754.

Plus récemment, on a pu voir fleurir diverses tentatives de généralisations partielles de la norme IEEE-754 en précision arbitraire, avec des implantations associées. Aucune n'a à ce jour réussi à s'imposer comme standard. On peut citer Arithmetic Explorer (Université d'Anvers), Mathematica,... Les membres de Spaces ont eux-mêmes implanté une généralisation de la norme IEEE-754, au-dessus de la bibliothèque GMP précédemment mentionnée. Cette bibliothèque, baptisée MPFR (pour **Multiple Precision Floating-Point Reliable**) est distribuée avec GMP depuis la version 3.1 d'août 2000.

L'arithmétique des corps premiers finis a été bien étudiée, et on sait bien comment l'optimiser en fonction de la caractéristique, en utilisant selon la taille du corps, des techniques spécifiques ou l'arithmétique des entiers. Mais le problème est plus compliqué pour les extensions de corps, qui non seulement dépendent de deux paramètres (la caractéristique et le degré de l'extension), mais dépendent aussi du choix d'un générateur, qui peut avoir une grande influence sur la complexité arithmétique. Il en résulte qu'il y a de nombreuses situations (notamment les corps de caractéristique 2 et de taille moyenne qui interviennent en cryptologie) où l'on ne sait pas optimiser correctement l'arithmétique.

Nos objectifs incluent, le lecteur l'aura compris, l'étude (de la conception à l'implantation **efficace**) d'algorithmes pour diverses arithmétiques exactes ou approchées. Ces algorithmes seront, bien entendu, pensés en étroite relation avec leur cadre d'utilisation, à savoir certaines des méthodes hybrides mentionnées dans la section 3.4.

L'arithmétique peut aussi offrir des problèmes d'intérêt intrinsèque. Par exemple, autour de V. Lefèvre, nous visons, en collaboration avec le projet Arénaire (INRIA Rhône-Alpes et ENS Lyon) la détermination efficace des pires cas des fonctions élémentaires. V. Lefèvre a mis au point dans sa thèse [Lef00] un algorithme performant à base d'approximations polynomiales hiérarchiques, mais qui nécessite de « descendre » jusqu'à une approximation de degré 1, dont la plage de validité est limitée à de l'ordre de 2^{16} points, ce qui rend la double précision (2^{53} points) faisable mais difficile, et la quadruple précision (2^{113} points) impossible.

Or des travaux récents de Noam Elkies présentés à la conférence ANTS IV semblent indiquer que des méthodes basées sur l'algorithme LLL pourraient autoriser l'utilisation d'approximations de degré 2 voire plus. Si cela est possible efficacement, cela permettrait de traiter la double précision de façon routinière, et pourquoi pas d'approcher la quadruple précision. Notons par ailleurs que l'algorithme LLL est un exemple typique où les méthodes hybrides permettent d'améliorer grandement les performances, même si les problèmes de stabilité numérique ne sont peut-être pas encore tous compris. Cela pourrait donner lieu à une étude approfondie.

À l'issue de l'étude des pires cas pour les fonctions élémentaires, on pourrait envisager — toujours en collaboration avec le projet Arénaire — la réalisation d'une bibliothèque garantissant l'arrondi exact sur les fonctions élémentaires ($\exp$, $\log$, $\sin$, $\cos$, ...).

Par ailleurs, avec le développement de nos applications en cryptologie, il faudra envisager aussi la réalisation de bibliothèques pour l'arithmétique des différents corps finis intervenant dans ce domaine.

3.4. Méthodes hybrides

Ces dernières années ont vu l'émergence de nombreux algorithmes basés sur des arithmétiques « hybrides ». Une arithmétique hybride consiste en l'utilisation conjointe de plusieurs arithmétiques de base comme celles mentionnées dans la section 2.5.1. Par exemple, on remplace dans un algorithme les calculs entiers par des calculs flottants, tout en pouvant décider de la plupart des signes des expressions rencontrées — à condition que l'arithmétique flottante ait une sémantique précise — et on n'effectue les calculs entiers que dans les rares cas où l'arithmétique flottante échoue. On parle alors d'algorithme hybride symbolique-numérique. Ces algorithmes, lorsqu'ils sont bien conçus, permettent de combiner les avantages d'une arithmétique exacte

[5] L'histoire de cette norme apparaît sur http://www.cs.berkeley.edu/~wkahan/ieee754status/754story.html.

(permettant par exemple de tester l'égalité à zéro du résultat) et d'une arithmétique approchée (principalement la rapidité).

Les algorithmes hybrides constituent un sujet de recherche particulièrement actif ; notons en particulier que le comité d'organisation permanent de la conférence ISSAC a décidé d'encourager les recherches autour de ce type d'algorithmes, avec la création dès 2001 de divers prix pour des travaux dans ce domaine.

On peut également utiliser des arithmétiques hybrides lors d'une pré-exécution pour produire des « traces de calcul ». Ainsi un premier calcul de base de Gröbner avec des coefficients flottants va produire une « trace » des paires critiques ayant conduit ou non à des polynômes nuls. Cette trace est ensuite réutilisée pour « guider » les calculs — beaucoup plus coûteux — avec des coefficients entiers, et donc éliminer la plupart des calculs « inutiles ». Cette technique est pertinente pour tous les algorithmes opérant sur des ensembles, où l'ordre des opérations est **a priori** indifférent, mais joue un rôle important et imprévisible en pratique.

Notre objectif principal est de poursuivre l'effort entrepris depuis quelques années pour mettre au point des algorithmes hybrides ayant une meilleure efficacité à la fois théorique et pratique. Cet effort porte à la fois sur les arithmétiques elles-mêmes (voir section 3.3) et leur utilisation pour la résolution de systèmes polynomiaux. Citons pour cette seconde partie un nouvel algorithme hybride d'isolation des racines réelles d'un polynôme de $\mathbb{Z}[x]$ [RZ01]. Pour les algorithmes hybrides entiers-flottants, nous basons nos implantations sur la bibliothèque MPFR [The01]. Nous essayons également d'étendre nos investigations à des arithmétiques hybrides non-classiques, par exemple l'utilisation conjointe de flottants et de nombres modulaires, de nombres p-adiques, etc.

4. Domaines d'application

4.1. (Sans titre)

Dans la section 2.6, nous avons décrit le rôle et l'importance des applications pour notre projet, ainsi que la variété des domaines scientifiques concernés. Dans cette section, nous nous sommes donc limités aux applications les plus importantes, auxquelles des membres du projet consacrent une part appréciable de leur activité.

Il s'agit de la simulation et du contrôle des robots parallèles (section 4.2), du raisonnement géométrique (section 4.3), de la cryptologie (section 4.4) et de la mécanique céleste (section 4.5).

4.2. Robots parallèles

Mots clés : *robot parallèle*, *planification de trajectoires*, *étalonnage*, *calcul exact*.

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

Glossaire

MGD Modèle Géométrique Direct

Les manipulateurs que nous étudions sont des robots parallèles généraux : les hexapodes sont des mécanismes complexes constitués de six chaînes cinématiques souvent identiques, d'une base (corps rigide fixe comprenant six joints ou articulations) et d'une nacelle (corps rigide mobile contenant six autres joints).

La conception et l'étude de robots parallèles nécessitent l'établissement et la résolution de modèles géométriques directs (calcul des coordonnées absolues des points d'attache de la plateforme connaissant la position et la géométrie de la base, la géométrie de la plateforme ainsi que les distances entre les points d'attache des chaînes cinématiques à la base et à la nacelle) et inverses (distances entre les points d'attache des chaînes cinématiques à la base et à la nacelle connaissant la position absolue de la base et de la nacelle). Le modèle géométrique direct se résout facilement. C'est par conséquent sur la résolution du modèle géométrique direct que portent nos efforts.

L'étude du modèle géométrique direct pour les robots parallèles est une activité récurrente de plusieurs membres du projet. On peut dire que les progrès effectués dans ce domaine illustrent parfaitement l'évolution des méthodes de résolution des systèmes algébriques. L'intérêt porté à ce sujet est ancien. Les premiers travaux auxquels ont participé des membres du projet [LM94][Laz92] ont essentiellement porté sur l'étude du nombre de solutions (complexes) du problème. Ils ont souvent été illustrés par des calculs de bases de Gröbner effectués avec le logiciel Gb (voir section 5.3). Un des points remarquables de cette étude est certainement la classification proposée dans [FL95]. Les efforts suivants ont porté sur les racines réelles et le calcul effectif des solutions [Rou95]. Les études se sont ensuite poursuivies au gré des divers progrès algorithmiques réalisés, jusqu'à ce que les outils développés permettent d'appréhender des problèmes non académiques. C'est alors (1999) que les divers efforts se sont concrétisés par un contrat industriel avec la PME vosgienne CMW (Constructions Mécaniques des Vosges Marioni) pour l'élaboration d'un robot dédié aux machines-outils. De nouveaux problèmes apparaissent désormais et concernent deux de nos directions de recherche :

- le calcul en temps réel du modèle géométrique direct. Ceci pourrait se faire en particulier en générant des programmes de calcul numérique à partir d'un calcul exact suffisamment générique (base de Gröbner).
- l'étude de systèmes de dimension positive pour, par exemple, pouvoir traiter des informations dépendant du temps (dimension 1) ou encore permettre de relâcher certains paramètres du problème pour mieux les étudier.

4.3. Raisonnement géométrique

Mots clés : *preuve de théorème, découverte automatique de théorème, géométrie différentielle, classification de nombre de solutions réelles, interpolation de Birkhoff.*

Participants : Philippe Aubry, Daniel Lazard, Fabrice Rouillier, Mohab Safey El Din, Dongming Wang.

Glossaire

CAGD Computer-Aided Geometric Design

Des méthodes et logiciels développés dans notre équipe ont été améliorés et étendus pour l'application au raisonnement géométrique. En particulier, nous avons découvert et prouvé plusieurs théorèmes en géométrie différentielle, résolu un problème géométrique soulevé par deux chercheurs chinois, ainsi que le problème d'interpolation de Birkhoff.

Le raisonnement géométrique est un sujet de recherche fondamental où la résolution algébrique (base de Gröbner, ensemble triangulaire, résolution réelle) a trouvé une application importante. Il a de nombreuses applications dans le domaine de la géométrie informatique, en particulier en CAGD (résolution des contraintes géométriques) et en vision par ordinateur (ajustement de modèles, dérivation des propriétés géométriques, etc.). Le problème du raisonnement géométrique consiste à étudier et établir automatiquement des relations entre objets géométriques, par exemple à prouver un théorème ou découvrir une relation géométrique inconnue. Il s'agit d'attaquer ce problème par des méthodes algébriques en traduisant des relations géométriques en expressions algébriques.

Par exemple, nous avons développé des méthodes et outils d'élimination différentielle pour prouver des théorèmes et dériver des relations géométriques dans la théorie locale des surfaces en géométrie différentielle [AW01]. Plusieurs nouveaux théorèmes ont été établis au moyen de nos méthodes et expérimentations. Une application de l'ensemble des méthodes et techniques développées récemment dans notre équipe permet de résoudre systématiquement un autre problème géométrique (la détermination du nombre de solutions réelles d'un système dépendant de paramètres en fonction de ces paramètres) soulevé par deux chercheurs chinois [Laz01]. Nous avons également réussi à résoudre le problème d'interpolation de Birkhoff par la méthode des points critiques [RSEDS00].

4.4. Cryptologie

Mots clés : *cryptologie, système à clé publique, jacobienne de courbe algébrique.*

Participants : Magali Bardet, Abdolali Basiri, Jean-Charles Faugère, Nicolas Gurel, Guillaume Hanrot.

Glossaire

HFE **Hidden Field Equation**

La cryptologie est le domaine de l'informatique qui s'intéresse aux problèmes de sécurité de l'information sous un sens général. Dans le cadre de Spaces, nous nous intéressons aux applications de nos méthodes dans ce domaine, qui se situent principalement dans le domaine des cryptosystèmes à clé publique.

Les travaux dans ce domaine d'application en sont encore à leurs débuts ; ils se situent dans le cadre de l'action PolyCrypt de l'Action Concertée Incitative (ACI) Cryptologie (voir 8.2).

L'importance cryptographique croissante prise par les jacobiennes de courbes algébriques sur les corps finis, au moins dans une logique prospective, rend de plus en plus fréquent d'avoir affaire à de tels idéaux dans les corps de fonctions des courbes ou de leurs jacobiennes. Calculer dans de tels corps de fonctions peut se faire en manipulant des idéaux d'algèbres de polynômes. Des travaux en préparation améliorent les meilleurs algorithmes connus dans le cas de certaines familles particulières de courbes.

En outre, les avancées sur le calcul du nombre de points de ces courbes en grande caractéristique, indispensable si l'on souhaite s'assurer que le système ne présente pas une faille grossière, semblent actuellement passer par les généralisations de l'algorithme de Schoof (utiliser l'action de l'endomorphisme sur le module de Tate $T_l(J)$), ce qui impose de savoir calculer efficacement modulo les idéaux de l-torsion, qui sont des idéaux d'une algèbre de polynômes en plusieurs variables. Ce problème d'efficacité, actuellement non résolu, est très limitant.

La difficulté supposée des problèmes de résolution de systèmes polynomiaux a conduit à l'élaboration de cryptosystèmes dont la sécurité repose, en particulier, sur la difficulté à résoudre des systèmes polynomiaux sur les corps finis. Notons que la difficulté précise de ces problèmes est encore mal comprise ; généralement la taille de la sortie (base de Gröbner) peut être exponentielle (voire doublement exponentielle dans certains cas très spécifiques) même si en pratique on constate un meilleur comportement moyen. L'un de nos objectifs est d'entreprendre une étude plus systématique du cryptosystème HFE de Patarin.

4.5. Mécanique Céleste

Mots clés : *mécanique céleste, configuration centrale.*

Participants : Jean-Charles Faugère, Daniel Lazard.

Glossaire

configuration centrale Configuration de n corps gravitationnels, telle que toutes les accélérations sont dirigées vers le centre de masse et proportionnelles à la distance à ce centre. Une configuration centrale plane reste semblable à elle-même au cours du temps.

L'étude du mouvement de plusieurs masses ponctuelles soumises à l'attraction newtonienne est un problème très ancien[Eul60]. Devant la difficulté du problème (dès que l'on a plus de trois corps le système différentiel décrivant le mouvement est non intégrable analytiquement) il apparaît naturel de se restreindre aux solutions « simples » de ces équations différentielles. Ainsi la recherche des configurations centrales, permet de se ramener à l'étude d'un problème algébrique. Toutefois l'étude de ces systèmes algébriques est particulièrement difficile en particulier à cause du degré élevé des équations de départ. Les membres de l'équipe collaborent sur ce sujet, depuis plusieurs années, avec des chercheurs du Bureau des Longitudes de l'Observatoire de Paris (autour d'Alain Albouy). Cette collaboration a donné lieu à une thèse co-encadrée (Ilias Kotsireas).

Plus récemment, les progrès algorithmiques du projet nous ont permis de faire progresser sensiblement cette question : Nous avons complètement déterminé, en fonctions des différentes masses, le nombre des

configurations centrales de 4 corps ayant un axe de symétrie, de masses m_1, m_2, m_3, m_4 (avec la contrainte $m_3 = m_4 = 1$). Il y a une, trois ou cinq configurations possibles, suivant la position du point (m_1, m_2) par rapport à une courbe délimitant des régions du plan où la nature et le nombre des solutions changent. Cette courbe est définie par un polynôme à deux variables de degré 424 de plus de 50 000 termes, et qui occupe 10 Mo sur un disque. Le tracé de cette courbe est un des problèmes qui nous ont conduits à mettre au point le logiciel TCI (section 5.8). C'est également la résolution de cet exemple qui nous a amenés à mettre au point un algorithme général de résolution de systèmes dépendant de paramètres (section 6.2).

Cette application est au centre de l'ACI « jeunes chercheurs » dont l'équipe est bénéficiaire (2001–2004) (section 8.2) .

5. Logiciels

5.1. (Sans titre)

L'objectif principal de l'activité logicielle du projet est la réalisation informatique de nouveaux algorithmes améliorant l'efficacité des calculs ou leur domaine d'application.

Le degré de développement et de diffusion de nos logiciels permet une première classification, suivant qu'il s'agisse de prototypes restreints à l'usage de leurs auteurs, ou qu'ils soient plus ou moins diffusés. Cette classification est destinée à évoluer avec le temps.

Une autre classification naturelle de nos logiciels réside dans leur lien avec notre recherche algorithmique. En effet, à côté des logiciels directement liés à notre activité de recherche, l'équipe est amenée à développer d'autres logiciels ayant leur intérêt propre, qu'il s'agisse de logiciels d'infrastructure (gestionnaire de mémoire, protocole de communication) ou de logiciels permettant d'appliquer nos algorithmes à des problèmes spécifiques. Cette classification est dans une certaine mesure arbitraire, notamment en ce qui concerne l'arithmétique qui est au centre des recherches de certains membres du projet, et qui fait partie de l'infrastructure pour d'autres. De même, la simulation du robot parallèle Hexapode n'était, au début, qu'une application ; mais le travail nécessité s'avère si importante qu'il est susceptible de devenir un axe de recherche autonome pour notre équipe. Malgré cet arbitraire relatif, c'est cette classification qui structure la description détaillée suivante.

5.2. MPFR/MPFI

MPFR *et* MPFI *sont des logiciels fondamentaux de l'équipe.*
Mots clés : *arithmétique d'intervalles, précision arbitraire, arrondi exact.*

Participants : David Daney, Guillaume Hanrot, Vincent Lefèvre, Yves Pétermann, Jean-Luc Rémy [avant-projet Adage], Fabrice Rouillier, Paul Zimmermann.

Glossaire

GMP GNU Multi-Precision library
mpfr Multi-Precision Floating point Reliable arithmetic
mpfi Multi-Precision Floating point Interval arithmetic

MPFR est une bibliothèque de calcul flottant en précision arbitraire. Basée sur la bibliothèque GMP, elle garantit ce qu'on appelle l'« arrondi exact » , à savoir que le résultat de chaque opération est le nombre flottant, représentable dans la précision donnée, qui soit le plus proche du résultat exact. Au cours de l'année 2001, les opérations de base (addition, soustraction, multiplication, division) ont été complètement réécrites, et les implantations correspondantes décrites dans un document en cours de rédaction, afin de permettre une preuve formelle de ces implantations. En outre, dans le cadre du post-doctorat de D. Daney, la plupart des fonctions mathématiques du standard C9X ont été implantées (fonctions hyperboliques et leurs inverses, fonctions trigonométriques et leurs inverses, logarithme et exponentielle en base 2 et 10, puissance, etc.). Enfin,

la plage d'exposants autorisés peut maintenant être définie par l'utilisateur, et un mécanisme de débordement (**overflow/underflow**) a été implanté. La version actuelle (environ 15000 lignes de code, plus autant de lignes de test) sera distribuée avec la prochaine version de GMP. En octobre 2001, Y. Pétermann, chercheur de l'Université de Genève invité par l'INRIA, a travaillé avec J.-L. Rémy (avant-projet Adage) sur l'implantation de la fonction zêta de Riemann en MPFR.

MPFI est une bibliothèque d'arithmétique d'intervalles, écrite en C (environ 1000 lignes), basée sur MPFR et développée en collaboration avec N. Revol (actuellement en délégation dans le projet Arénaire à Lyon). Initialement, MPFI a été développée pour les besoins d'un nouvel algorithme hybride pour l'isolation de zéros réels de polynômes. Les fonctionnalités sont actuellement étendues pour permettre l'implantation et l'adaptation d'autres méthodes de résolution (principalement basées sur l'algorithme de Newton) mais le code est déjà disponible et documenté.

5.3. Gb/FGb

Les logiciels GB/FGB *sont des logiciels fondamentaux de l'équipe.*
Mots clés : *Gröbner, interface, logiciel*.

Participant : Jean-Charles Faugère.

GB est un logiciel pour le calcul de bases de Gröbner développé en C++ (100000 lignes). Ce logiciel est très stable et fait l'objet d'une diffusion par le Web. Il est interfacé avec plusieurs logiciels de calcul formel (Maple, MuPAD et Axiom) et avec les logiciels REALSOLVING et RS. FGB est un nouveau logiciel expérimental (91000 lignes de C) dans lequel sont implantés les nouveaux algorithmes F_4, F_5 et F_7. Il est interfacé avec Maple et utilisable via le Web. Toutefois son aspect expérimental n'en permet pas encore une diffusion générale.

5.4. RS/RealSolving

Les logiciels RS *et RealSolving sont des logiciels fondamentaux de l'équipe.*
Mots clés : *zéro réel, système de dimension zéro, polynôme en une variable*.

Participant : Fabrice Rouillier.

RS est un logiciel dédié à l'étude des zéros réels des systèmes algébriques. Il est entièrement développé en C (100000 lignes environ) et succède à REALSOLVING développé lors des projets européens PoSSo et FRISCO. RS contient principalement des fonctions pour le comptage et l'isolation des zéros réels de systèmes algébriques admettant un nombre fini de racines complexes. Les interfaces utilisateur de RS sont entièrement compatibles avec celles de GB/FGB (ASCII, MuPAD, Maple). RS est utilisé dans le projet depuis plusieurs mois et une version de développement a été mise à disposition de diverses équipes. La version actuelle semble suffisamment stable pour être diffusée.

5.5. CharSets/Epsilon

Les logiciels CharSets/Epsilon sont des logiciels fondamentaux de l'équipe.
Mots clés : *ensemble caractéristique, ensemble triangulaire, décomposition de système polynomial, décomposition primaire, factorisation algébrique, preuve de théorème, dessin automatique*.

Participant : Dongming Wang.

Nous avons uniformisé, testé, et complété la documentation de CharSets, GEOTHER et de deux modules d'ensembles triangulaires.

Le module CharSets est une implantation complète de la méthode des ensembles caractéristiques ou triangulaires (voir section 3.1). Ses versions 1.0, 1.1 et 1.2 sont incluses dans la bibliothèque partagée de Maple et distribuées mondialement depuis 1991. La version courante CharSets 2.0 est disponible sur

le web (http://calfor.lip6.fr/ wang/charsets.html) pour des utilisations académiques. Les différents algorithmes implantés dans CharSets sont basés sur le calcul d'ensembles caractéristiques de Wu [Wan01a][Wan01c].

Epsilon est une bibliothèque d'outils pour l'élimination polynomiale, encore en développement. Elle sera constituée d'une nouvelle version du module CharSets, de l'environnement GEOTHER [CGL+00] (http://calfor.lip6.fr/ wang/GEOTHER/) conçu et implanté pour la manipulation et la preuve de théorèmes géométriques, et de modules qui implantent deux méthodes d'ensembles triangulaires proposées par D. Wang. Epsilon permet de

- trianguliariser des systèmes quelconques de polynômes à plusieurs variables,
- décomposer de tels systèmes en systèmes triangulaires de différentes sortes (réguliers, simples, irréductibles, ou munis de propriétés de projection), et donc les résoudre,
- décomposer des variétés algébriques en composantes irréductibles ou équidimensionnelles,
- décomposer des idéaux polynomiaux en composantes primaires,
- factoriser des polynômes sur des extensions algébriques successives de corps,
- prouver des théorèmes géométriques et dessiner des diagrammes automatiquement,
- traduire des spécifications géométriques en expressions algébriques et en langue naturelle automatiquement, et
- accomplir une partie des tâches ci-dessus pour des systèmes de polynômes différentiels ordinaires.

5.6. UDX

UDX *est un logiciel d'infrastructure de l'équipe.*
Mots clés : *protocole binaire, efficacité, portabilité.*

Participants : Fabrice Rouillier, Jean-Charles Faugère.

Glossaire

UDX Universal Data eXchange
XDR eXternal Data Representation

UDX est un logiciel pour l'échange de données binaires. Il a été implanté à l'origine pour montrer la puissance d'un nouveau protocole, objet d'un dépôt de brevet par l'INRIA et l'UPMC (numéro de dépôt 99 08172, 1999). Le code résultant, écrit en C ANSI, est très portable et d'une efficacité certaine, même lorsque le protocole breveté n'est pas utilisé. UDX est composé de cinq modules indépendants :

- base : système optimisé de tampons et de synchronisation des entrées et sorties ;
- supports : opérations de lecture/écriture sur des supports variés (**sockets**, fichiers, mémoire partagée, etc.) ;
- protocoles : protocoles d'échanges variés (protocole breveté, XDR, etc.) ;
- échanges de types composites : flottants simple et double précision, entiers, rationnels, flottants multi-précision ;
- interfaces : interfaces utilisateur de haut niveau faisant appel aux quatre autres modules.

UDX est utilisé dans le projet pour interfacer les différents composants logiciels entre eux ou avec des outils de calcul extérieurs (MuPAD par exemple). L'application la plus spectaculaire utilisant UDX est certainement l'interface entre les logiciels MuPAD et Scilab.

5.7. Interfaces

Ces interfaces sont des logiciels d'infrastructure de l'équipe.
Participants : Jean-Charles Faugère, Fabrice Rouillier.

Afin de rendre aisée l'utilisation des divers logiciels développés dans le projet, des conventions d'échanges de fichiers ASCII et binaires ont déjà été mises au point et permettent une utilisation souple des serveurs de calcul GB/FGB/RS.

5.8. Tracé de courbes implicites (TCI)

TCI *est un logiciel d'application de l'équipe.*
Mots clés : *courbe implicite, tracé certifié.*

Participants : Jean-Charles Faugère, Daniel Lazard, Fabrice Rouillier.

Dès qu'il s'agit d'applications réelles, les polynômes issus de procédés d'élimination (bases de Gröbner, ensembles triangulaires) sont très souvent trop gros pour pouvoir être étudiés par les outils généraux de calcul formel. Dans le cas de polynômes en 2 variables, un tracé certifié suffit dans beaucoup de cas pour *résoudre* le problème étudié (c'est le cas notamment pour certaines applications en mécanique céleste). Ce type de tracé est maintenant possible grâce aux différents outils développés dans le projet.

5.9. Simulateur de planification de trajectoires (SPT)

SPT *est un logiciel d'application de l'équipe.*
Le projet SPT a pour but d'homogénéiser et d'exporter l'ensemble des outils développés dans le cadre de nos applications en robotique parallèle. Les composants logiciels de SPT sont essentiellement des implantations en C suivant les standards du logiciel RS, des algorithmes écrits en MuPAD ou Maple utilisant les prototypes d'interfaces pour GB et RS. L'encapsulation de tous ces composants dans une distribution unique ne posera donc aucun problème et est planifiée à court terme (début 2002). Un des objectifs (moyen et long terme) est également d'intégrer ou d'interfacer ce produit avec des composants extérieurs (principalement développés dans l'avant-projet COPRIN à Sophia Antipolis), de sorte à fournir une solution globale pour l'étude des manipulateurs parallèles.

6. Résultats nouveaux

6.1. Bases de Gröbner

Participants : Jean-Charles Faugère, Magali Bardet, Gwenolé Ars, Abdolali Basiri.

Mots clés : *base de Gröbner, paramètre, groupe cyclique, racine cyclique.*

Une base de Gröbner n'est souvent pas la meilleure façon de représenter les solutions d'un système algébrique. Toutefois l'expérience montre que, d'une part, c'est l'objet que l'on peut calculer le plus efficacement actuellement et que, d'autre part, à partir d'une base de Gröbner (ordre lexicographique) il est relativement aisé de calculer une forme plus commode à utiliser (ensemble triangulaire, RUR, ...). Le premier résultat obtenu est une variante de l'algorithme F_5 très efficace sur presque tous les systèmes. C'est ainsi que les problèmes servant de test de référence (Cyclic 9 et Cyclic 10) ont été traités avec succès [Fau01]. Cela permet de gagner un ordre de grandeur par rapport à l'algorithme F_4 et ce aussi bien pour les systèmes de dimension zéro que pour les systèmes de dimension positive. Ainsi Cyclic 9 est un système de dimension positive (dimension 2) et Cyclic 10 admet 34940 solutions complexes.

6.2. Systèmes dépendant de paramètres

Participants : Jean-Charles Faugère, Daniel Lazard, Magali Bardet, Gwenolé Ars, Abdolali Basiri.

Nous avons proposé une méthode pour résoudre des systèmes dépendant de paramètres pour un grand nombre de valeurs de ceux-ci : le calcul est effectué une première fois à l'aide du programme FGb sur une **instance** du problème. Ce programme génère une trace du calcul sous forme de programme C contenant uniquement des opérations arithmétiques élémentaires. Le programme ainsi généré est ensuite compilé et peut être exécuté très

rapidement pour **d'autres valeurs** des paramètres. Cette méthode a été employée pour générer un programme permettant le décodage d'un code correcteur (temps d'exécution 10^{-4} sec) et un programme effectuant le calcul de toutes les solutions d'un robot parallèle (10^{-3} à 10^{-2} seconde). Le passage de l'application en robotique au décodage d'un code correcteur nécessite simplement de changer l'arithmétique. Le logiciel FGb permet le choix de 18 arithmétiques (entiers, corps finis, arithmétiques hybrides).

Dans certains cas, on peut également utiliser les techniques de bases de Gröbner pour générer des **formules** qui sont ensuite **évaluées** pour un jeu de paramètres. C'est ce qui a été fait pour le calcul dans les groupes jacobiens de courbes super-elliptiques (généralisation de l'arithmétique des courbes elliptiques). Les formules ainsi produites permettent de compter avec précision le nombre d'opérations (additions et multiplications) et de générer un programme très efficace (10^{-5} seconde pour faire une addition dans le groupe d'une courbe C_{34}).

Par ailleurs, nous avons développé un algorithme général pour calculer, en fonction des paramètres, le nombre de solutions réelles d'un système d'équations polynomiales qui satisfont certaines inégalités. Cet algorithme se décompose en deux grandes étapes, la décomposition en composantes irréductibles des solutions complexes, suivie d'un algorithme adapté aux idéaux premiers.

La première étape est décrite et illustrée sur un exemple de géométrie élémentaire dans [Laz01]. La deuxième étape, techniquement beaucoup plus complexe, a été appliquée à un problème de mécanique céleste (voir section 4.5). Un article est en cours de rédaction pour à la fois décrire et démontrer cet algorithme, et l'illustrer par cette application.

6.3. Zéros réels des systèmes de dimension positive

Mots clés : *composante connexe, zéro réel, système algébrique, point critique.*

Participants : Philippe Aubry, Fabrice Rouillier, Marie-Françoise Roy, Mohab Safey El Din.

Dans [RRSED00], nous développons un algorithme pour le calcul d'au moins un point sur chaque composante connexe pour toute variété définie par une seule équation. Le principe en est simple : on calcule les points critiques de la fonction distance à un point bien choisi de manière à se ramener à l'étude d'un système zéro-dimensionnel, lorsque la variété étudiée ne contient qu'un nombre fini de points singuliers. Dans le cas contraire (et uniquement dans ce cas), l'algorithme proposé effectue une (et une seule) déformation infinitésimale.

Ces travaux ont été complétés par un algorithme certifié calculant une Représentation Univariée Rationnelle d'un système d'équations polynomiales de dimension zéro à coefficients infinitésimaux. Ceci a permis d'implanter (en Magma) dans son intégralité l'algorithme proposé dans [RRSED00] afin de le tester, y compris dans les cas pathologiques.

Parallèlement, nous avons consolidé les résultats de [ARSED00]. L'objectif de ces travaux vise à explorer une alternative aux déformations infinitésimales, pour la gestion des singularités. L'algorithme obtenu ne prend plus en entrée une unique équation, mais un système d'équations polynomiales. Le principe consiste toujours à calculer les points critiques d'une fonction bien choisie (distance à un point), mais lorsque des singularités sont présentes dans la variété de départ, celles-ci sont étudiées comme une variété algébrique à part entière, de dimension strictement inférieure à celle de la variété initiale. Cet algorithme effectue donc une induction sur la dimension des variétés intermédiaires apparaissant en cours de calcul. Un premier prototype (en Maple, sur la base des logiciels Gb/RS) de cet algorithme a été implanté. Nous avons par ailleurs proposé plusieurs variantes de cet algorithme pour :

- étudier les variétés algébriques réelles compactes;
- substituer aux calculs de bases de Gröbner des calculs d'ensembles triangulaires.

Dans [RSEDS00], nous comparons expérimentalement les deux approches algorithmiques de [RRSED00] et [ARSED00] sur le problème d'interpolation de Birkhoff (il s'agit de décider si une variété définie par une unique équation admet des solutions réelles). Certaines de ces variétés admettent une infinité de singularités. Pour celles-ci, la deuxième stratégie s'est avérée plus efficace (pour certains exemples, notre prototype permet de conclure en quelques dizaines de secondes, là où la première stratégie ne le permet pas).

Ces tests ont été élargis à des systèmes polynomiaux connus et ont montré que nos algorithmes sont considérablement plus efficaces que la CAD. L'ensemble de ces résultats est exposé dans [SED01].

Le travail actuellement en cours donne une bonne idée de l'ampleur de la tâche à accomplir dans ce domaine. En effet, le calcul des objets mathématiques de base intervenant dans les algorithmes envisagés (principalement des décompositions du radical d'un idéal en composantes équidimensionnelles) est considéré actuellement comme délicat. Mais notre travail met surtout en évidence l'intersection très forte entre les divers thèmes de recherche du projet : la plupart des méthodes à l'étude font appel de manière conséquente à des algorithmes de décomposition d'idéaux ou de variétés (décomposition en idéaux premiers, ensembles triangulaires, etc.), et leur étape finale nécessite d'isoler les zéros réels de polynômes en une variable.

6.4. Polynômes univariés

Mots clés : *zéro réel, isolation, factorisation, résolution par radicaux, corps fini.*

Participants : Ethan Cotterill [stagiaire MIT], Guillaume Hanrot, Fabrice Rouillier, Paul Zimmermann.

Deux types de résultats ont été obtenus sur l'étude des racines des polynômes en une variable. D'une part, nous avons revisité l'algorithme d'USPENSKY mis au point par Collins et Akritas en proposant une nouvelle variante, optimale en occupation mémoire. Nous avons cette année étendu ces résultats en mettant au point une version hybride (utilisant une arithmétique flottante) et généralisé l'utilisation de ces techniques au cas des polynômes à coefficients algébriques réels.

D'autre part, des résultats ont été obtenus sur la factorisation de polynômes sur les extensions algébriques de $\mathbb{Q}$ (corps de nombres) ou l'expression de racines d'équations par radicaux quand cela est possible.

6.4.1. Zéros réels

Dans [RZ01], nous proposons une description unifiée de l'ensemble des variantes de l'algorithme d'USPENSKY pour l'isolation des zéros réels (par des intervalles à bornes rationnelles) de polynômes en une variable à coefficients dans un corps archimédien. L'ensemble des outils introduits nous a permis de produire une nouvelle variante de l'algorithme, utilisant le même nombre d'opérations arithmétiques que les variantes les plus efficaces, mais optimale en occupation mémoire. Ce progrès considérable permet de traiter maintenant de nouvelles classes d'exemples jusqu'alors inaccessibles (polynômes orthogonaux de degré 1000, polynômes issus de processus d'élimination).

Nous avons aussi cherché à remplacer l'arithmétique exacte, jusqu'alors utilisée, par une arithmétique d'intervalles multi-précision (MPFI) basée sur la bibliothèque MPFR (voir section 5.2), avec pour objectif de fournir un algorithme plus rapide (arithmétique en faible précision) mais préservant l'exactitude du résultat. Il y avait deux obstacles principaux :

- contrôler l'accumulation d'erreurs numériques (gérée par MPFI) ;
- contourner le prérequis théorique de l'algorithme d'Uspensky : le polynôme à traiter doit être sans facteur carré, notion n'ayant pas de sens précis dès lors que ses coefficients sont définis par des intervalles.

L'utilisation de la règle de Descartes n'impose pas une précision trop grande puisque seuls les signes des coefficients calculés importent. Dans le cadre de l'utilisation d'une arithmétique d'intervalles, il convient donc de rajouter un test d'arrêt dès lors que l'on ne peut calculer cette borne c'est-à-dire en gros dès lors que 0 apparaît dans un intervalle. Les spécifications de l'algorithme résultant sont de ce fait modifiées : il prend en entrée un polynôme et retourne une liste d'intervalles contenant une et une seule racine, et éventuellement une autre liste d'intervalles pour lesquels aucune décision concernant la présence de zéros réels n'a été possible.

Lorsque les coefficients des polynômes sont représentés par des intervalles, l'algorithme traite, en quelque sorte, une famille de polynômes. On peut montrer simplement que, dès lors que les arbres de calcul induits par l'application de l'algorithme sur ces polynômes divergent (intervalles actifs différents), le même calcul effectué avec une arithmétique d'intervalles induira des problèmes de décision de signes que l'on pourra traiter comme ci-dessus. En particulier, si une famille de polynômes représentée par un polynôme dont les coefficients sont des intervalles contient un polynôme avec facteurs multiples, un problème de décision de signe apparaîtra nécessairement. Ceci implique que l'algorithme d'Uspensky avec arithmétique d'intervalles muni des spécifications précédentes terminera toujours. Dans le cas où le polynôme d'entrée est à coefficients exacts, on peut alors faire le calcul en précision fixe (arithmétique d'intervalle) pour ne traiter ensuite que les ouverts de la droite réelle pour lesquels aucune décision n'a été possible. Grâce à l'ensemble de fonctions décrites dans [RZ01], le calcul peut alors être repris à l'endroit exact où il a échoué avec une précision plus grande ou une arithmétique exacte, ce qui donne un algorithme adaptatif, utilisant le minimum de précision nécessaire, optimal en mémoire, plus rapide que toutes les versions existantes et fournissant à coup sûr un résultat exact.

Cette nouvelle variante de l'algorithme d'USPENSKY sert de base aux routines de tracé de courbes implicites (TCI), et d'opération terminale dans le processus d'isolation de zéros réels de systèmes zéro-dimensionnels.

Avec un algorithme permettant de traiter, au moins partiellement, des polynômes dont les coefficients sont donnés par des intervalles, il était naturel de s'intéresser à la résolution de polynômes à coefficients algébriques réels (c'est-à-dire de systèmes triangulaires en deux variables), et plus généralement aux solutions réelles de systèmes triangulaires zéro-dimensionnels. A priori, trois méthodes permettent d'appréhender ce type de problèmes :

- (I) calculer une Représentation Univariée Rationnelle du système triangulaire pour se ramener en une passe à un problème en une variable;
- (II) conserver les nombres algébriques réels en tant que variables. L'algorithme d'Uspensky, dans sa version la plus efficace, ne faisant intervenir que des additions, on assure alors un bon contrôle des coefficients intervenant en cours de calcul, l'opération principale consistant à déterminer le signe d'un polynôme en une variable évalué en les zéros d'un autre polynôme.
- (III) remplacer les nombres algébriques réels par une approximation à l'aide d'intervalles.

La version (I) est la plus simple mais également la moins efficace puisqu'elle concentre toute l'information sur le système en un seul polynôme, parfois délicat à calculer. La version (III) ne permet pas de conclure assurément puisque le polynôme à traiter ne sera jamais connu exactement. Enfin, la version (II) pose un problème puisqu'on ne peut, a priori, traiter que des polynômes sans facteur carré. Cette difficulté peut cependant être facilement contournée en assurant certaines propriétés : ensemble triangulaire régulier séparable dont le premier polynôme (en une variable) est sans facteur carré.

Comme l'illustre le travail effectué par E. Cotterill durant son stage au LORIA, la bonne façon de résoudre le problème consiste à débuter avec la méthode (III) et de basculer sur (II) en cas d'échec. Cette stratégie a été employée avec succès sur plusieurs séries d'intersections de quadriques proposées par le projet ISA (INRIA Lorraine) : la rapidité des calculs engendrés et l'exactitude des résultats fournis semblent montrer que cet outil de calcul est bien adapté pour le traitement d'applications réelles.

6.4.2. Résolution par radicaux

La section précédente explique l'étude de la recherche de solutions numériques de polynômes en une variable. Il arrive que l'on soit intéressé par les solutions formelles exprimées comme une imbrication de radicaux. Un résultat théorique dû à Galois et dans le principe effectif explique à quelle condition cela est possible (**i.e.**, que le groupe de Galois de l'équation soit résoluble).

G. Hanrot et François Morain (LIX, École Polytechnique) ont repris le travail de Galois et ont complètement explicité comment, à partir d'approximations numériques des racines dans une clôture algébrique complexe ou p-adique et d'une description de l'action du groupe de Galois comme permutation sur les racines, il est possible de construire la suite d'extensions dont la théorie de Galois prédit l'existence, et de résoudre chacune

des équations relatives correspondantes par radicaux. Ce travail a été publié dans les actes de la conférence ISSAC'01 [HM01]. Notons que ce travail pourrait avoir des applications dans le cadre des preuves de primalité utilisant l'algorithme ECPP (**Elliptic Curve Primality Proving**), dû à Atkin et Morain.

6.4.3. Factorisation sur les corps de nombres

Un des problèmes importants du calcul formel est la factorisation des polynômes en une variable sur un corps donné. Un algorithme récent de van Hoeij [vH] a nettement amélioré la situation dans le cas où le corps de base est $\mathbb{Q}$. Un travail de Belabas (Université Paris XI), Hanrot et Zimmermann [BHZ01] montre comment résoudre les problèmes qui se posent quand on essaie d'étendre ces méthodes au cas des extensions algébriques de $\mathbb{Q}$.

Notons que le travail du paragraphe précédent nécessite en particulier la factorisation de polynômes cyclotomiques sur les corps de nombres (polynômes dont les racines sont les racines primitives de l'unité d'un certain ordre).

Avec D. Lin (Académie des Sciences de Chine, Pékin), D. Wang [WL01] a présenté une méthode pour la factorisation de polynômes sur des extensions algébriques de corps, méthode qu'il a introduite et implantée en 1992.

6.4.4. Factorisation de trinômes sur GF (2)

Les polynômes creux sont très intéressants pour les applications car ils donnent lieu à des calculs plus rapides. Sur le corps fini $\mathrm{GF}(2)$ à deux éléments, les polynômes creux les plus intéressants sont les trinômes x^r+x^s+1. Une des applications importantes est la génération aléatoire : un polynôme primitif de degré r sur $\mathrm{GF}(2)$ permet de construire un générateur aléatoire de période 2^r-1. Avec R. Brent (Oxford) et Samuli Larvala (Helsinki), nous nous sommes intéressés aux trinômes sur $\mathrm{GF}(2)$ dont le degré r est un exposant de Mersenne, c'est-à-dire tels que 2^r-1 est premier. Cela permet de ramener le test de primitivité à celui (plus simple) d'irreductibilité. Dans [BLZ00], un nouvel algorithme est proposé pour tester l'irreductibilité d'un trinôme sur $\mathrm{GF}(2)$; cet algorithme apporte à la fois un gain en mémoire et en temps de calcul. Une implantation en C de cet algorithme nous a permis de vérifier les calculs effectués par Kumada et **al.**[6], et de trouver un trinôme « oublié » par ces auteurs, à savoir $x^{859433}+x^{170340}+1$. Nous avons effectué une recherche complète pour $r=3021377$ [BLZ00], et la recherche pour $r=6972593$ (plus grand exposant de Mersenne connu) est en cours, grâce notamment au soutien du CINES.

6.5. Ensembles triangulaires

Mots clés : *ensemble triangulaire, sous-résultant, décomposition triangulaire.*

Participants : Philippe Aubry, Marie-Françoise Roy, Mohab Safey El Din, Dongming Wang.

6.5.1. Sous-résultants

Le calcul de la suite des sous-résultants est à la base de nombreux algorithmes calculant des ensembles triangulaires de polynômes, d'où l'intérêt de l'optimiser au maximum.

Dans [LRSED00], de nouveaux théorèmes de structure de cette suite sont donnés et un algorithme en est déduit. L'intérêt de cet algorithme ne réside pas dans le nombre d'opérations effectuées (quadratique en le degré des polynômes donnés en entrée, comme les algorithmes classiques) mais dans le meilleur contrôle de la taille des coefficients intermédiaires (qui n'excèdent pas deux fois la taille des mineurs de la matrice de Sylvester). Il en résulte que les implantations de cet algorithme sont plus efficaces que celles de ses concurrents, en particulier dans le cas de polynômes univariés à coefficients polynomiaux.

Dans [HW00], nous donnons une représentation explicite et simple des sous-résultants en termes de mineurs de la matrice de Bézout. Cette représentation donne une technique effective pour le calcul des chaînes de sous-résultants par l'évaluation de déterminants. L'avantage de cette approche réside dans le fait que la dimension

[6] T. Kumada, H. Leeb, Y. Kurita et M. Matsumoto, **New primitive t-nomials $(t=3, 5)$ over $\mathrm{GF}(2)$ whose degree is a Mersenne exponent**, Mathematics of Computation 69 (2000).

de la matrice de Bézout est inférieure à celle de la matrice de Sylvester. Un algorithme est déduit de ces résultats et semble avoir un intérêt pour une certaine classe de problèmes.

6.5.2. Calcul d'ensembles triangulaires

Nous avons proposé dans [Wan01c] un algorithme généralisé pour le calcul des ensembles caractéristiques de Wu au moyen de pseudo-réductions au lieu de pseudo-divisions. Avec une sélection optimale de réducteurs et une génération heuristique de S-polynômes, notre algorithme peut accélérer le calcul considérablement et produire des sorties plus simples pour une large classe de problèmes.

6.5.3. Ensembles triangulaires et théorie de Galois

En théorie de Galois algébrique effective nous avons contribué à une nouvelle approche qui développe des liens avec l'algèbre commutative, liens qui se limitaient jusqu'à maintenant aux notions d'idéal des relations et d'idéal des relations symétriques. Dans [AV00] nous précisons la structure d'idéaux plus généraux qui interviennent en théorie de Galois et proposons de nouveaux algorithmes qui permettent de calculer de manière algébrique les résolvantes relatives ainsi que la base de Gröbner lexicographique de tels idéaux, qui a pour particularité d'être aussi un ensemble triangulaire. Il est ainsi possible de déterminer l'idéal des relations entre les racines d'un polynôme, pour travailler ensuite dans le corps de décomposition de ce polynôme.

6.6. Raisonnement géométrique

Mots clés : *preuve de théorème, découverte automatique de théorème, géométrie différentielle, algèbre de Clifford, ensemble triangulaire.*

Participants : Philippe Aubry, Dongming Wang.

Dans [AW01] nous avons développé des méthodes et outils d'élimination différentielle pour la preuve et la découverte de théorèmes en géométrie différentielle. Un algorithme de décomposition de systèmes différentiels réguliers est basé sur les techniques algébriques de scindage des ensembles triangulaires développées dans le projet. Plusieurs nouveaux théorèmes dans la théorie locale des surfaces ont été découverts et prouvés au moyen de nos méthodes et expérimentations.

En collaboration avec X. Hou, H. Li et L. Yang de l'Académie des Sciences de Chine, D. Wang a montré, en utilisant des méthodes algébriques [HLWY01], comment on peut prouver automatiquement l'un des théorèmes géométriques présentés comme des défis par S. Markelov, un expert russe en géométrie.

Dans la continuité de son travail sur le raisonnement géométrique utilisant l'algèbre de Clifford, D. Wang a expérimenté une approche effective pour la preuve d'identités dans cette algèbre, utilisant le principe d'induction avec simplification heuristique [Wan01e].

6.7. Géométrie informatique

Participants : Jean-Charles Faugère, Daniel Lazard, Fabrice Rouillier.

Mots clés : *géométrie, quadrique, intersection, graphe d'aspect.*

Au cours de l'année 2001, une collaboration importante s'est développée avec le projet ISA (Sylvain Lazard et Sylvain Petitjean). Au-delà de la mise à la disposition d'ISA des logiciels de Spaces, elle a porté sur deux directions.

En premier lieu, nous avons pu montrer que le graphe d'aspect d'une surface de degré 4 à 6 pouvait être effectivement calculé. Il s'agit là d'un résultat tout à fait préliminaire, qui nécessite encore un travail d'expérimentation important avant d'être publié.

Ensuite, l'essentiel de notre collaboration a porté sur la paramétrisation des surfaces quadriques et de leurs intersection, en vue du calcul des configurations d'arcs de courbes définis par une famille de quadriques.

Ainsi, nous avons pu mettre au point un algorithme robuste pour paramétriser ces quadriques et leurs intersections, et nous avons montré que, dans tous les cas, il est optimal du point de vue du nombre nécessaire d'extraction de racines carrées [DLPL01].

6.8. Robots parallèles

Mots clés : *robot parallèle, modèle géométrique direct, planification de trajectoires, étalonnage.*

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

Glossaire

MGD Modèle Géométrique Direct

Dans notre projet, les travaux liés à l'étude des robots parallèles sont principalement dictés par les besoins liés à la mise au point du robot CMW 300 par la société CMW (Constructions Mécaniques des Vosges Marioni). Nous travaillons selon 2 axes :

- la planification de trajectoires,
- les problèmes d'étalonnage.

6.8.1. Planification de trajectoires

La conception d'un serveur de calcul dédié à la résolution du modèle géométrique direct et utilisant l'algorithme F_4 [J.-99] nous a permis de gagner un ordre de grandeur important en termes de temps de calcul et de remettre en question les mises en équations jusqu'alors utilisées. Sur ce problème précis, nous avons pu constater que le calcul direct, par l'algorithme F_4, d'une base de Gröbner pour un ordre lexicographique est certainement le meilleur moyen de calculer une Représentation Univariée Rationnelle.

Bien que cette solution soit en elle-même d'une efficacité appréciable (quelques secondes de calcul pour obtenir les solutions exactes du problème géométrique direct pour n'importe quel type de robot) nous l'avons adossée à une méthode de Newton hybride certifiée (calcul utilisant une arithmétique d'intervalles avec des tests forts[7] de convergence et de précision du résultat) pour étudier une trajectoire donnée. Ainsi, la méthode exacte n'est lancée que dans les cas où la méthode de Newton hybride échoue. Au final, l'outil d'étude de trajectoires est tout à fait satisfaisant du point de vue de l'utilisation pratique : il nous a permis de diagnostiquer facilement et en quelques minutes les insuffisances de certains systèmes de commandes. Nous profitons de ces avancées pour participer à l'optimisation de parcours d'outils dans le cadre de l'usinage à grande vitesse de pièces complexes, pour le compte de la société CMW.

6.8.2. Étalonnage

La commande des robots parallèles est dépendante de la connaissance des paramètres qui composent leur modélisation géométrique. Mais les défauts de fabrication et d'assemblage du robot détériorent cette connaissance et ainsi affectent leur précision de positionnement. Une procédure d'étalonnage est alors nécessaire, à travers la résolution de systèmes sur-contraints d'équations liant les paramètres géométriques à identifier et les mesures de positionnement du robot. Afin de simplifier la mise en oeuvre expérimentale, nous nous sommes intéressés, en collaboration avec I. Emiris (avant-projet Galaad, INRIA Sophia Antipolis), à des techniques algébriques d'élimination de variables. Le but est ici de supprimer une partie de l'information nécessaire à l'identification des paramètres géométriques. Les techniques utilisées sont basées sur l'obtention d'une matrice du résultant ou de sa généralisation d'après la méthode d'élimination dialytique. Ainsi, nous avons montré [DE01a][DE01b] comment l'élimination algébrique permet d'obtenir des équations de contraintes qui ne sont dépendantes que de la mesure de la position du robot dans différentes configurations. La mesure de son orientation, plus sensible aux bruits des mesures et plus difficile à réaliser, est algébriquement supprimée. Comparés aux résultats fournis par quelques techniques existantes, les systèmes d'équations obtenus par ces techniques sont beaucoup plus stables numériquement, surtout pour des configurations de mesures situées à la frontière de l'espace de travail. Dans ce cas, la fiabilité et la simplicité de l'étalonnage sont particulièrement améliorées.

6.9. Arithmétique

Mots clés : *précision arbitraire, arrondi exact, produit médian, théorie des nombres.*

[7] Si le test est vrai, alors l'algorithme converge assurément.

Participants : Guillaume Hanrot, Vincent Lefèvre, Paul Zimmermann.

6.9.1. Arithmétique des ordinateurs

Plusieurs membres du projet ont participé à un numéro spécial de la revue « Calculateurs Parallèles » sur l'arithmétique des ordinateurs. V. Lefèvre a rédigé un article sur la multiplication par une constante [Lef01b], et P. Zimmermann un article de synthèse sur l'arithmétique en précision arbitraire (nombres entiers et flottants) [Zim02].

En ce qui concerne la multiplication par une constante, le but est de générer du code effectuant une multiplication par des constantes entières à l'aide d'opérations élémentaires (décalages, additions et soustractions), les constantes pouvant être des entiers de plusieurs centaines ou milliers de bits. Un tel code peut être utilisé notamment dans les algorithmes du style Toom-Cook servant à multiplier des entiers à grande précision et dans le calcul approché de valeurs consécutives d'un polynôme, qui intervient dans la recherche des pires cas pour les fonctions élémentaires.

Dans le cadre d'un projet BQR (Bonus Qualité Recherche) de l'Université Henri Poincaré Nancy 1, en collaboration avec Joël Rivat et Gérald Tenenbaum de l'IECN (Institut Élie Cartan de Nancy), G. Hanrot et P. Zimmermann ont résolu une conjecture de Jean-Michel Muller (projet Arénaire, INRIA Rhône-Alpes) sur la proportion de nombres flottants x admettant un inverse (**i.e.**, pour lesquels il existe un nombre flottant y tel que l'arrondi de $x \cdot y$ donne exactement 1). Ce résultat sera publié dans un numéro spécial de la revue **Theoretical Computer Science** [HRTZ02].

6.9.2. Arithmétique des séries

Le travail sur le « produit médian » [HQZ00] a été accepté pour publication dans la revue AAECC (**Applicable Algebra in Engineering, Communication and Computing**), après révision. Rappelons que le résultat du produit médian de deux polynômes consiste en les coefficients médians de ce produit. Au cours de cette révision, plusieurs résultats nouveaux ont été mis en évidence. Notamment, il a été montré que tout circuit calculant le produit de deux polynômes de degré $n-1$ (avec des noeuds addition, négation, duplication, multiplication) peut se transformer en un circuit calculant le produit médian de deux polynômes de degré $n-1$ et $2n-2$, et réciproquement. On a donc formellement l'équivalence $\mathrm{MP}(n) \sim M(n)$ entre les coûts asymptotiques $\mathrm{MP}(n)$ pour le produit médian et $M(n)$ pour la multiplication. Ce résultat est fortement lié au « principe de transposition » concernant le calcul de produit matrice-vecteur.

La version finale de l'article sur le « produit médian » (en cours de rédaction) améliore toutes les meilleures complexités connues pour la division et la racine carrée de séries formelles, avec ou sans reste, dans les classes de complexité de la multiplication par l'algorithme de Karatsuba et par la transformée de Fourier rapide (FFT). Pour la multiplication via Karatsuba, ces algorithmes sont implantés dans un module MAPLE disponible sur `http://www.loria.fr/ zimmerma/papers/MP.mpl`.

6.9.3. Théorie des nombres

Enfin, citons deux résultats de G. Hanrot utilisant de manière conséquente ses travaux antérieurs sur l'algorithmique des équations diophantiennes. L'un [BHV01] en collaboration avec Y. Bilu et P. Voutier résoud complètement une conjecture sur les diviseurs des entiers de la forme $(\alpha^n-\beta^n)/(\alpha-\beta)$. L'autre [HSS01] avec N. Saradha et T. Shorey de Bombay progresse dans l'étude d'une conjecture d'Erdös-Selfridge, qui cherche à déterminer quand un produit d'entiers consécutifs, duquel on peut omettre au plus un terme, est presque une puissance pure.

7. Contrats industriels

7.1. Interface MuPAD-Scilab

Mots clés : *calcul formel, calcul numérique, interface.*

Participants : Fabrice Rouillier, Paul Zimmermann.

En novembre 2002 a été signé entre la société SciFace (qui distribue le logiciel de calcul formel MuPAD) et l'INRIA Lorraine (représentant l'Université Pierre et Marie Curie) un accord de coopération autour du programme UDXF d'échange de données binaires. UDXF prend en paramètre un protocole de communication, par exemple celui du brevet UDX, mais ce peut être aussi un autre protocole. Dans le cadre de cet accord, SciFace a le droit d'utiliser UDXF pour l'interface MuPAD-Scilab. En échange, SciFace participe au développement et à l'amélioration de UDX.

8. Actions régionales, nationales et internationales

8.1. Actions régionales

8.1.1. Projet Bonus Qualité Recherche

Participants : Guillaume Hanrot, Paul Zimmermann.

Un projet BQR (Bonus Qualité Recherche) sur les « Applications en Théorie des Nombres » a été financé par l'Université Henri Poincaré Nancy 1 pour la période 2000-2001. Il est coordonné par Joël Rivat de l'IECN (Institut Élie Cartan de Nancy). En 2001, plusieurs groupes de travail ont eu lieu, ayant conduit à une publication [HRTZ02] résolvant une conjecture de Jean-Michel Muller.

8.2. Actions nationales

8.2.1. Action concertée incitative Cryptologie « PolyCrypt »

Participants : Guillaume Hanrot [chef de projet], Magali Bardet, Abdolali Basiri, Jean-Charles Faugère, Nicolas Gurel.

L'action « PolyCrypt », menée dans le cadre de l'ACI Cryptologie pilotée par le ministère, regroupe les chercheurs de Spaces listés ci-dessus ainsi que D. Augot (projet CODES), N. Brisebarre (Université de St-Étienne) et Ph. Gaborit (Université de Limoges). Cette action a été acceptée en juillet 2001 pour une durée de 3 ans, et la première réunion s'est tenue le 23 octobre 2001.

Il s'agit d'une action d'interface, étudiant des applications à la cryptologie de domaines situés en amont de celle-ci, principalement ceux de la compétence du projet, à savoir l'arithmétique (dans le cas des corps finis) et les systèmes polynomiaux.

Nos premiers objectifs sont d'étudier l'algorithmique des corps finis et d'en offrir une implantation la plus efficace et générique possible (au-delà des applications cryptographiques, il serait souhaitable d'avoir une arithmétique pour des valeurs moyennes du degré et de la caractéristique), et d'étudier les problèmes de systèmes polynomiaux quand le corps de base est spécifiquement fini ; dans ce cas on dispose d'équations supplémentaires du type $x^N = x$ sur les variables, et on peut se demander de quelle façon les exploiter dans la procédure d'élimination.

Les applications à la cryptographie sont décrites dans la section 4.4.

8.2.2. Action concertée incitative « Jeunes Chercheurs » - Résolution des systèmes algébriques avec paramètres

Participants : Jean-Charles Faugère [chef de projet], Fabrice Rouillier.

Cette action se déroulera sur une durée de 3 ans à compter de janvier 2001. Le but est de mettre au point des outils performants pour la résolution des systèmes algébriques à paramètres. Un certain nombre de travaux en cours dans le projet (zéros réels des systèmes de dimension positive, tracé de courbes implicites) sont directement liés à cette action. Initialement porté par deux personnes le nombre de participants à cette action grandit puisque Ph. Aubry, M. Safey El Din, S. Corvez et M. Bardet contribuent désormais aux travaux de recherche.

8.3. Actions européennes

8.3.1. Projet européen RAAG

Participants : Fabrice Rouillier [contact pour le projet Spaces], Marie-Françoise Roy.

RAAG (**Real Algebraic and Analytic Geometry**) est un réseau de formation par la recherche (**Research Training Network**) du programme "**Human Potential**" de la Commission Européenne, subventionné pour une durée de 48 mois à compter de mars 2002. Le réseau est géré par l'Université de Passau (Allemagne), la coordination du travail de l'équipe française étant assurée par l'équipe de géométrie réelle et calcul formel de l'Université de Rennes I.

Les thèmes abordés dans RAAG sont la géométrie et l'algèbre réelles ainsi que les algorithmes et leur complexité. Le projet Spaces participe principalement à la formation de jeunes chercheurs (pré- ou post-doctorants) aux outils de calcul permettant d'étudier les zéros réels des systèmes d'égalités et d'inégalités polynomiales.

8.3.2. Coopération avec l'Université de Paderborn

Participants : Fabrice Rouillier, Paul Zimmermann.

Depuis janvier 2001, le protocole de communication UDX est inclus dans la version de développement du complexe logiciel MUPAD/SCILAB. Ce travail est effectué dans le cadre de l'accord de coopération établi pour la distribution commune des logiciels MuPAD et Scilab.

8.4. Actions internationales

8.4.1. Coopération avec l'Université de Sydney

Participants : Guillaume Hanrot [responsable], Paul Zimmermann.

Depuis janvier 2001, une collaboration est en place entre l'équipe développant le logiciel MAGMA à l'Université de Sydney (autour de J. Cannon) et trois équipes françaises : l'équipe de J.-M. Couveignes au GRIMM (Université de Toulouse 2), l'équipe de F. Morain au LIX (École Polytechnique), et l'équipe Spaces. Cette collaboration a pour objectif d'utiliser MAGMA comme base de tests pour l'implantation des algorithmes développés dans ces trois équipes, autour de l'algorithmique des nombres et polynômes, des courbes, de leurs jacobiennes, avec des applications à la génération de cryptosystèmes présumés sûrs basés sur les jacobiennes de courbes algébriques.

Cette collaboration est partiellement financée par l'ambassade de France à Canberra pour l'année civile 2001.

Côté Spaces, P. Zimmermann s'est rendu deux semaines à Sydney en octobre-novembre 2001. Durant ce séjour, outre de nombreuses discussions autour de la sémantique des calculs flottants et de l'arithmétique d'intervalles, ont été implantés en MAGMA un algorithme de division sans reste basé sur le produit médian, qui a permis de gagner un facteur 2 comme prévu sur l'implantation existante, ainsi qu'une version efficace de la seconde étape (**Stage 2**) des algorithmes de factorisation ECM et Pollard $p-1$.

8.5. Visites, et invitations de chercheurs

8.5.1. Invitations

Participants : Richard Brent, Yves Pétermann, Paul Zimmermann.

L'INRIA Lorraine a permis l'invitation de deux chercheurs étrangers en 2001. R. Brent (Université d'Oxford) a été invité en août et septembre ; il a donné deux conférences, dont l'une pour une large audience sur la factorisation d'entiers, et a travaillé avec P. Zimmermann sur le calcul flottant en précision arbitraire, ainsi que la recherche de trinômes primitifs sur $\mathrm{GF}(2)$. Y. Pétermann a été invité en octobre : avec J.-L. Rémy (avant-projet Adage), il a effectué un calcul d'erreur précis pour l'évaluation de la fonction zêta de Riemann, ce qui devrait permettre l'implantation de cette fonction dans la bibliothèque MPFR.

9. Diffusion des résultats

9.1. Articles et conférences de synthèse

La conférence invitée [Laz00b] de D. Lazard fait le point sur l'état de l'art de la résolution des systèmes d'équations algébriques. Elle fait suite et complète sur ce sujet particulier l'article plus général [Laz00] paru dans le numéro spécial « an 2000 » de la revue TSI.

Aux Journées Nationales de la Recherche en Robotique (JNRR 2001), D. Lazard a présenté l'histoire des relations entre l'étude des robots parallèles et la résolution des systèmes polynomiaux [Laz00].

L'article [Wan01b] présente un examen des théories, méthodes, implantations et applications diverses de l'élimination polynomiale.

Le livre [Wan01a] par D. Wang contient un traitement systématique des algorithmes d'élimination pour décomposer des systèmes arbitraires de polynômes à plusieurs variables en systèmes triangulaires de différentes sortes (réguliers, simples, irréductibles, ou munis de propriétés de projection), en fournissant les décompositions des ensembles des zéros associés. Certains algorithmes pertinents comme ceux fondés sur les résultants ou les bases de Gröbner sont passés en revue. Des applications de ces méthodes d'élimination sont présentées, concernant des aspects algorithmiques en géométrie algébrique, la théorie des idéaux de polynômes, la résolution des systèmes algébriques, la démonstration automatique en géométrie, etc.

L'article [Zim02] fait une synthèse des meilleurs algorithmes connus pour la manipulation de nombres entiers ou flottants en précision arbitraire (multiplication, division, racine carrée, fonctions élémentaires et spéciales). Y sont décrits à la fois les algorithmes naïfs de complexité quadratique, les algorithmes de type « diviser pour régner » comme celui de Karatsuba pour la multiplication, et les algorithmes asymptotiquement optimaux comme ceux basés sur la transformée de Fourier rapide (FFT).

9.2. Exposés invités

F. Rouillier a fait un exposé invité au workshop « **Algorithmic and Quantitative Aspects of Real Algebraic Geometry in Mathematics and Computer Science** » à DIMACS (**The Center for Discrete Mathematics and Theoretical Computer Science**, Rutgers University).

J.-C. Faugère a fait des exposés invités en 2001: Cocoa 7 (Kingston Canada), ASCM 2001 (Matsuyama Japon), Dagstuhl (Allemagne) et Newton Institute (Cambridge Angleterre).

D. Wang a fait des exposés invités à l'Université de Pékin (Chine, Septembre 2001), à l'Université de Kobe (Japon, Septembre 2001), et au séminaire Dagstuhl 01421 (Allemagne, Octobre 2001).

G. Hanrot, V. Lefèvre et P. Zimmermann ont chacun donné un exposé dans le cadre de l'école de printemps « Arithmétique des ordinateurs », organisée à Prapoutel-les-sept-Laux en mars.

G. Hanrot a donné des exposés invités au séminaire de théorie des nombres de Lyon 1 et aux séminaires d'algorithmique arithmétique et de théorie des nombres de Bordeaux 1.

9.3. Organisation de conférences et journées

F. Rouillier a co-organisé avec M.-F. Roy (Université de Rennes 1) et L. Gonzales-Vega (Université de Santander) des journées « **Applications of real algebraic geometry outside mathematics** » à Rennes en juin 2001.

F. Rouillier a organisé un mini-symposium « Calcul Formel » au premier congrès SMAI, dans lequel sont intervenus D. Lazard et J.-C. Faugère.

F. Rouillier a organisé une journée de concertation sur l'« avenir du calcul formel en France » en septembre.

D. Wang a co-organisé en décembre 2000 la conférence internationale ASCM 2000 (**Asian Symposium on Computer Mathematics**).

Le 26 novembre 2001, à l'occasion de la soutenance de l'habilitation à diriger des recherches de P. Zimmermann [Zim01], une journée « arithmétique » a été organisée au LORIA, où Jean-Michel Muller et Jean Vuillemin ont fait chacun un exposé.

Les membres du projet ont organisé les 27 et 28 novembre à Nancy la réunion finale de l'action de recherche coopérative AOC (Arithmétique des Ordinateurs Certifiée).

9.4. Comités de programme et de rédaction

D. Wang est coéditeur des actes de colloque ASCM 2000 [GW00], dont il a été co-président du comité de programme.

En tant que co-président du comité de programme, D. Wang a co-rédigé avec J. Richter-Gebert les actes du colloque ADG 2000 [RGW01] sur le raisonnement automatique en géométrie.

D. Wang est membre des comités de programme pour :

- ATCM 2000 (**Asian Technology Conference in Mathematics**, Chiang Mai, Thaïlande, 17–21 décembre 2000),
- ATCM 2001 (Melbourne, Australie, 15–19 décembre 2001),
- ASCM 2001 (Matsuyama, Japon, 26–28 septembre 2001).

P. Zimmermann a fait partie du comité de programme de la conférence ARITH'15, qui s'est tenue à Vail (Colorado) en juin 2001.

D. Lazard est membre depuis l'origine du comité de programme de la conférence MEGA (Efffective Methods in Algebraic Geometry) dont la prochaîne édition aura lieu en juin 2003 à Kaiserslautern (Allemagne). Il fait également partie du comité de programme de ISSAC 2002 (International Symposium on Symbolic and Algebraic Computation) qui aura lieu à Lille en juillet 2002.

9.5. Enseignement

G. Hanrot occupe un poste de chargé d'enseignement à temps partiel à l'École Polytechnique. Dans ce cadre, il a assuré les travaux dirigés de cours de cryptologie et d'algorithmique aux étudiants de dernier semestre. En outre, il a assuré conjointement avec F. Morain (École Polytechnique) le cours « théorie algorithmique des nombres » du DEA Algorithmique (ENS, ENS Cachan, ENST, Universités Paris 6, 7, 11, École Polytechnique). À compter de novembre 2001, il assure le cours « théorie algébrique des nombres » du DEA de mathématiques pures de l'Université Henri Poincaré Nancy 1.

G. Hanrot et P. Zimmermann assurent le cours de « théorie algorithmique des nombres et cryptographie » du DEA informatique de Nancy.

J.-C. Faugère a dispensé un cours de DEA sur les bases de Gröbner (Université de Rennes 1).

J.-C. Faugère, D. Lazard et F. Rouillier ont assuré les cours de la filière « Calcul formel » du DEA Algorithmique (ENS, ENS Cachan, ENST, Universités Paris 6, 7, 11, École Polytechnique).

10. Bibliographie

Bibliographie de référence

[ALM99] 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.

[Aub99] P. AUBRY. *Ensembles triangulaires de polynômes et résolution de systèmes algébriques.* thèse de doctorat, Université Paris 6, 1999.

[FL95] J. FAUGÈRE, D. LAZARD. *The Combinatorial Classes of Parallel Manipulators.* in « Mechanism and Machine Theory », volume 30, 1995, pages 765–776.

[J.-99] J.-C. FAUGÈRE.. *A new efficient algorithm for computing Gröbner bases (F4).*-. in « Journal of Pure and Applied Algebra », numéro 1–3, volume 139, 1999, pages 61–88.

[Laz92] D. LAZARD. *Stewart platforms and Gröbner basis.* in « Proceedings of Advances in Robotics Kinematics », pages 136-142, 1992.

[Laz00] D. LAZARD. *Calcul formel : tendances et progrès récents.* in « Technique et Science Informatique », volume 19, 2000, note : Numéro spécial, Informatique.

[Lef00] V. LEFÈVRE. *Moyens arithmétiques pour un calcul fiable.* Thèse de doctorat, École Normale Supérieure de Lyon, janvier, 2000.

[LM94] D. LAZARD, J.-P. MERLET. *The (true) Stewart platform has 12 configurations.* in « Proc. of IEEE Conference on Robotics and Vision, San Diego », 1994.

[Rou95] F. ROUILLIER. *Real Root Counting For some Robotics problems.* in « Solid Mechanics and its Applications, Kluwer Academic Publishers », volume 40, 1995, pages 73-82.

[Rou99] F. ROUILLIER. *Solving zero-dimensional systems through the rational univariate representation.* in « Journal of Applicable Algebra in Engineering, Communication and Computing », numéro 5, volume 9, 1999, pages 433–461.

Livres et monographies

[GW00] X.-S. GAO, D. WANG. *Computer Mathematics – Proceedings of the Fourth Asian Symposium (ASCM 2000).* série Lecture Notes Series on Computing, World Scientific Publishing, address Singapore, 2000.

[RGW01] éditeurs J. RICHTER-GEBERT, D. WANG., *Automated Deduction in Geometry, LNAI 2061.* éditeurs J. RICHTER-GEBERT, D. WANG., Springer, Berlin Heidelberg, 2001.

[Wan01a] D. WANG. *Elimination methods.* série Texts and Monographs in Symbolic Computation, Springer, address Wien New York, 2001.

Thèses et habilitations à diriger des recherche

[SED01] M. SAFEY EL DIN. *Résolution réelle des systèmes polynomiaux de dimension positive.* Thèse d'université, Université Paris 6, janvier, 2001.

[Zim01] P. ZIMMERMANN. *De l'algorithmique à l'arithmétique via le calcul formel.* Habilitation à diriger des recherches, Université Henri Poincaré Nancy 1, 2001.

Articles et chapitres de livre

[AV00] P. AUBRY, A. VALIBOUZE. *Using Galois ideals for computing relative resolvents.* in « Journal of Symbolic Computation », numéro 6, volume 30, décembre, 2000, pages 635–651, note : Special Issue on Algorithmic Galois Theory.

[BHV01] Y. BILU, G. HANROT, P. VOUTIER. *Existence of primitive divisors of Lucas and Lehmer numbers.* in « Journal Für die reine und angewandte Mathematik », volume 539, octobre, 2001, pages 75-122.

[HLWY01] X. HOU, H. LI, D. WANG, L. YANG. *"Russian Killer" No. 2: A Challenging Geometric Theorem with Human and Machine Proofs.* in « The Mathematical Intelligencer », volume 23(1), 2001, pages 9-15.

[HRTZ02] G. HANROT, J. RIVAT, G. TENENBAUM, P. ZIMMERMANN. *Density results on floating-point invertible numbers.* in « Theoretical Computer Science », 2002, note : À paraître.

[HSS01] G. HANROT, N. SARADHA, T. SHOREY. *Almost perfect powers in consecutive integers.* in « Acta Arithmetica », numéro 1, volume 99, 2001, pages 13–25.

[Lef01b] V. LEFÈVRE. *Multiplication par une constante.* in « Réseaux et Systèmes Répartis, Calculateurs Parallèles », numéro 3, volume 13, 2001, pages 465-484.

[LRSED00] H. LOMBARDI, M.-F. ROY, M. SAFEY EL DIN. *New structure theorems for subresultants.* in « Journal of symbolic computation », numéro 4-5, volume 29, avril, 2000, pages 663-690.

[RRSED00] F. ROUILLIER, M.-F. ROY, M. SAFEY EL DIN. *Finding at least one point in each connected component of a real algebraic set defined by a single equation.* in « Journal of Complexity », volume 16, 2000, pages 716-750.

[Wan01b] D. WANG. *Elimination Theory, Methods, and Practice.* éditeurs D. LIN, W. LI, Y. YU., in « Mathematics and Mathematics-Mechanization », Shandong Education Publishing House, Jinan, 2001, pages 91-137.

[Wan01d] D. WANG. *Geometric Algebra and Reasoning.* éditeurs E. B.-C. ETG. SOBCZYK., in « Geometric Algebra: A Geometric Approach to Computer Vision, Quantum and Neural Computing, Robotics and Engineering », Birkhauser, Boston, 2001.

[Wan01e] D. WANG. *Geometric Reasoning with Geometric Algebra.* éditeurs E. BAYRO-CORROCHANO, G. SOBCZYK., in « Advances in Geometric Algebra with Applications », Birkhäuser, Boston, 2001, pages 87-111.

[WL01] D. WANG, D. LIN. *A Method for Multivariate Polynomial Factorization over Successive Algebraic Extension Fields.* éditeurs D. LIN, W. LI, Y. YU., in « Mathematics and Mathematics-Mechanization », Shandong Education Publishing House, Jinan, 2001, pages 138-172.

[Zim02] P. ZIMMERMANN. *Arithmétique en précision arbitraire.* in « Calculateurs Parallèles (Hermès) », 2002, note : À paraître..

Communications à des congrès, colloques, etc.

[AW01] P. AUBRY, D. WANG. *Reasoning about Surfaces Using Differential Zero and Ideal Decomposition.* éditeurs D. W. J. RICHTER-GEBERT., in « Third International Workshop on Automated Deduction in Geometry - ADG'2000, Zurich », série Lecture Notes in Artificial Intelligence, volume 2061, organisation J. Richter-Gebert, D. Wang, Springer-Verlag, pages 154-174, address Berlin Heidelberg, 2001.

[DE01a] D. DANEY, I. Z. EMIRIS. *Robust parallel robot calibration with partial information.*. in « IEEE International Conference on Robotics and Automation (ICRA), Corée, Séoul », 2001.

[DE01b] D. DANEY, I. Z. EMIRIS. *Variable elimination for reliable parallel robot calibration.*. éditeurs F. C. PARK, C. C. IURASCU., in « In 2nd Workshop on Computational Kinematics (CK), Seoul, Korea », organisation School of Mechanical and Aerospace Engineering, 2001.

[DLPL01] L. DUPONT, S. LAZARD, S. PETITJEAN, D. LAZARD. *Towards the Robust Intersection of Implicit Quadrics.* in « Workshop on Uncertainty in Geometric Computations, Sheffield, UK », juillet, 2001.

[Fau01] FAUGÈRE J.C. *Finding all the solutions of Cyclic 9 using Gröbner basis techniques.*. éditeurs K. SHIRAYANAGI, K. YOKOYAMA., in « Computer Mathematics – Proceedings of the 5th Asian Symposium (ASCM 2001) », série Lecture Notes Series on Computing, volume 9, World Scientific, Singapore, pages 1–12, Septembre, 2001, note : Conférence invitée.

[HM01] G. HANROT, F. MORAIN. *Solvability by radicals from an algorithmic point of view.* éditeurs B. MOURRAIN., in « ISSAC, London, Ontario », ACM, juillet, 2001.

[HW00] X. HOU, D. WANG. *Subresultants with the Bézout Matrix.* éditeurs X.-S. GAO, D. WANG., in « Computer Mathematics – Proceedings of the 4th Asian Symposium (ASCM 2000) », World Scientific, Singapore, pages 19–28, 2000.

[Laz00b] D. LAZARD. *Resolution of polynomial systems* . in « 4th Asian Symposium on Computer Mathematics - ASCM 2000, Chiang Mai, Thailand », série Lecture Notes Series on Computing, volume 8, World Scientific, pages 1 - 8, décembre, 2000.

[Laz01] D. LAZARD. *On the specification for solvers of polynomial systems.* in « 5th Asian Symposium on Computer Mathematics - ASCM 2001, Matsuyama, Japon », série Lecture Notes Series on Computing, volume 9, World Scientific, pages 66 - 75, septembre, 2001.

[LM01] V. LEFÈVRE, J.-M. MULLER. *Worst Cases for Correct Rounding of the Elementary Functions in Double Precision.* éditeurs N. BURGESS, L. CIMINIERA., in « 15th IEEE Symposium on Computer Arithmetic - ARITH 2001, Vail, Colorado », pages 111-118, juin, 2001.

[Rol01] L. ROLLAND. *Méthodes algébriques pour la résolution du modèle géométrique de robots parallèles, applications à haute cadence et grande précision.* in « Quatrièmes Journées du Pôle Microrobotique, Lyon, France », organisation Laboratoire d'Automatique Industrielle, juillet, 2001.

[RSEDS00] F. ROUILLIER, M. SAFEY EL DIN, E. SCHOST. *Solving the Birkhoff interpolation problem via the critical point method : an experimental study.* in « Automated Deduction in Geometry, Zurich, Switzerland », 2000.

[Wan01c] D. WANG. *A Generalized Algorithm for Computing Characteristic Sets.* éditeurs K. SHIRAYANAGI, K. YOKOYAMA., in « Computer Mathematics – Proceedings of the 5th Asian Symposium (ASCM 2001) », pages 165–174, 2001.

Rapports de recherche et publications internes

[ARSED00] P. AUBRY, F. ROUILLIER, M. SAFEY EL DIN. *Real solving for positive dimensional systems.* Rapport de recherche, septembre, 2000.

[BHZ01] K. BELABAS, G. HANROT, P. ZIMMERMANN. *Tuning and Generalizing Van Hoeij's Algorithm.* Rapport de recherche, institution INRIA, février, 2001, http://www.inria.fr/rrrt/rr-4124.html

[BLZ00] R. P. BRENT, S. LARVALA, P. ZIMMERMANN. *A Fast Algorithm for Testing Irreducibility of Trinomials mod 2.* Rapport de recherche, institution Oxford University Computing Laboratory, décembre, 2000.

[Lef01a] V. LEFÈVRE. *Multiplication by an Integer Constant.* Rapport de recherche, institution INRIA, mai, 2001, http://www.inria.fr/rrrt/rr-4192.html

[RZ01] F. ROUILLIER, P. ZIMMERMANN. *Efficient Isolation of a Polynomial Real Roots.* Rapport de recherche, institution INRIA, février, 2001, http://www.inria.fr/rrrt/rr-4113.html

Divers

[The01] THE MPFR TEAM. *The MPFR library.* howpublished http://www.mpfr.org/, 2001.

Bibliographie générale

[ASZ00] J. ABBOTT, V. SHOUP, P. ZIMMERMANN. *Factorization in Z $[x]$: the searching phase.* éditeurs C. TRAVERSO., in « Proceedings of ISSAC'2000 », pages 1–7, 2000, http://www.shoup.net/papers/asz.ps.Z

[CGL+00] S.-C. CHOU, X.-S. GAO, Z. LIU, D.-K. WANG, D. WANG. *Geometric Theorem Provers and Algebraic Equations Solvers.* éditeurs X.-S. GAO, D. WANG., in « Mathematics Mechanization and Applications », Academic Press, London, 2000.

[Col75] G. COLLINS. *Quantifier elimination for real closed fields by cylindrical algebraic decomposition.* in « Springer Lecture Notes in Computer Science 33 », volume 33, 1975, pages 515-532.

[dW55] V. DERWAERDEN. *Moderne Algebra.* Springer-Verlag, 1955.

[ePM97] A. H. K. ETP. MARKSTEIN. *High-Precision Division and Square Root.* in « ACM Transactions on Mathematical Software », numéro 4, volume 23, 1997, pages 561–589.

[Eul60] L. EULERUS. *Opera omnia. Series secunda (Opera mechanica et astronomica), Vol. XXV: Commentationes astronomicae ad theoriam perturbationum pertinentes, Vol. primum.* Auctoritate et impensis Societatis Scientiarum Naturalium Helveticae. Orell Füssli, Zurich, 1960, chapitre Considerationes de motu corpurum coelestium, note : version originale en 1762.

[HQZ00] G. HANROT, M. QUERCIA, P. ZIMMERMANN. *Speeding up the Division and Square Root of Power Series.* Rapport de recherche, numéro 3973, 2000, http://www.inria.fr/RRRT/RR-3973.html

[Mul00] T. MULDERS. *On short multiplications and divisions.* in « AAECC », numéro 1, volume 11, 2000, pages 69–88.

[vH] M. VANHOEIJ. *Factoring polynomials and the knapsack problem.* in « Journal of Number Theory », http://www.math.fsu.edu/~hoeij/papers.html note : À paraître.