
Projet SAFIR

Systèmes Algébriques et Formels pour l'Industrie et la Recherche

Localisation : *Sophia Antipolis*

Mots-clés : algèbre linéaire, analyse de document, anneau local, arithmétique exacte, arithmétique modulaire probabiliste, base de données déductives, base de formules, bézoutien, calcul formel, certification du résultat, communication, complexité, dualité, éditeur de formules, électronique, élimination, FFT, génération de code, grammaire de graphe, Hankel, implémentation d'algorithmes géométriques, interface homme-machine, Lisp, matrice structurée, modélisation, optique, perturbation symbolique infinitésimale, polynôme multivariable, précision limitée, protocole, reconnaissance de caractères, réécriture de graphe, résolution de systèmes polynomiaux, résidu, résultant, simulation, système inverse, Toeplitz, valeur singulière.

1 Composition de l'équipe

Responsables scientifiques

Loïc Pottier, CR Inria
Stephen Watt, Professeur UNSA, chercheur IBM

Responsable permanent

José Grimm, CR Inria

Assistante de projet

France Limouzis, TR

Personnel Inria

Stéphane Dalmas, IR
Ioannis Emiris, CR
Bernard Mourrain, CR
Yves Papegay, CR

Personnel de l'Université de Nice-Sophia Antipolis (UNSA)

Frédérique Barkats, Professeur Agrégé
Mohamed Elkadi, Maître de conférences
Frédéric Eyssette, Maître de conférences
Marc Gaëtano, Maître de conférences
André Galligo, Professeur
Roger Marlin, Professeur
Michel Merle, Professeur

Chercheurs invités

David Jeffrey, University of Western Ontario, du 1^{er} février au 30 avril
 Victor Pan, City University of New-York, du 1^{er} avril au 31 août

Ingénieurs experts

Christèle Faure
 Nicole Rostaing-Schmidt, à l'Ifremer depuis le 1^{er} février

Chercheurs post-doctorants

Jérôme De Sousa, boursier ADER-PACA, du 1^{er} mars au 31 août
 Alain Sausse, boursier FRISCO, du 1^{er} mars au 31 mai

Chercheurs doctorants

Olivier Arzac, allocataire Inria
 Didier Bondyfalat, depuis octobre 1996
 Eric Hassold, allocataire MESR
 Claude Huchet, allocataire MESR
 Stéphane Lavirotte, allocataire MESR
 Noëlle Stolfi, allocataire MESR
 Mohamed Tadjouddine, allocataire MESR
 Vladimir Vyskocil, allocataire MESR

Stagiaires

Noël Arnous, DEA Info UNSA, 2 mois
 Luc Desouche, DEA Math UNSA, 9 mois
 Yannis Chicha, DEA Info UNSA, 5 mois
 Gabriel Dos Reis, ENS, 4 mois
 Serge Fantino, DEA Math UNSA, 4 mois
 Thierry Giordano, DEA Math UNSA, 6 mois
 Michel Guillaume, DEA Math UNSA, 2 mois
 Fabrice Livigni, ESSI, 3 mois
 Florent Tournois, ENS, 3 mois

2 Présentation du projet

SAFIR est un projet commun à l'Inria, au CNRS et à l'université de Nice-Sophia Antipolis, via le laboratoire de mathématiques, J.-A. Dieudonné (URA 168) et le laboratoire d'informatique, signaux et systèmes, I3S (URA 1376).

En continuité avec les années précédentes, les recherches du projet concernent trois facettes du calcul formel :

- l'algorithmique en algèbre et en géométrie, où monômes et polynômes jouent un rôle central,
- les logiciels : interfaces, communication, déductions et preuves,
- l'application au calcul numérique, avec la différentiation automatique de programmes Fortran, la modélisation et la simulation de systèmes physiques.

Comme c'est le cas dans un projet Inria, ces thèmes ont en commun, à des degrés divers, la recherche de résultats théoriques nouveaux, le développement de prototypes logiciels et leur application/validation sur des domaines concrets. Il est important de noter que ces domaines ne sont pas indépendants — chacun réagit avec les autres dans ce domaine pluridisciplinaire qu'est le calcul formel.

Le projet s'insère dans la communauté scientifique nationale (PRC-GDR AMI, GDR Medicis, GDR Météorologie), européenne (réseau SAC et réseau SOL, consortium FRISCO), et internationale (comités de programmes de colloques du domaine, organisation et participation à des ateliers thématiques).

En outre, SAFIR étant un projet commun à l'Inria et à l'université de Nice-Sophia Antipolis, l'enseignement lié au calcul formel est aussi une activité importante des membres du projet (DEA à Nice, Marseille, et Paris, école d'ingénieurs ESSI, premier et second cycles, et classes préparatoires).

L'année 96 a été marquée par l'intégration dans le projet de Stephen Watt, spécialiste reconnu des systèmes de calcul formel (Maple, Scratchpad, Axiom, A²) et actuel président de SIGSAM. Après avoir à plein temps participé à la direction du projet, depuis octobre, Stephen Watt y consacre le tiers de son temps, étant le reste du temps chercheur au centre de recherche d'IBM à Yorktown Heights (USA). Le projet a profité de cette venue sous plusieurs aspects scientifiques et techniques :

- le projet européen ESPRIT-LTR FRISCO a débuté cette année, pour trois ans, sur le thème de la résolution numérique et symbolique de systèmes d'équations polynomiales dans les problèmes industriels,
- le protocole de communication ASAP développé dans le projet est maintenant au cœur du prototype OpenMath (futur standard international pour l'échange d'objets mathématiques),
- la différentiation automatique a non seulement trouvé de nouvelles applications (ESSILOR, EDF), mais s'est aussi orientée vers des réflexions plus théoriques, notamment sur les problèmes d'analyse de programmes,
- l'atelier SNAP sur les calculs approchés en algorithmique des polynômes organisé en juillet à Sophia Antipolis a attiré tous les spécialistes du sujet.

Le projet a aussi largement profité de la venue pendant six mois de Victor Pan comme professeur invité, spécialiste du calcul matriciel, avec qui nous avons travaillé sur le développement et l'utilisation de techniques matricielles pour le calcul effectif de solutions d'équations polynomiales.

3 Actions de recherche

3.1 Algorithmique en algèbre et en géométrie

3.1.1 Algèbre linéaire pour les polynômes

Participants : Ioannis Emiris, Bernard Mourrain, Thierry Giordano, Florent Tournois, Fabrice Livigni

Mots-clés : résolution de systèmes polynomiaux, algèbre linéaire, élimination, résultant, bézoutien.

Ce travail sur la résolution de systèmes polynomiaux par des méthodes matricielles fait suite à celui entamé l'année dernière. Nous considérons en particulier des systèmes dont les coefficients sont donnés de manière approchée, et nous nous intéressons à la stabilité numérique des algorithmes proposés en vue d'applications en géométrie, vision et robotique. Ce travail se divise en plusieurs étapes : constructions de matrices d'élimination (résultants, résultants creux, bézoutiens,...), méthode de réduction et compression de ces matrices et enfin, obtention des solutions par des calculs de valeurs propres.

Parmi les méthodes d'élimination, le *résultant creux* généralise le résultant de Sylvester de deux polynômes univariés, le déterminant des coefficients d'un système linéaire ainsi que le résultant classique de Macaulay d'un système dense homogène. Pour un système de n polynômes en n variables, le *volume mixte* des polyèdres de Newton donne la borne de Bernstein sur le nombre des racines, qui est plus précise que la borne de Bézout. Durant le stage de T. Giordano, nous avons proposé et implémenté une version distribuée de l'algorithme *Lift-Prune* qui est, à ce jour, la façon la plus rapide de calculer le volume mixte¹.

¹voir <http://www.inria.fr/safir/SAM/Mixvol-distrib/mixvol.html>

Ce volume mixte exprime la complexité intrinsèque dans le cadre de l'élimination creuse, tandis que la complexité des algorithmes est typiquement une fonction du volume de la somme de Minkowski des polytopes de Newton. Une relation entre le volume mixte et le volume de cette somme a été établie dans [440], ce qui permet d'obtenir des bornes sur la complexité en fonction de la structure creuse.

Le stage de F. Tournois était, lui, consacré au calcul de bézoutiens par interpolation et FFT. Ce travail s'intègre dans la bibliothèque ALP, écrite en C++ et développée par F. Livigni durant son stage, consacré à l'implémentation de méthodes de compression et réduction de matrices issues de systèmes polynomiaux. Cette bibliothèque utilise certaines fonctions de la bibliothèque LAPACK et permet de construire une algèbre quotient approchée puis, par des calculs de valeurs propres, de retrouver les racines du système.

Les différentes méthodes utilisées dans cette approche matricielle sont présentées dans un rapport Inria à paraître² et appliquées aux calculs de configurations d'une molécule à 6 atomes³. Ce travail s'inscrit dans le cadre du projet européen FRISCO. Des exposés sur ce thème ont été faits lors de l'atelier de travail SNAP à Sophia Antipolis.

3.1.2 Matrices structurées et systèmes polynômiaux

Participants : Ioannis Emiris, Bernard Mourrain, Victor Pan, Gabriel Dos Reis

Mots-clés : matrice structurée, Toeplitz, Hankel, FFT, polynôme multivariable.

La visite de V. Pan comme professeur invité dans notre projet nous a conduit à un nouveau champ d'investigations, à savoir les liens entre la théorie des matrices structurées et la résolution des systèmes polynômiaux.

Dans l'article ⁴, V. Pan et B. Mourrain ont présenté de nouvelles classes de matrices structurées généralisant celles des matrices de Toeplitz et de Hankel au cas multivariables et comprenant les matrices de résultants, de résidus et les matrices de bézoutiens. La généralisation d'algorithmes rapides pour multiplier de telles matrices avec des vecteurs et de la notion de rang de déplacement a ainsi conduit à un algorithme de sélection de racines de systèmes polynômiaux permettant de réduire la complexité asymptotique d'un ordre de grandeur.

Par ailleurs, V. Pan et I. Emiris se sont intéressés à l'utilisation de ces structures pour le calcul des résultants creux, permettant également de réduire la complexité asymptotique du calcul par un facteur presque linéaire en la dimension de la matrice.

Dans son stage, G. Dos Reis s'est intéressé à une amélioration de l'algorithme de Weyl proposée par V. Pan et à son implémentation en C++. Cet algorithme (ayant une très bonne complexité théorique) s'appuie sur les transformées de Fourier et l'utilisation des structures de type Toeplitz ou Hankel.

3.1.3 Calcul symbolique en géométrie

Didier Bondyfalat commence sa thèse (coencadré par O. Faugeras et B. Mourrain).

3.1.4 Algorithmes pour polynômes approchés

Participants : Ioannis Emiris, André Galligo, Stephen Watt

Mots-clés : précision limitée, certification du résultat, valeur singulière.

Ayant défini la notion du Plus Grand Commun Diviseur (PGCD) pour des polynômes en une variable dont les coefficients sont connus en précision limitée, nous continuons ce travail en étudiant des bornes

²I. Z. Emiris et B. Mourrain, Polynomial system solving; the case of a 6-atom molecule

³voir aussi en démonstration <http://www.inria.fr/safir/DEMO/cyclohex/>

⁴«Solving special polynomial systems by using structured matrices and algebraic residues», Foundations of Computational Mathematics, Springer-Verlag, à paraître

plus précises et des algorithmes plus efficaces. Les applications potentielles, par exemple en vision, en robotique, en biologie moléculaire et en automatique, sont nombreuses, puisque les données provenant d'une expérience sont, en général, bruitées.

L'article [447] de Emir, Galligo et Lombardi a, en particulier, amélioré un ancien théorème de certification pour le degré du PGCD. Nous définissons un saut dans les valeurs singulières des matrices de sous-résultant qui garantit le degré du PGCD approché. La preuve du théorème est constructive et donne un algorithme pour calculer le PGCD lui-même. Cet algorithme a une complexité polynomiale en les degrés des polynômes donnés et du PGCD calculé. De plus, l'algorithme est numériquement stable, puisqu'il utilise la décomposition en valeurs singulières (SVD). Une norme pondérée est aussi présentée, parce qu'elle aboutit à des bornes plus précises.

Un problème lié, qui se réduit au calcul SVD, est la factorisation absolue de polynômes à deux variables, ayant pour coefficients des nombres flottants. Si le polynôme est sans facteur carré, on considère en un point générique une solution implicite du polynôme, que l'on développe en série. Des candidats facteurs sont obtenus en considérant des polynômes de degrés croissants, qui admettent des développements limités de la solution implicite comme solutions implicites tronquées.

3.1.5 Systèmes inverses et points avec multiplicités

Participants : Mohamed Elkadi, Bernard Mourrain

Mots-clés : dualité, système inverse, résidu, anneau local, résultant.

Une méthodologie en pleine évolution en calcul formel (poursuivant en cela les travaux de Jacques Morgenstern) consiste à regarder les polynômes (et plus généralement les fonctions) comme des algorithmes qui calculent une valeur en un point. Dans un travail récent, B. Mourrain s'intéresse aux évaluations pour elles-mêmes, c'est-à-dire comme élément du dual de l'algèbre des polynômes. Il propose un algorithme travaillant dans le dual et permettant de calculer (avec une meilleure complexité que celle existant actuellement) l'anneau local associé à un point avec multiplicité. Quelques applications au calcul algébrique de résidus locaux, aux connexions entre résultants et résidus, à l'analyse locale de branches réelles de courbes intersections complètes y sont proposées. Ces travaux sont présentés dans [455].

Par ailleurs, poursuivant le travail sur la théorie algébrique des résidus et les applications, M. Elkadi et B. Mourrain ont synthétisé dans un rapport de recherche ([462]) les principales propriétés des intersections complètes, qui découlent de la théorie des résidus, fournissant entre autres des preuves simples de résultats classiques.

3.1.6 Classification de singularités de surfaces

Participant : Noëlle Stolfi

N. Stolfi, dans son travail de thèse sous la direction de Michel Merle, étudie une classification des singularités «des ombres et niveaux de gris» d'une surface éclairée. Une surface éclairée est la donnée d'une surface, de deux directions (observateur et lumière), et d'une fonction d'intensité lumineuse sur cette surface. Le modèle étudié est lambertien et opaque, d'où plusieurs courbes apparaissant sur la surface : la courbe d'ombre (ensemble des points où les rayons lumineux sont tangents à la surface), la courbe d'ombre portée (définie par les points qui sont dans le prolongement des points de la courbe d'ombre, le long de la direction des rayons lumineux), les isophotes qui sont les ensembles de points où l'intensité lumineuse est constante. On définit une relation d'équivalence entre surfaces éclairées (deux surfaces éclairées équivalentes ont des courbes d'ombre, d'ombre portée, des isophotes et le contour apparent difféomorphes dans le plan visuel). On donne alors une classification des singularités stables, ainsi qu'une classification des singularités génériques de surfaces éclairées.

3.1.7 Variétés toriques et programmation linéaire entière

Participant : Loïc Pottier

Nous avons présenté à ISSAC en août, et au «Workshop on tests sets in integer programming» à Berlin en octobre, la dernière version de l'algorithme d'Euclide étendu à la dimension n . Les domaines d'application de cet algorithme ont été étendus : il permet maintenant de calculer directement la base de Graver d'un réseau (dont nous avons obtenu une nouvelle borne supérieure), la base de Hilbert d'un cône convexe rationnel, ainsi que les bases de Gröbner d'idéaux toriques et quasi-toriques.

L'implémentation en C de cet algorithme a été améliorée, et des comparaisons avec d'autres algorithmes (Hosten, Macaulay) ont montré son efficacité.

Lors de l'atelier tenu à Berlin, Lovasz a insisté sur le fait que la structure intime des bases de Hilbert reste inconnue, et a formulé une conjecture sur leurs représentations comme réunions de points entiers de polytopes en nombre polynomial. Le sujet est peu étudié encore, aussi est-ce raisonnable d'espérer des résultats nouveaux rapidement dans cette direction, en ce qui nous concerne avec des outils d'algèbre commutative.

Dans le cadre du projet européen FRISCO, ces techniques sont appliquées pour obtenir des bornes supérieures sur le nombre de solutions de systèmes polynomiaux, en particulier sur des exemples industriels.

3.1.8 Problèmes algébriques en géométrie algorithmique

Participant : Ioannis Emiris

Mots-clés : implémentation d'algorithmes géométriques, perturbation symbolique infinitésimale, arithmétique exacte, arithmétique modulaire probabiliste.

Certaines questions fondamentales dans l'implémentation des algorithmes en géométrie combinatoire et algorithmique peuvent être considérées comme des problèmes purement algébriques. Nous étudions deux problèmes importants et souvent complémentaires : l'arithmétique exacte et la singularité des données.

Pour répondre au deuxième problème, une méthode de perturbation infinitésimale, symbolique et efficace a été proposée, avec la collaboration de John Canny (Université de Californie à Berkeley, USA). Cette perturbation des données n'affecte pas un algorithme conçu dans le cadre générique, sous certaines conditions ; par exemple quand on calcule le volume de l'enveloppe convexe d'un ensemble de points. Son efficacité est due au fait qu'elle ne change pas la complexité asymptotique de plusieurs algorithmes, grâce à un calcul algébrique sur des matrices. Pour les applications en robotique et en algèbre géométrique, on est aussi intéressé par la phase de post-traitement, souvent regardée comme le talon d'Achille des perturbations.

Cette perturbation est améliorée et généralisée dans l'article⁵ de Emiris, Canny et Seidel. Une application concrète se trouve dans l'article⁶ de I. Emiris.

Le problème d'arithmétique exacte est fortement lié aux perturbations. Dans le deuxième article, nous avons proposé une méthode efficace et générale qui est fondée sur l'arithmétique modulaire. Il s'agit d'une approche d'arithmétique modulaire probabiliste, qui combine les avantages de l'arithmétique dans un corps fini avec une technique aléatoire, afin d'améliorer l'efficacité du théorème du reste chinois.

⁵ « Efficient Perturbations for Handling Geometric Degeneracies », *Algorithmica, Special Issue on Computational Geometry Applied to Manufacturing*. À paraître.

⁶ « A Complete Implementation for Computing General Dimension Convex Hulls », *International Journal of Computational Geometry & Applications*. À paraître.

3.1.9 Algorithmes de calcul des réductions de Hermite et de Smith

Participant : Roger Marlin

Mots-clés : complexité.

Dans un anneau principal discret à division explicite, les algorithmes de calcul des réductions de Hermite et de Smith sont étudiés avec un souci de contrôle de complexité incluant la taille des objets élémentaires manipulés. Une revue des méthodes classiques a été faite. Certains de ces algorithmes ont été éclaircis, améliorés et généralisés. Ce travail s'effectue en collaboration avec H. Lombardi (de l'université de Besançon) et S. Labhalla (de l'université de Marrakech, Maroc).

3.1.10 Maxima de polynômes généralisés en norme l_p

Participants : Laurent Baratchart, Marc Berthod, Loïc Pottier

Le travail commencé il y a 7 ans, repris l'année dernière, a été soumis cette année au Journal of Convex Analysis. La conjecture concernant la croissance du polynôme lorsqu'on se déplace d'un point de la sphère vers le point de la sphère où la normale est parallèle au gradient du point de départ s'est révélée fausse (contre-exemples simples).

3.2 Logiciel : interfaces, communication, déduction et preuves

3.2.1 Algorithmes de calcul formel certifiés

Participant : Loïc Pottier

Le système Coq a de nombreuses originalités, et celle qui nous intéresse particulièrement est de permettre l'écriture d'algorithmes, de développer leurs preuves de correction, et de les compiler en Caml ou OCaml (modulo quelques adaptations syntaxiques).

Après une initiation au système Coq, qui a consisté à écrire et prouver en Coq une arithmétique des entiers en représentation binaire (addition, multiplication et pgcd), nous avons encadré avec Laurent Théry du projet Croap une étudiante du DEA MDFI de Marseille, dont le sujet était de développer et de prouver en Coq l'arithmétique des polynômes à une variable à coefficients dans un anneau.

Nous avons ensuite repris ce travail, et l'avons poursuivi jusqu'à la division euclidienne. Le code OCaml généré, en utilisant les grands entiers de Caml, est alors 3 à 4 fois plus rapide que Maple. Et il est prouvé correct (modulo la correction de Coq, du compilateur OCaml, etc).

Nous souhaitons poursuivre cette expérience originale de développement d'un noyau d'algorithmes de calcul formel certifiés et efficaces, par exemple avec la transformée de Fourier rapide, la multiplication de Karatsuba, l'algorithme de Buchberger et l'algorithme d'Euclide en dimension n .

Ces travaux ont été exposés lors de la première réunion du groupe de travail «Déduction et calcul formel» du PRC-GDR AMI, à Paris, en juin. Ils s'intègrent aussi dans un réseau européen TMR sur le même thème soumis pour renouvellement (exposé à la réunion de Rome en février).

3.2.2 Architectures logicielles pour le calcul formel, communication

Participants : Stéphane Dalmas, Marc Gaëtano, Stephen Watt

Mots-clés : communication, protocole.

Dans le cadre de notre participation à l'effort international *OpenMath* (dont le but est la définition d'un standard pour la communication d'objets mathématiques, cf. 5.2.2), nous avons défini deux interfaces de bibliothèques, prototypes de futures interfaces OpenMath standard, l'une pour la langage C, l'autre pour *Aldor* (le nouveau langage du système *Axiom*). Deux implémentations correspondantes ont été développées. La bibliothèque C a été intégrée à des versions expérimentales des systèmes *Maple* (avec l'aide de Waterloo Maple Incorporated, Canada) et *Reduce* (avec l'aide de Winfried Neun, du ZIB à

Berlin). La bibliothèque Aldor a été intégrée au logiciel de calcul formel *Bernina* (développé à l'ETH Zurich). Dans le cadre du groupe de travail HTML-Math, dont le but est d'intégrer des expressions mathématiques dans des documents WEB, le projet a aidé à définir les interactions entre *OpenMath* et le protocole HTML Math, cf. 5.2.3

Par ailleurs, le *Central Control* qui a été développé dans ce thème de recherche les années précédentes, a été utilisé par I. Emiris et T. Giordano pour implémenter un algorithme parallèle de calcul du volume mixte (cf. 6.5).

3.2.3 Base de formules

Participants : Stéphane Dalmas, Luc Desouche, Marc Gaëtano, Claude Huchet

Mots-clés : calcul formel, base de formules, base de données déductives.

Les systèmes de calcul formel ont besoin de certaines connaissances de base que l'on peut qualifier de déclaratives, comme par exemple le fait que la dérivée de la fonction sinus est la fonction cosinus ou plus généralement les diverses identités remarquables, des intégrales et des sommations particulières, etc. Ces connaissances sont en général stockées de manière algorithmique et dispersées à l'intérieur des systèmes (reconnaissance de formes particulières dans les algorithmes d'intégration ou de sommation formelle, par exemple). Dans le cadre de la thèse de C. Huchet, nous étudions la conception et la réalisation d'une base de données de relations mathématiques destinée à stocker ces connaissances. À terme, cette base devrait se présenter comme un serveur, accessible depuis n'importe quel système en utilisant le protocole de communication OpenMath.

Une requête à cette base a la forme « sous les conditions C_1, \dots, C_n , la formule P est-elle vraie ? », où P peut contenir des variables particulières qui devront être instanciées par une expression dans la réponse, ce qui permet par exemple, de poser des questions comme « quelle est l'expression (une expression) qui représente la primitive d'une fonction donnée ». La réponse à une requête est multiple et conditionnelle : des conditions de validité supplémentaires peuvent être données dans la réponse.

Les réponses sont obtenues par un processus de déduction original qui combine une unification associative-commutative entre la requête et les formules de la base (il s'agit en fait d'un cas particulier d'unification associative-commutative) et une ou plusieurs étapes qui s'apparentent à la paramodulation : certaines des équations provenant de l'échec de l'unification sont résolues en utilisant la base elle-même ou un algorithme spécialisé.

Un premier prototype (écrit en Standard ML) a été réalisé cette année. Un algorithme de typage des formules (prenant en compte surcharge et inclusion) y a été implémenté par L. Desouche pendant son stage de DEA. Les formules contenues dans la base sont organisées dans une table d'adressage dispersé (indexée par les opérateurs) contenant des « splay trees » (une fonction d'ordre appropriée ayant été définie sur les formules). Nous avons saisi environ 500 formules pour alimenter notre prototype et l'expérimenter.

3.2.4 Interfaces pour le calcul formel, éditeur de formules mathématiques

Participants : Olivier Arsac, Stéphane Dalmas, Marc Gaëtano

Mots-clés : interface homme-machine, éditeur de formules, Lisp.

Les interfaces actuelles des systèmes de calcul formel laissent beaucoup à désirer et de gros efforts semblent devoir être faits pour que le mathématicien ou l'ingénieur y trouve le confort auquel il pourrait prétendre. Nous nous sommes donc engagés dans le projet de conception d'une interface de qualité, tant au niveau de l'affichage des formules que de l'ergonomie. L'expérience de l'utilisation des systèmes de calcul formel que nous avons acquise dans différents contextes (de l'enseignement en premier et second cycles universitaires à la résolution de problèmes industriels) nous amène naturellement à vouloir expérimenter un certain nombre d'idées dans le domaine des interfaces. Ce projet comporte donc une part importante d'expérimentation et l'interface en question doit être conçue pour pouvoir aisément tester nos idées.

La première étape de ce travail est la réalisation d'un composant d'édition de formules mathématiques, *Emath*. *Emath* se présente sous la forme d'un serveur à qui une ou plusieurs applications délèguent la gestion d'une ou plusieurs fenêtres destinées à l'affichage et à l'édition de formules mathématiques. Pour faciliter le développement, l'expérimentation et une grande paramétrisation, l'éditeur contient un petit interprète Lisp (*Klone*, un interprète embarquable développé par l'équipe KOALA). Dans un esprit similaire à celui de l'éditeur de texte *Emacs*, la plupart des fonctionnalités d'*Emath* sont décrites par des fonctions *Klone* et sont donc modifiables et extensibles. Pour garantir une bonne interactivité, l'éditeur utilise un moteur d'affichage incrémental avec cache de « pixmaps » et une bibliothèque de processus légers (*threads*) qui permet, par exemple, que l'édition dans une fenêtre ne bloque pas le réaffichage dans une autre.

La bibliothèque d'édition (en Lisp) réalisée l'année dernière a été améliorée et adaptée au nouveau mode de définition dynamique des opérateurs. Une étude des besoins d'interface en calcul formel a été réalisée. Ceci nous a conduit à proposer un premier modèle d'évaluation d'interfaces homme-machine pour le calcul formel. Un prototype d'interface utilisant *Emath* et intégrant les résultats de notre étude est en cours de réalisation. Olivier Arzac soutiendra sa thèse sur le sujet avant l'été 1997.

3.2.5 Reconnaissance de formules planes

Participants : Stéphane Lavirotte, Loïc Pottier

Mots-clés : grammaire de graphe, réécriture de graphe, analyse de document, reconnaissance de caractères.

Parmi les améliorations possibles de la saisie de formules, on peut envisager l'utilisation de formules déjà imprimées (traitement automatisé pour remplir une base de formules par exemple) ou de les écrire sur une tablette graphique. Le problème à résoudre est alors : *Comment reconstruire une formule* (arbre de syntaxe abstraite) *à partir des données graphiques* (les caractères qui constituent la formule et leurs positions).

Notre approche consiste en l'utilisation d'une grammaire de graphes contextuelle. Nous avons découpé le problème en trois parties bien distinctes :

- la reconnaissance des caractères à l'aide d'un OCR (*Optical Character Recognition*) intégrant des fonctionnalités spécifiques à la reconnaissance des symboles mathématiques,
- la construction d'un graphe à partir des données fournies par l'OCR,
- une grammaire de graphes permettant, par étapes successives de réécriture, de construire l'arbre de syntaxe de la formule.

Durant cette année, nous nous sommes plus particulièrement penchés sur le problème de la grammaire de graphes. L'élimination des ambiguïtés entre les différentes règles de la grammaire se fait grâce à la construction automatisée d'un contexte d'application de la règle. Le contexte est produit par examen successif des recouvrements, deux à deux, des graphes qui constituent les règles. On est amené à résoudre des problèmes d'unification de termes et de graphes.

Une première implémentation pour cette grammaire de graphes a été réalisée en Lisp pour pouvoir expérimenter rapidement les solutions présentées. Nous avons aussi commencé une implémentation du constructeur de graphes. Le problème de la reconnaissance des caractères mathématiques a volontairement été laissé de côté pour cette année, mais nous envisageons un travail sur ce sujet en partant d'un OCR existant que l'on enrichirait pour le dédier plus spécifiquement à la reconnaissance des symboles mathématiques.

3.2.6 Logiciel hyperion

Participant : José Grimm

Le développement du logiciel *hyperion* s'est poursuivi cette année. Il s'agit d'un logiciel d'approximation de fractions rationnelles, conçu en collaboration avec le projet MIAOU. La description du travail est décrite en détail dans le rapport du projet MIAOU.

3.2.7 Catégories et symétries dans les calculs data-parallèles

Participants : Stephen Watt

Travail fait en collaboration avec A. D. Kennedy et R. Edwards du Supercomputer Computations Research Institute à Florida State University.

Les calculs dans les théories de *lattice gauge fields* (chromodynamique quantique) sont extrêmement gourmands en temps de calcul, et sont effectués sur des machines vectorielles ou massivement parallèles. Au fur et à mesure de l'évolution des architectures, les codes existants doivent être adaptés aux nouveaux schémas de communications et à l'allocation de registres. En général, les parties critiques sont remplacées par du code micro- ou macro-assembleur écrit à la main.

Dans le but d'automatiser ce processus, nous étudions quelles catégories décrivent des calculs massivement parallèles, et quelles extensions à A^\sharp et *Foam* (la machine abstraite de A^\sharp) sont nécessaires pour pouvoir implémenter de façon efficace ces calculs.

On peut voir la classe des calculs comme des problèmes dont soit la structure mathématique sous-jacente est une catégorie de fibrés, soit la nature calculatoire sous-jacente est «data-parallel». Les groupes de symétrie du problème calculatoire et du schéma de communication peuvent être représentés indépendamment comme des domaines A^\sharp . Ces domaines deviennent alors les paramètres d'un code générique pour le problème. La relation entre le groupe de symétrie des calculs et des communications sur le matériel donne une structure pour l'allocation automatique de processeurs.

Cette année nous avons étudié le problème de la construction des groupes de symétrie en A^\sharp , et commencé à examiner leur combinaisons via des produits semi-directs.

3.3 Application du calcul formel au calcul numérique

3.3.1 Différentiation automatique

Participants : Frédéric Eyssette, Christèle Faure, André Galligo, José Grimm, Loïc Pottier, Nicole Rostaing-Schmidt, Mohamed Tadjouddine, Vladimir Vyskocil, Stephen Watt

Mots-clés : différenciation automatique, transformation de programme, analyse statique, calcul formel.

Le logiciel de différenciation automatique *Odyssée*, développé dans note projet, dérive des programmes Fortran 77. Des études sont en cours pour accepter d'autres langages sources, comme C, Fortran 90, ou Ada, mais l'essentiel de nos travaux consiste à rendre le code généré par *Odyssée* praticable pour des codes industriels.

La différenciation automatique de code définit deux modes de dérivation : le mode direct (dérivées calculées dans le sens du code initial), et le mode inverse (dérivées calculées dans le sens inverse du code initial).

En mode inverse, le problème est de diminuer le stockage nécessaire au parcours du graphe de calcul à l'envers. En effet, alors qu'en mode direct des codes de 100.000 lignes ont été dérivés (avec *Adi-for*, un concurrent américain d'*Odyssée*), en mode inverse ceci n'a pas encore pu être réalisé. Pour cela plusieurs axes de recherche ont été développés : le remplacement du stockage par du recalcul, la diminution du stockage par l'analyse du code, mais aussi le mélange mode direct-mode inverse.

Le calcul de la jacobienne en mode direct pose lui aussi un problème de taille mémoire. Pour résoudre ce problème nous étudions à la fois un algorithme d'extraction d'informations fines d'un graphe de dépendances de tableau, mais aussi un nouvel algorithme de dérivation.

La composante essentielle de notre travail est l'utilisation de l'analyse du code source permettant l'extraction d'informations nécessaire à la génération de codes plus efficaces. Ce travail est fait en collaboration avec Laurent Hascoët (projet SINUS) et Alain Deutsch (projet A3). Les contrats industriels en cours nécessitent souvent une connaissance approfondie des méthodes d'analyse numérique. Dans ce cadre, des collaborations suivies se sont nouées avec Jean Charles Gilbert (projet PROMATH) et Jean-Michel Malé (projet SINUS).

Le fait qu'Odyssee offre les deux modes de différentiation nous permet de nombreuses expérimentations impossibles à faire dans les autres systèmes. Adifor est un système de différentiation automatique implémentant le mode direct et utilisable en boîte noire. Une comparaison entre les capacités de chacun des deux systèmes, ainsi que la génération de dérivées composites utilisant le mode direct d'Adifor et le mode inverse d'Odyssee est en cours.

Compromis stockage/recalcul en mode inverse. Nous avons poursuivi le travail sur les compromis espace/temps en mode inverse en mettant en œuvre dans Odyssee la stratégie optimale démontrée dans [451]. Il s'agit d'une stratégie optimale en temps de calcul pour différencier en mode inverse une boucle en utilisant un nombre de registres fixé par l'utilisateur. On suppose que cette boucle calcule un vecteur dont la valeur est modifiée à chaque itération, ce qui est par exemple le cas d'un processus d'évolution. D'autre part, nous avons commencé à étudier le cas des boucles emboîtées et des boucles dont le temps de calcul à chaque itération n'est pas constant.

Cette stratégie optimale a été validée sur un exemple fourni par le projet M3N. Elle a également été testée sur un programme résolvant un système d'équations aux dérivées partielles unidimensionnel fourni par EDF.

Une activité de recherche est en cours sur le lien entre la propagation d'erreurs d'arrondi et la différentiation automatique suite à l'étude réalisée pour le CNES l'année dernière.

Architecture logicielle. Les différents traitements nécessaires à la différentiation ont été isolés dans des composants logiciels indépendants : le pré-traitement du code, l'analyse du code, la dérivation, le post-traitement. Cela nous a permis de valider chacun de ces composants et ainsi d'augmenter la fiabilité du logiciel.

Ces composants logiciels ont ensuite été étendus au traitement des appels de fonctions en considérant un appel de fonction $y = f(x_1, \dots, x_n)$ comme un appel de procédure `Call f(x1, ..., xn, y)`, et une définition de fonction comme celle d'une procédure avec un argument supplémentaire.

L'algorithme de remplissage de la base d'informations utilisée par *Odyssee* pour les routines en boîte noire a été modifié de manière à prendre en compte des informations de types sur les arguments. De cette manière, cette base par défaut est semblable à celle que peut écrire un utilisateur lisant le code de l'appel. Cet algorithme est basé sur trois hypothèses : les routines (fonctions ou procédures) en boîte noire ne calculent ni n'utilisent de variables globales, les fonctions en boîte noire ne font pas d'effet de bord sur leurs arguments, les procédures font des effets de bord sur tous leurs arguments. Ces informations par défaut peuvent être modifiées par l'utilisateur.

L'intégration de l'ensemble de ces travaux a donné lieu à la réalisation de la version 1.6 d'*Odyssee*, déposée à l'APP. De plus, *Odyssee* a été traduit de Caml Light en Objective Caml et porté sous Windows 95.

L'interface d'*Odyssee* a été modifiée et améliorée pour tenir compte du changement de langage de commandes et des nouvelles versions d'*Odyssee* ; elle permet maintenant l'impression du graphe d'appel des routines sous forme graphique.

Mixage des deux modes de différentiation. La version actuelle d'Odysée permet de dériver un programme en appliquant le même mode à toutes les unités qui le composent. Des travaux sur le mélange des deux modes dans un même programme sont en cours. Le mode inverse est efficace lorsque le code contient moins de sorties que d'entrées. Il est clair qu'une telle propriété n'est jamais vérifiée simultanément sur toutes les unités d'un programme. Un test est en cours pour définir la validité d'une dérivation mixte dans ce cadre.

De plus, la dérivation en mode inverse d'une boucle est très coûteuse en mémoire, car chaque variable est stockée à chaque tour de boucle pour permettre le parcours de la boucle à l'envers. On peut remarquer que, si une boucle est vectorisable, le résultat du calcul est indépendant du sens de parcours de la boucle. Dans ce cas, il est inutile en mode inverse d'exécuter la boucle à l'envers. Un premier test très prometteur a été réalisé dans le cadre d'un contrat avec ESSILOR. Une boucle vectorisable a été identifiée et dérivée en mode direct alors que le corps de la boucle était dérivé en mode inverse. Les différents morceaux ont été recombinaés à la main dans le reste du code qui était, lui, dérivé en mode inverse.

Ces résultats préliminaires nous amènent à une réflexion plus théorique concernant la validité de chaque mode, ainsi que l'analyse du code permettant l'automatisation d'une stratégie mixte. En effet, il est difficile à un utilisateur de choisir pour chaque unité le mode le plus approprié, il faut donc automatiser ce choix et lui permettre de le remettre en question. Ceci est très important pour rendre le mode inverse praticable sur un grand code et donnera lieu au développement de nouveaux algorithmes.

Analyse de jacobienne creuse. En différentiation automatique, le calcul d'une jacobienne passe par un calcul d'un graphe de dépendance qui exprime chaque variable calculée comme une fonction des entrées du programme à dériver.

Actuellement, Odysée calcule ce graphe en raisonnant sur les symboles des variables, donc « un tableau a dépend d'un tableau b » signifie que chaque case de a dépend de toutes les cases de b . Cette interprétation maximaliste engendre des calculs inutiles et un accroissement de l'occupation mémoire.

Pour augmenter l'efficacité du code généré, nous cherchons à affiner cette analyse en appliquant des méthodes utilisées en parallélisation. Nous développons un algorithme de calcul des dépendances « fines » dans Foresys/Partita dédié à Odysée. Cet algorithme utilise le graphe de flot de données calculé par Partita. Ce dernier nous fournit l'ordre d'exécution séquentielle du code et donc un certain nombre d'informations sur l'enchaînement des opérations atomiques (lecture/écriture de variables, opérations arithmétiques, contrôle d'une instruction de boucle, etc.) effectuées dans le programme.

À partir de ce graphe nous cherchons, pour chaque case de tableau calculée, toutes les cases des tableaux ayant servi à ce calcul, ce qui revient à faire une analyse sur les zones mémoires et non sur les variables.

Odysée utilisera cette information pour générer une jacobienne plus efficace et moins gourmande en place mémoire. Il reste pour cela à définir un algorithme de dérivation qui utilise ces informations de façon optimale.

3.3.2 Outils pour la modélisation et la simulation

Participant : Yves Papegay

Mots-clés : optique, électronique, modélisation, génération de code, simulation.

La modélisation, puis la simulation, du comportement d'un système physique est une composante importante de nombreux processus industriels (de la conception à la fabrication).

Le processus de modélisation-simulation se décompose naturellement en trois étapes :

1. La formalisation du problème physique : à partir de la description physique du problème (ou de l'objet à modéliser) et de la théorie physique s'appliquant à ce problème, on définit les paramètres qui le caractérisent, ainsi qu'un modèle, c'est-à-dire des équations — algébriques et différentielles — qui gouvernent ces paramètres et leurs évolutions.

2. La résolution du problème mathématique : à partir de l'ensemble des équations qui gouvernent les paramètres du problème, d'une part, et en choisissant parmi les paramètres quels sont ceux supposés connus (les entrées) et ceux que l'on veut déterminer (les sorties) d'autre part, on détermine un algorithme de simulation qui calculera ces sorties en fonction des entrées.
3. L'obtention de résultats numériques : à partir de l'algorithme de simulation déterminé précédemment on engendre un programme de simulation qui, prenant un ensemble de données numériques caractérisant les propriétés du système étudié, rendra un autre ensemble de données numériques représentant les caractéristiques du comportement que l'on veut simuler.

Dans la première étape, ce sont les fonctionnalités de manipulation d'expressions symboliques des systèmes de calcul formel qui sont intéressantes à utiliser : il s'agit de disposer d'un langage de description gérant différents niveaux de détails et qui permette de stocker et de réutiliser les équations représentant la connaissance physique ou les modèles étudiés dans des bibliothèques. Au-delà de l'implémentation d'un tel langage, se pose le problème d'associer à une équation un certain nombre d'inéquations qui définissent son domaine de validité, la théorie se basant souvent sur des approximations qui ne sont valides que dans un domaine restreint. La combinaison de ces inéquations pour définir, même interactivement, un modèle cohérent, nécessite de développer dans les systèmes de calcul formel un ensemble d'opérations sur les inéquations qui n'existent pas actuellement, et, à plus long terme de s'intéresser à des cas particuliers du problème général de la résolution et de la simplification d'un ensemble de contraintes algébriques.

Dans la seconde étape, les problèmes mathématiques qui se posent dépendent essentiellement de la nature des équations à résoudre pour déterminer un algorithme de calcul. Ainsi, cela nécessite de s'intéresser dans des cas particuliers à la résolution de systèmes d'équations polynomiales (mécanique des corps articulés) ou dans d'autres cas aux problèmes de parcours et de recherche de sous-ensembles connexes minimaux dans des hypergraphes (conception d'instruments optiques) puis d'implémenter les outils correspondants. Il faut aussi, à ce niveau, pouvoir traiter les problèmes qui n'ont de solutions qu'à l'aide de méthodes numériques, et donc pouvoir considérer un programme de calcul d'une expression — qu'il soit sous la forme d'une description symbolique d'un algorithme, ou sous la forme d'un véritable code numérique (Fortran ou C par exemple) — comme une représentation de cette expression, au même titre qu'une formule symbolique, ou qu'une valeur numérique. Il est donc intéressant de développer, dans les systèmes de calcul formel, un ensemble d'opérations qui utilisent au mieux cette représentation et la possibilité de changer de représentation pour améliorer les performances et les temps de calcul. Un gros travail reste à fournir dans ce domaine, qui est fortement connexe avec l'activité de différentiation automatique.

Dans la troisième étape, il s'agit de pouvoir transformer des algorithmes de calcul en des programmes de simulations numériques performants, et ce sont les fonctionnalités de génération de code des systèmes de calcul formel qui sont utilisées. Si la possibilité de traduire des expressions symboliques en un code qui les calcule est déjà existante, un travail important reste à fournir pour obtenir des codes de calcul efficaces. Un autre champ d'investigation important consiste en la possibilité de faire communiquer des outils de calcul formel entre eux et avec d'autres logiciels de calculs, que ce soit des bibliothèques de programmes numériques ou des logiciels d'optimisation ou de visualisation. Ce travail se rapproche des recherches déjà effectuées dans le projet autour du Central Control.

Après avoir analysé ces questions dans le cas particulier de la mécanique des solides polyarticulés dans sa thèse, Yves Papegay s'intéresse depuis 1994 au cas de la calibration d'instruments optiques embarqués sur satellites (en collaboration avec l'aérospatiale), et depuis 1995 à celui de la conception de circuits intégrés (en collaboration avec Siemens, Munich).

Cette année, le travail réalisé en optique, qui a été en grande partie financé par l'aérospatiale a consisté en la production d'un logiciel écrit en *Maple* dans le cadre du stage post-doctoral de Jérôme De Sousa (cf. 4.3) et en l'ajout de fonctionnalités de génération de code numérique et de communication avec les bibliothèques numériques NAG et IMSL au prototype *circe* développé l'an passé. Ce prototype construit, à partir d'un ensemble d'équations, un hypergraphe représentant les dépendances des paramètres à travers ces équations, et permet de parcourir cet hypergraphe pour exprimer une performance en fonction

d'un ensemble de paramètres fixés. Il produit ensuite automatiquement un code C de simulation numérique spécialisé.

Le travail réalisé en électronique en collaboration avec Siemens consiste en l'utilisation de *Maple* pour la production semi-automatique du code Fortran de simulation du comportement de certains transistors. Cette simulation nécessite aussi le calcul de la dérivée de certains paramètres de sortie par rapport aux paramètres d'entrée. De nombreux tests ont été réalisés afin de comparer les diverses méthodes de génération du code de calcul simultané de la simulation et des dérivées (calcul symbolique explicite des dérivées, différentiation automatique du code *Maple*, ou différentiation automatique du code numérique engendré).

4 Actions industrielles

4.1 Différentiation automatique

Participants : Frédéric Eyssette, Christèle Faure, André Galligo, José Grimm, Loïc Pottier, Nicole Rostaing-Schmidt, Mohamed Tadjouddine, Vladimir Vyskocil, Stephen Watt, Jean Charles Gilbert, Jean-Michel Malé

4.1.1 GENIE

Le projet participe à l'action GENIE, contrat entre l'Inria et Dassault Aviation. L'utilisation du mode inverse de la différentiation automatique avait révélé la nécessité d'un compromis entre stockage et recalcul des valeurs de la fonction à dériver. Nous avons donc mis en œuvre dans Odyssée une stratégie du type « diviser pour régner ». Cette stratégie a été testée sur un code calculant la traînée autour d'un profil défini par un maillage, fourni par le projet M3N. La description de cette stratégie, ainsi que les résultats en place mémoire/temps de calcul obtenus sur cet exemple, sont exposés dans le rapport final du contrat.

4.1.2 EDF

Ce contrat a pour objectif de valider la différentiation automatique en mode inverse sur un code industriel développé à EDF-DER. L'algorithme de dérivation optimale en mode inverse (produit espace-temps) d'une boucle implémenté dans le cadre du contrat GENIE sur une unité seule a été étendu à la dérivation d'une boucle dans un programme. Cette nouvelle implémentation nous a permis de valider cette approche sur une maquette monodimensionnelle du code tridimensionnel THYC. De plus la dérivation du code THYC en mode direct (sans modification du source) est en cours d'achèvement, ce qui sera le record du plus grand code dérivé automatiquement.

4.1.3 ESSILOR

Le cadre de cette étude est l'optimisation surfacique de verres de lunettes. Le problème est de trouver l'équation d'une surface dont on se donne certains paramètres : les cartes de courbure et éventuellement la carte des gradients de courbure. L'objectif de cette étude était d'évaluer l'apport de la différentiation automatique ainsi que de méthodes d'optimisation de type quasi-Newton en termes de vitesse de convergence, de précision sur la solution obtenue et de contrôle du processus d'optimisation. Les outils utilisés étaient d'une part Odyssée et d'autre part des modules d'optimisation de la bibliothèque Modulopt (développée dans le projet PROMATH).

Cette étude a été réalisée par l'Inria à partir de codes fournis par la société ESSILOR et a prouvé que les temps d'optimisation par N1QN3 (une procédure de modulopt) utilisant la dérivation en mode inverse ont été divisés par un facteur variant entre 19 et 76 pour un seuil choisi suivant le jeu de données.

4.2 CNES

Participant : José Grimm

La collaboration avec le CNES s'est poursuivie avec la signature, avec un an de retard, d'un nouveau contrat, en commun avec le projet MIAOU, dans le but d'intégrer de nouveaux algorithmes d'identification de filtres hyperfréquences dans le logiciel *hyperion*.

4.3 Développement de Circe

Participants : Jérôme De Sousa, Yves Papegay

CIRCE est un logiciel écrit en *Maple*, développé sur une commande de l'aérospatiale par Jérôme De Sousa, dans le cadre d'un stage post-doctoral effectué sous la responsabilité scientifique d'Yves Papegay. CIRCE génère automatiquement des bilans de performances d'un système optique sous la forme de documents *excel* à partir d'une description symbolique des équations modélisant ces performances.

5 Actions nationales et internationales

5.1 Actions nationales

Le groupe de travail « Formes & Formules » commun avec le laboratoire de mathématiques de l'université de Nice a été organisé par M. Elkadi et B. Mourrain. Le thème de cette année était *les intersections complètes*. B. Mourrain était également responsable des séminaires « Math-Info » se déroulant à Sophia Antipolis et plus orientés vers les applications du calcul formel⁷.

5.1.1 PRC AMI

SAFIR est une équipe du PRC-GDR AMI (Algorithmes, Modèles et Infographie)⁸, créé au début de cette année. Nous participons à deux groupes de travail d'AMI : « Résolution effective des systèmes algébriques et applications à la robotique » et « Dédution et calcul formel ».

5.2 Actions internationales

5.2.1 SNAP

I. Emiris, A. Galligo, B. Mourrain et S. Watt (avec l'aide de R. Corless et F. Limouzis) ont organisé l'atelier de travail SNAP, sur les méthodes numériques et formelles pour les algèbres de polynômes. Lorsque les coefficients des polynômes sont connus uniquement de manière approximative, les algorithmes d'algèbre polynomiale exacte ne s'appliquent pas. Les participants ont proposé de nouvelles méthodes pour le calcul polynomial dans ce contexte, par exemple les pgcd de polynômes, la factorisation, les méthodes matricielles, etc. Une sélection des meilleurs articles présentés seront publiés dans un numéro spécial du Journal of Symbolic Computation.

5.2.2 OpenMath

Le projet SAFIR participe à l'effort international *OpenMath* (cf. 3.2.2). Ce consortium rassemble des industriels (IBM, NAG, Waterloo Maple Incorporated) et des universitaires. Le but d'OpenMath est la définition d'un protocole de communication d'expressions mathématiques entre applications, ces

⁷voir <http://www.inria.fr/safir/seminaires.html>

⁸<http://www.univ-mlv.fr/ami/>

applications pouvant être des systèmes de calcul formel, des interfaces graphiques, des bases de données ou des systèmes de traitement de textes. S. Watt est membre du comité de direction d'OpenMath.

5.2.3 Groupe de travail W3C HTML-Math

S. Watt et S. Dalmas sont membres du groupe de travail HTML-Math du consortium World Wide Web. Le but de ce groupe de travail est de proposer une extension de HTML pour l'affichage des formules mathématiques qui puisse être utilisée pour l'enseignement et l'édition scientifique qui puisse permettre l'interaction avec d'autres applications (comme des systèmes de calcul formel), la synthèse vocale et une conversion relativement automatique des formats existants (principalement TeX).

5.2.4 AD-CAPE

Le projet européen AD-CAPE a pour but de promouvoir l'utilisation de dérivées analytiques dans les codes industriels de génie des procédés. L'équipe de différentiation automatique de SAFIR participe à la mise en place de ce projet, coordonné par SIMULOG et en partenariat avec ELF et BP. Ce projet a été soumis pour cofinancement à la commission des communautés européennes dans le cadre du programme ESPRIT.

5.3 Le réseau SAC

Avec le GDR Médecis, nous gérons le nœud français du réseau SAC. Cette année a vu la mise en place du serveur Web «CAIN» sur le calcul formel en Europe⁹. Sur ce serveur, nous maintenons une liste d'annonces de conférences autour du calcul formel. Le réseau soutient l'organisation de séminaires thématiques.

5.4 Le Projet Frisco

Nous participons au projet européen Frisco (a Framework for Integrated Symbolic/Numeric Computation, ESPRIT-LTR 21.024) qui a débuté au mois de mars 1996. Frisco est dédié à l'étude et la réalisation de composants logiciels pour la résolution de systèmes polynomiaux, et à leurs applications aux problèmes industriels. Ce projet reprend en partie les travaux réalisés au sein du projet PoSSo et conjugue les efforts de différents partenaires : la société NAG (The Numerical Algorithms Group, UK), l'université de Pise (Italie), l'université de Santander (Espagne), l'université de Rennes et le projet SAFIR. Notre projet intervient dans les parties intégration, algorithmique et applications. S. Watt est le coordinateur principal de ce projet

5.5 ISSAC

La conférence ISSAC (International Symposium on Symbolic and Algebraic Computation) est l'une des plus importantes conférences annuelles dans le domaine du calcul formel. Elle a eu lieu cette année à Zurich (Suisse). Le projet SAFIR y a participé au moyen de contributions, et S. Watt en a été le président du *Steering Commitee*.

⁹<http://www.can.nl/>

6 Diffusion des résultats

6.1 Enseignement

6.1.1 Cours

- *DEA de Mathématiques de Nice* :
S. Dalmas (10h), I. Emiris (16h), C. Faure (3h), B. Mourrain (16h), L. Pottier (16h), S. Watt (25h).
- *DEA MDFI de Marseille* :
I. Emiris, A. Galligo, B. Mourrain, L. Pottier (16h).
- *DEA X-ENS Paris* : B. Mourrain (12h).
- *DEA d'Informatique de Nice* : S. Dalmas (12h).
- *Maîtrise Math-Info de Nice* : B. Mourrain, Y. Papegay, L. Pottier.
- *École d'été CEA-Inria-EDF* : (25h) C. Faure, A. Galligo, E. Hassold.
- *Cours Inria, Techniques de développement de codes numériques* : (4h) C. Faure, Y. Papegay.
- *École de calcul formel pour les professeurs de classes préparatoires, Tunis, Nice* : (2x20h.) B. Mourrain, Y. Papegay.

6.1.2 Thèses

1. Olivier Arzac, « Interfaces homme-machine pour le calcul formel », UNSA.
2. Didier Bondyfalat, « Démonstration automatique en géométrie et application à la vision »
3. Eric Hassold, « Différentiation automatique appliquée à l'optimisation » UNSA.
4. Claude Huchet, « Une base de donnée déductives pour des formules mathématiques », UNSA.
5. Stéphane Laviotte, « Reconnaissance de formules planes appliquée aux systèmes de calcul formel », UNSA.
6. Noëlle Stolfi, « Ombres et niveaux de gris d'une surface éclairée : classification de leurs singularités », UNSA, thèse soutenue le 19 décembre 1996.
7. Mohamed Tadjouddine, « Différentiation automatique de programmes de conception optimale », UNSA.
8. Vladimir Vyskocil, « La différentiation automatique appliquée à l'analyse de la propagation d'erreurs d'arrondi », UNSA.

S. Watt a participé à deux jurys de thèses, comme président et comme rapporteur.

6.1.3 Stages

1. N. Arnous, « Visualisation scientifique progressive », stage de DEA informatique, ESSI.
2. Y. Chicha, « Animations d'algorithmes de bases de Groebner en A^{\sharp} », stage de DEA informatique, ESSI.
3. L. Desouche, « Analyse et typage d'expressions mathématiques pour une base de formules », du 1^{er} novembre 95 au 31 juillet 96, DEA de Mathématiques, UNSA.

4. G. Dos Reis, «Implémentation efficace pour calculer les zéros d'un polynôme», juillet-septembre, stage de 1^{ère} année, ENS Ulm.
5. S. Fantino, «Application de la différentiation automatique à l'optique», stage de DEA de mathématiques, UNSA.
6. T. Giordano, «Implémentation distribuée pour le calcul du volume mixte», juillet-août, DEA de Mathématiques, UNSA.
7. M. Guillaume, «Bases de Groebner booléennes», juillet-août, DEA de Mathématiques, UNSA.
8. F. Livigni, «Algèbre linéaire pour les polynômes», avril-septembre, stage ESSI 3^{ème} année.
9. F. Tournois, «Calcul de Bézoutien par FFT», juillet-septembre, stage de 1^{ère} année, ENS Ulm.

6.2 Participation à des conférences et colloques

Ioannis Emiris a donné des communications à la conférence nationale en géométrie algorithmique à Le Bessat, à MEGA'96 à Eindhoven en juin, au congrès international d'analyse non-linéaire à Athènes, à SNAP en juillet, et au «Workshop on Toeplitz matrices: Structure, Algorithms and Applications» à Cortona. Il a également fait un exposé à l'atelier sur les polytopes de Newton à Limoges et a assisté au «Workshop on Algorithmic Foundations of Robotics»(WAFR-96) en juillet à Toulouse.

C. Faure, E. Hassold et N. Rostaing-Schmidt ont participé au «Second workshop on Computational Differentiation».

C. Faure, E. Hassold et V. Vyskocil ont participé à ECCOMAS'96.

Marc Gaëtano, Stéphane Dalmas et Stephen Watt ont assisté aux réunions *OpenMath* à Zurich en juillet et à Dublin en décembre. Ils y ont présenté les prototypes réalisés (cf. 3.2.2).

Marc Gaëtano a fait un exposé de présentation du Central Control à la conférence *Interval Arithmetic and Symbolic Computation* (université fédérale du Pernambuco, Recife, Brésil) et au Colloque africain sur la recherche en informatique (Libreville, Gabon).

Claude Huchet et Marc Gaëtano ont participé à la conférence DISCO 96 (Design and Implementation of Symbolic Computation Systems) et à la conférence IMACS-ACA (Applications of Computer Algebra), où ils ont présenté leurs travaux sur les bases de formules.

Bernard Mourrain a présenté ses travaux à la conférence MEGA'96 (Eindhoven, Hollande), à SNAP'96 (Sophia Antipolis), à la conférence IMACS (Linz, Autriche), à l'atelier sur les matrices structurées (Cortona, Italie). Il a également participé aux journées de GDR AMI à La Buissonnière et aux journées sur l'élimination à Luminy.

Yves Papegay a participé au colloque annuel de l'International Society for Optical Engineering où ses travaux sur l'utilisation du calcul formel pour la modélisation d'instruments optiques ont été présentés.

Yves Papegay s'est rendu au Japon en août. Il a fait deux exposés au 2nd Asian Symposium for Computational Mathematics à Kobe. Puis il a séjourné quelques jours à ElectroTechnical Laboratory à Tsukuba, invité par H. Hirukawa.

Yves Papegay a présenté une communication en septembre à Lausanne (Suisse) à la conférence AIHENP 96.

Loïc Pottier et Yves Papegay ont participé à la réunion préparatoire du projet TMR «Math Assistant» à Rome (Italie) en mars. Loïc Pottier y a présenté des applications de Coq en calcul formel.

Loïc Pottier a fait un exposé «Expériences en algorithmique avec Coq» et Stephen Watt a présenté le système A[#] aux journées du groupe de travail AMI «Déduction et calcul formel» à Paris en juin.

Loïc Pottier a fait une conférence invitée à l'atelier «Tests sets in integer programming» à Berlin (Allemagne) en octobre, a présenté une communication à ISSAC à Zurich (Suisse) en juillet, et a participé aux journées sur la programmation linéaire en nombres entiers à Rocquencourt en octobre.

6.3 Organisation de colloques et de cours

N. Rostaing-Schmidt a fait partie du comité de programme du «Second workshop on Computational Differentiation», qui a eu lieu du 12 au 15 février à Santa Fe au Nouveau Mexique (USA).

Des membres du projet (I. Emiris, A. Galligo, B. Mourrain et S. Watt) ont organisé le premier «Workshop on Symbolic-Numeric Algebra for Polynomials». Les actes et de plus amples informations sont disponibles à l'URL : <http://www.inria.fr/safir/MEETING/snap.html>.

6.4 Conférences invitées

S. Watt a fait les conférences invitées suivantes :

Motivations in the Design of Aldor, Computer Algebra Workshop, INFORMATIK'96, Klagenfurt, Autriche,

The A[#] Programming Language: Reconstructing an Algebra System from Elementary Particles, CERN, Geneva, Suisse,

Systems Issues in Computer Algebra – Past, Present and Future, Maths for Information Technology national meeting, Computing Laboratory, Oxford, Angleterre,

The A[#] Programming Language: Reconstructing an Algebra System from Elementary Particles, GMD, Bonn, Allemagne,

The A[#] Programming Language: Reconstructing an Algebra System from Elementary Particles, German Axiom Users' Group Meeting, U. Karlsruhe, Allemagne,

Programming Language Support For Memory Management, Workshop on Computer Algebra Software, Schloss Dagstuhl, Wadern, Allemagne.

6.5 Diffusion de produits

Odyssée a été fourni à Dassault-Aviation, à EDF-DER dans le cadre de contrats et au CEMEF (école des Mines de Paris), à l'université de Limoges, à l'INSA de Lyon, à Argonne National Laboratory (USA), Marquette University (USA) et à l'Ifremer dans le cadre de mises à disposition gratuites.

T. Giordano et I. Emiris ont conçu et implémenté un programme distribué qui calcule le volume mixte d'un ensemble de polytopes en dimension arbitraire. (accessible par <ftp://www.inria.fr/safir>) ou par les pages Web du projet.

Le programme *bastat* écrit par L. Pottier est régulièrement testé sur le Web sur des exemples variés, et téléchargé.

Mise à disposition publique du logiciel ASAP, motivée par notre participation au consortium OpenMath.

6.6 Autres

I. Emiris a passé deux semaines en novembre au « University of North Carolina at Raleigh », invité de Dinesh Manocha, afin de collaborer avec lui et Aaron Wallack sur la résolution de systèmes polynomiaux par des méthodes matricielles. Il a également passé une semaine en novembre 1996 au « University of California at Berkeley », invité de John Canny, sur le même sujet de recherche.

S. Watt est président de SIGSAM (the ACM special interest group in Symbolic and Algebraic Computation).

José Grimm a été responsable local de la rédaction du rapport d'activité 1996.

7 Publications

Thèses

- [437] N. STOLFI, *Ombres et niveaux de gris d'une surface éclairée : Classification de leurs singularités*, thèse de doctorat, Université de Nice-Sophia Antipolis, décembre 1996.

Articles et chapitres de livre

- [438] P. COMON, B. MOURRAIN, «Decomposition of Quantics in sums of power of linear forms», *Signal Processing, Special issue on High-Order Statistics* 53, 2, 1996.
- [439] I. Z. EMIRIS, A. GALLIGO, «Review of *Polynomial and Matrix Computations. Volume 1: Fundamental Algorithms*, by D. Bini and V. Pan, Birkhäuser, Boston, 1994», *SIGSAM Bulletin*, 1996.
- [440] I. Z. EMIRIS, «On the Complexity of Sparse Elimination», *J. Complexity* 12, 1996, p. 134–166.
- [441] S. LABHALLA, H. LOMBARDI, R. MARLIN, «Algorithmes de calcul de la réduction de Hermite d'une matrice à coefficients polynomiaux», *Theoretical Computer Science* 161, 1996, p. 69–92.

Communications à des congrès, colloques, etc.

- [442] J. P. CARDINAL, B. MOURRAIN, «Algebraic approach of residues and Applications», in : *Proc. AMS-SIAM Summer Seminar on Math. of Numerical Analysis*, J. Renegar, M. Shub, S. Smale (réd.), Park City, Utah, juillet 1996, <ftp://www.inria.fr/safir/9601-BM-residues-utah.ps.gz>.
- [443] S. DALMAS, M. GAËTANO, C. HUCHET, «A Deductive Database for Mathematical formulas», in : *Design and Implementation of Symbolic Computation Systems*, LNCS, 1128, Springer Verlag, p. 287–296, 1996.
- [444] S. DALMAS, M. GAËTANO, A. SAUSSE, «Un logiciel pour le calcul formel distribué», in : *Actes du troisième Colloque Africain sur la Recherche Informatique*, P. Moukeli (réd.), ORSTOM éditions, p. 380–389, 1996.
- [445] S. DALMAS, M. GAËTANO, «Making Systems Communicate and Cooperate: the Central Control Approach», in : *Design and Implementation of Symbolic Computation Systems*, J. Calmet, C. Limongelli (réd.), LNCS, 1128, Springer Verlag, p. 308–319, 1996.
- [446] C. DUVAL, P. EYRHARD, C. FAURE, J. C. GILBERT, «Application of the Automatic Differentiation Tool Odyssee to a system of thermohydraulic equations», in : *Proc. of ECCOMAS'96*, J.-A. Désidéri, P. Le Tallec (réd.), *Numerical Methods in Engineering'96*, John Wiley and Sons, p. 795–802, septembre 1996.
- [447] I. Z. EMIRIS, A. GALLIGO, H. LOMBARDI, «Certified Approximate Univariate GCDs», in : *Proc. 4th Intern. Symp. Effective Methods in Algebraic Geometry (MEGA)*, juin 1996.
- [448] I. Z. EMIRIS, A. GALLIGO, H. LOMBARDI, «Numerical Univariate Polynomial GCD», in : *Proc. AMS-SIAM Summer Seminar on Math. of Numerical Analysis, July 1995, Park City, Utah*, J. Renegar, M. Shub, S. Smale (réd.), *Lectures in Applied Math.*, 32, p. 323–343, 1996.
- [449] F. EYSSETTE, V. VYSKOCIL, «Roundoff Errors Propagation Analysis with Automatic Differentiation of Fortran 77 Code», in : *Numerical Methods in Engineering'96*, J.-A. Désidéri, P. Le Tallec, E. O. nate, J. Périaux, E. Stein (réd.), John Wiley and Sons, p. 1018–1021, septembre 1996.
- [450] C. FAURE, «Splitting of Algebraic Expressions for Automatic Differentiation.», in : *Computational Differentiation : Techniques, Applications, and Tools*, M. Berz, C. Bischof, G. Corliss, A. Griewank (réd.), SIAM, Philadelphia, Penn., p. 117–128, 1996.
- [451] J. GRIMM, L. POTTIER, N. ROSTAING-SCHMIDT, «Optimal time and minimum space-time product for reversing a certain class of programs», in : *Computational Differentiation: Applications, Techniques, and Tools*, Berz, Bischof, Corliss, Griewank (réd.), SIAM, p. 95–106, 1996. disponible également sous forme de rapport de recherche INRIA, <http://www.inria.fr/RRRT/RR-2794.html>.

- [452] E. HASSOLD, «Automatic Differentiation applied to a Nonsmooth Optimization Problem», in : *Numerical Methods in Engineering'96*, J.-A. Désidéri, P. Le Tallec, E. O. nate, J. Périaux, E. Stein (réd.), John Wiley and Sons, p. 835–841, septembre 1996.
- [453] J.-M. MALÉ, N. ROSTAING-SCHMIDT, N. MARCO, «Automatic Differentiation: an Application to Optimum Shape Design in Aeronautics», in : *Minisymposia of ECCOMAS 96*, J. A. Désidéri, C. Hirsch, P. Le Tallec, E. Oñate, M. Pandolfi, J. Périaux, E. Stein (réd.), John Wiley & Sons, Ltd, septembre 1996.
- [454] B. MOHAMMADI, J.-M. MALÉ, N. ROSTAING-SCHMIDT, «Automatic Differentiation in Direct and Reverse Modes: Application to Optimum Shapes Design in Fluid Mechanics», in : *Computational Differentiation : Techniques, Applications, and Tools*, M. Berz, C. Bischof, G. Corliss, A. Griewank (réd.), SIAM, 1996.
- [455] B. MOURRAIN, «Isolated points, duality and residues», in : *Proc. 4th Intern. Symp. Effective Methods in Algebraic Geometry (MEGA), June 1996*, 1996. (to appear), <ftp://www.inria.fr/safir/9604-BM-invsyst.ps.gz>.
- [456] Y. PAPEGAY, S. DALMAS, «QUATERMAN vs. QUATERNICA: a Comparative Implementation of Quaternions in Maple and Mathematica», in : *Proceedings of the 2nd ASCM*, H. Kobayashi (réd.), Scientists Inc., p. 113–120, Tokyo, août 1996.
- [457] Y. PAPEGAY, W. KLEIN, «Automatic Generation of a MOS Transistor Simulation Model with Maple V», in : *Proceedings of the AIHENP 96 Conference*, Lausanne, 1996.
- [458] Y. PAPEGAY, S. PHILOREAU, «Design and Performances Analysis of Complex Optical Devices using Symbolic Computation», in : *Proceedings of the 2nd ASCM*, H. Kobayashi (réd.), Scientists Inc., p. 55–61, Tokyo, août 1996.
- [459] S. PHILOREAU, D. MIRAS, Y. PAPEGAY, D. SIMEONI, «IASI: A new way to manage budget performances», in : *Proceedings of the SPIE 96 Annual Meeting, volume 2817*, Infrared Spaceborne Remote Sensing Conference, SPIE, août 1996.
- [460] L. POTTIER, «Euclidean algorithm in dimension n, applications to points in cones, toric ideals, and integer programming», in : *Workshop on test sets in integer programming, ZIB Berlin*, octobre 1996.
- [461] L. POTTIER, «Euclidean's algorithm in dimension n», in : *Proc. of ISSAC 96, Zurich*, ACM Press, juillet 1996.

Rapports de recherche et publications internes

- [462] M. ELKADI, B. MOURRAIN, «Approche Effective des Résidus Algébriques», *Rapport de Recherche n°2884*, INRIA, 1996.
- [463] F. EYSSETTE, J. C. GILBERT, C. FAURE, N. ROSTAING-SCHMIDT, «Applicabilité de la différentiation automatique à un système d'équations aux dérivées partielles régissant les phénomènes thermohydrauliques dans un tube chauffant», *Rapport de recherche n°2795*, INRIA, février 1996, <http://www.inria.fr/RRRT/RR-2795.html>.
- [464] V. Y. PAN, «Numerical Computation of a Polynomial GCD and Extensions», *rapport de recherche n°2969*, INRIA, Sophia-Antipolis, France, août 1996, <http://www.inria.fr/RRRT/RR-2969.html>.
- [465] V. Y. PAN, «On Approximating Complex Polynomial Zeros: Modified Quadtree (Weyl's) Construction and Improved Newton's Iteration», *rapport de recherche n°2894*, INRIA, Sophia-Antipolis, France, mai 1996, <http://www.inria.fr/RRRT/RR-2894.html>.
- [466] A. SAUSSE, «Étude par le calcul formel d'une question issue du problème à quatre corps», *Rapport de Recherche n°2756*, INRIA, décembre 1995.

8 Abstract

The research of the SAFIR project concerns three facets of computer algebra:

- algorithms in algebra and geometry, where monomials and polynomials play a central role,
- mathematical software: systems, interfaces, communication, program transformation, deductions and proofs,
- application to numeric computation, with the automatic differentiation of programs, modelling and simulation of physical systems.

Each of these aspects involves, to varying degrees, research toward new theoretical results, the development of software prototypes, and application and validation in concrete domains. These are not three activities – each interacts with the others as part of a multidisciplinary field.

SAFIR is a common project between the University of Nice-Sophia Antipolis, INRIA, and the CNRS (the Dieudonné Mathematical Lab, URA 168, and the I3S Lab, URA 1376), so education in areas of computer algebra is an important activity for many members of the project. This takes the form of graduate and undergraduate programs, as well as short courses for professionals.

Finally, the project participates in the scientific community at the national, European, and international levels through groups such as: PRC-GDR AMI, GDR Medicis and GDR Meteorologie; the SAC network, SOL network and FRISCO consortium; program committees for meetings in the field, as well as organization and participation in specialist workshops.

The year 1996 was marked by the integration of Stephen Watt, a specialist in systems for computer algebra, and the visit of Victor Pan, a specialist in algebraic algorithms based on matrices.