Rapport INRIA 1994 — Programme 2

Systèmes Algébriques Formels pour l'Industrie et la Recherche

PROJET SAFIR

3 mai 1995

PROJET SAFIR

Systèmes Algébriques Formels pour l'Industrie et la Recherche

Localisation : *Sophia-Antipolis*

Mots-clés : algèbre (1, 12, 17), algèbre linéaire (23), algorithme (1, 9), analyse de programme (21), automatique (11, 23), base de données déductives (5), base de formules (5), base de Gröbner (13, 15), calcul formel (1, 5, 9, 11), calcul réparti (4, 17), calcul scientifique (21), calcul symbolique (1), complexité (16, 17), courbe (10, 18), différentiation automatique (1, 16, 21), élimination (13), expression contrainte (5), forme linéaire (16), géométrie (1, 12), géométrie algébrique (1, 12, 17, 18), géométrie différentielle (19), grammaire (8), interface graphique (7), interface homme-machine (7), invariant géométrique (12), langage de programmation (9), Lisp (7, 11), mécanique des solides (1, 20), modélisation (1), optimisation (23), optique (20), programmation en nombres entiers (13), reconnaissance des formes (8), réseau arithmétique (15), robotique (12), Scheme (4), simulation (20), singularité (18), surface (10), système polynomial (12, 13), vision par ordinateur (12, 19), visualisation (10).

1 Composition de l'équipe

Responsable scientifique

Jacques Morgenstern, Professeur à l'université de Nice-Sophia Antipolis, URA 1376, jusqu'en avril

Loïc Pottier, CR INRIA, depuis septembre

Coordinateur scientifique

André Galligo, Professeur, URA 168

Responsable permanent

José Grimm, CR INRIA

Secrétaire

France Limouzis, TR INRIA

Personnel INRIA

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

Personnel université de Nice-Sophia Antipolis

Marc Gaëtano, Maître de conférences, URA 1376
Frédéric Eyssette, Maître de conférences, URA 168
Nathalie Eyssette, ATER,URA 1376
Norbert Kajler, ATER, URA 1376, à partir de septembre
Roger Marlin, Professeur, URA 168
Michel Merle, Professeur, URA 168
Nicole Rostaing-Schmidt, ATER, URA 168

Conseiller scientifique

Joos Heinz, Professeur à l'université de Santander (Espagne)

Ingénieurs experts

Christèle Faure, à partir de mars

Chercheurs doctorants

Olivier Arsac, boursier INRIA
Patrick Capolsini, boursier INRIA jusqu'en mars
Jérôme De sousa, boursier DRET
Leonbattista Donati, boursier italien
Eric Hassold, allocataire MESR
Claude Huchet, allocataire MESR
Catherine Moulinet-Ossola, allocataire MESR
Alain Sausse, allocataire MESR
Noëlle Stolfi, allocataire MESR

Stagiaires

Frédérique Peyrot, DEA robotique et vision

2 Présentation du projet

La rédaction de ce rapport d'activité est une tâche particulièrement émouvante cette année. Jacques Morgenstern, notre chef de projet, nous a quitté à la fin du mois d'avril, à la suite d'une longue maladie. A l'origine, avec André Galligo, de la création de SAFIR, il avait su lui insuffler un esprit particulier que nous avons tous apprécié. Plus que le chef de projet, c'est l'homme qui nous manquera.

Dirigé par Jacques Morgenstern jusqu'à son décès, SAFIR est un projet commun à l'INRIA, à l'Université de Nice-Sophia Antipolis et au CNRS avec ses laboratoires associés de Mathématiques : J.A. Dieudonné, URA 168 (par l'intermédiaire de son équipe *Géométrie et Calculs*) d'une part, et d'Informatique, Signaux et Systèmes I3S, URA 1376, d'autre part. Une réunion du comité de concertation INRIA-UNSA-CNRS qui s'est tenue en juin a entériné cette collaboration suivant la nouvelle procédure de création de projets communs mise en place l'année dernière.

L'organisation du projet a été précisée pour l'année 94-95 : André Galligo assure la coordination scientifique, Loïc Pottier la responsabilité scientifique pour l'INRIA et les responsabilites administratives de chef de projet.

Jacques Morgenstern avait su donner au projet des directions de recherche dynamiques dans un domaine, le calcul formel, qui nécessite des compétences variées, dans des domaines comme les mathématiques, l'algorithmique, la logique, l'informatique théorique, où il était un spécialiste reconnu et écouté. Deux soucis constants qu'il a su faire partager aux membres de l'équipe sont ceux de l'effectivité et de l'applicabilité de nos recherches.

L'effectivité consiste à étudier ou mettre au point des algorithmes praticables et à décrire leur complexité (borne inférieure, par exemple, pour l'élimination algébrique). L'applicabilité des résultats consiste à identifier et abstraire, à partir d'exemples concrets (industriels ou académiques), des problèmes que les outils théoriques permettent de résoudre ou d'aborder. Avec comme conséquence prévisible des résultats théoriques, et leur application à d'autres problèmes pratiques plus ou moins voisins.

Le projet a ainsi puisé des exemples en mécanique des corps articulés, en robotique, en géométrie, en calcul numérique et en combinatoire, dont la résolution a conduit à des résultats en algorithmique, en complexité, et à des développements de logiciels originaux pour le calcul symbolique. Citons comme exemple la recherche de cette année en différentiation automatique, où les problèmes proviennent de contrats d'étude industriels, et ont conduit à un logiciel, *Odyssée*, appliqué avec succès à différents codes FORTRAN industriels, et à des résultats en complexité. On peut aussi citer notre contribution au projet européen PoSSo de résolution de systèmes polynomiaux, où nous avons mis au point un environnement logiciel (Central Control) permettant la coopération de systèmes de calcul formel et algébrique, tout en trouvant des résultats et algorithmes originaux (complexité de l'élimination algébrique, théorème des zéros effectif, calculs sur les invariants, bézoutiens, bases de Gröbner toriques).

L'activité du projet s'inscrit dans la communauté du calcul formel par sa participation au GDR Medicis, et au PRC Math-Info, au réseaux européens HCM SAC et SOL. Nous sommes également actifs dans différents DEA (de Mathématique et d'Informatique à Nice, ainsi qu'à Paris et à Marseille).

Avant de passer au détail de nos activités, nous reprendrons ce qu'écrivait J.Morgenstern l'année dernière :

Si nous n'avons pas l'ambition de proposer le langage parfait pour le Calcul Algébrique Formel ou des algorithmes "temps réels" pour la géométrie algébrique, nous désirons néanmoins nous appuyer sur une solide expérience pour proposer à la communauté du Calcul Scientifique des méthodes adéquates et des prototypes d'outils intégrés utilisables.

3 Actions de recherche

3.1 Aspects logiciels du calcul formel

3.1.1 Calcul formel distribué

Participants : Stéphane Dalmas, Marc Gaëtano, Alain Sausse

Notre but à moyen terme est de concevoir une architecture pour un environnement de calcul scientifique qui inclurait les outils de calculs dont l'ingénieur ou le scientifique a besoin : aussi bien des composants symboliques (un ou plusieurs systèmes de calcul formel, spécialisés ou généraux), que des composants numériques (génération de code, utilisation de bibliothèques ou de serveurs numériques, outils de transformation de programmes), une interface graphique, des outils de visualisation de données *etc*. Pour connecter ces systèmes, nous avons défini une architecture de communication organisée autour d'un composant appelé *Central Control* (CC dans la suite) et un protocole de communication, ASAP ([14]). Le rôle du CC est de gérer le lancement (éventuellement sur des machines distantes) et la terminaison des serveurs et l'échange des données et résultats en fournissant les possibilités de conversions nécessaires afin d'imposer le moins de contraintes possibles aux différents serveurs.

Le premier prototype de CC (écrit en Standard ML) a été terminé cette année par Alain Sausse. Ce prototype se présente comme un interprète pour un langage spécialisé, dans lequel il n'y pas de mécanisme de calcul propre (ceux-ci étant délégués aux différents serveurs), mais un mécanisme d'application de règles qui permet de spécifier les différentes transformations et conversions. Ce prototype a été connecté à *Maple*, *Macaulay*, GB et *Alpi* (ces trois derniers systèmes étant spécialisés dans les calculs de bases de Gröbner). Alain Sausse l'a utilisé pour ses expériences en décomposition primaire. Il a également réalisé une petite interface graphique simple à l'aide des outils TCL/TK. A l'issue de cette première expérience, nous avons décidé d'abandonner l'optique d'un langage trop spécialisé pour le pilotage du CC. Nous avons donc entrepris la réalisation d'une nouvelle version du CC qui se présente comme une extension d'un interprète *Scheme*. Les fonctionnalités définies pour le premier prototype ont été introduites dans un interprète *Scheme* écrit en C (il s'agit de SCM) comme de nouvelles primitives. A cette occasion, le protocole ASAP a été révisé et une nouvelle bibliothèque a été réalisée.

Ce travail entre dans le cadre du projet européen PoSSo (Polynomial Systems Solving) où le Central Control est utilisé pour faire coopérer les différents logiciels développés (une première version sera livrée en décembre). Dans ce cadre, nous avons reçu plusieurs visiteurs des équipes du projet PoSSo, venus interfacer leurs logiciels au CC: Hannes Schönmann de l'université de Kaiserlautern, Jean-Charles Faugère (CNRS et Institut Blaise Pascal) et Roberto Deias (université de Pise).

3.1.2 Expressions et formules

Participants : Christèle Faure, Claude Huchet, Stéphane Dalmas, Marc Gaëtano

Les systèmes de calcul formel, bien que dotés de fonctionnalités puissantes, effectuent essentiellement des calculs qu'on pourrait qualifier de génériques au sens où ils ne tiennent pas compte des domaines des fonctions ni des valeurs particulières que pourraient prendre les paramètres. Ainsi, certains résultats peuvent être faux pour certaines valeurs particulières des paramètres du problème. Les systèmes actuels laissent de telles vérifications à la charge totale de l'utilisateur. Une solution pour résoudre ces problèmes est de définir des expressions générales plus sophistiquées. De telles expressions, que nous appelons *expressions contraintes* (ou *expressions conditionnelles*) contiennent explicitement leurs domaines de validité et peuvent être définies par cas. On peut alors leur donner une sémantique plus proche des fonctions mathématiques. Le résultat d'un calcul réalisé avec ces expressions se présente donc en général comme une suite de résultats et de conditions de validité associées. C'est encore une expression contrainte.

Les problèmes qui se posent avec de telles expressions sont de deux ordres. D'abord des problèmes d'interface liés à la manipulation de ces résultats multiples et ensuite des problèmes algorithmiques de simplification des conditions. Deux implémentations d'expressions conditionnelles ont été réalisées. La première, dans *Maple*, a été réalisée par Claude Huchet. Elle permet d'avoir des conditions très générales et a été appliquée à l'écriture d'une fonction `solve` plus sûre. La deuxième a été définie par Christèle Faure par l'adjonction de nouveaux types dans le système *Axiom*. Ils ont été appliqués aux calculs d'intégrales (en collaboration avec J.H. Davenport de l'université de Bath) et aux calculs de limites (avec J.C. Williamson de l'ETH Zurich) ([15], [3]). Dans ce cadre, les formes des conditions sont plus restreintes mais les calculs réalisés pour leurs simplifications sont plus sophistiqués que dans l'implémentation dans *Maple*.

Ces expressions conditionnelles permettent d'exprimer toutes les relations mathématiques sous conditions, en particulier toutes celles que l'on trouve dans les tables de formules, d'intégrales ou de sommes. Ces connaissances sont en général stockées de manière algorithmique et dispersée à l'intérieur des systèmes (reconnaissance de formes particulières dans l'algorithme d'intégration par exemple). Nous étudions la réalisation d'une base de données de relations mathématiques qui regrouperait toute cette connaissance non algorithmique. Une telle base est destinée à fonctionner comme un outil indépendant de tout système, interrogeable par un langage de requête spécifique et disposant d'un petit moteur de déduction. Elle fournit des réponses qui sont conditionnelles, garantissant ainsi la justesse des résultats.

Un travail de réflexion et de spécification a été mené avec comme fil conducteur des exemples que les systèmes actuels ne peuvent traiter correctement. Ils ont permis de dégager les problèmes essentiels que sont la puissance du filtrage, l'expressivité du langage de requêtes, l'organisation des formules dans la base et le contrôle de la déduction. Une première implémentation a été effectuée en SML. Nous avons également défini un format externe pour la base de formules et réalisé son analyseur syntaxique.

3.1.3 Interfaces

Participants : Olivier Arsac, Stéphane Dalmas, Marc Gaëtano

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. Il y a donc une part importante d'expérimentation dans ce projet et l'interface en question devra être, au moins dans un premier temps, conçue pour pouvoir tester nos nombreuses idées. La première étape de ce travail est la réalisation d'un éditeur de formules mathématiques.

L'éditeur se présente sous la forme d'un processus serveur à qui une ou plusieurs applications délèguent la gestion d'une ou plusieurs fenêtres destinée à l'affichage et à l'édition de formules mathématiques. Pour faciliter le développement et une plus grande paramétrisation, l'éditeur contient un petit interprète Lisp dans lequel certaines de ses fonctionnalités sont décrites. Nous avons choisi d'utiliser *Klone*, un interprète embarquable développé par l'équipe KOALA.

Les parties de l'éditeur concernant la communication (y compris la délégation de fenêtres) et la contrôle ont été réalisées ainsi que l'inclusion de

Klone. L'éditeur est multi-contextes (utilisant une bibliothèque de processus légers ou *threads*) ce qui permet, par exemple, que l'édition dans une fenêtre ne bloque pas le réaffichage dans une autre. Des procédures de gestion graphique des caractères mathématiques de tailles variables (comme le signe de sommation ou l'intégrale) ont également été développées. Nous avons été en contact avec des membres du projet OPERA pour ces questions de typographie et d'édition de formules.

Un premier prototype capable d'afficher une formule devrait être prêt avant la fin de l'année. Une première interface complète devrait être produite l'année prochaine. Cette interface se connectera au Central Control et permettra donc d'accéder à plusieurs systèmes.

3.1.4 Reconnaissance de formules planes

Participants : Loïc Pottier

La recherche commencée l'année dernière sur la reconnaissance de formules mathématiques données par leur écriture dans le plan a été appliquée cette année dans un stage de Maîtrise Math-Info. Le cas des formules imprimées par Latex a été abordé, essentiellement par l'implémentation en C d'un algorithme rudimentaire de reconnaissance des caractères Latex, et sa connection à un module de reconnaissance de formules utilisant des grammaires contextuelles de graphes. Les résultats sont encourageants, tant sur la variété des formules reconnues, que sur le temps de calcul (polynomial en la taille de la formule, et négligeable en pratique).

Ce travail devrait se poursuivre au cours d'un stage de DEA, où l'accent sera mis sur la mise au point de la grammaire spécialisée pour Latex, ainsi que sur les primitives géométriques qu'elle utilise. Les applications en vue sont la numérisation de textes mathématiques, la saisie et la sélection à l'écran de formules imprimées par différents logiciels (*Maple*, *Mathematica* ou Latex), et la saisie par tablette graphique ou par scanner de formules manuscrites lorsque les logiciels de reconnaissance de l'écriture seront suffisamment satisfaisants.

3.1.5 Bibliothèque et constructeurs de structures mathématiques effectives

Participants : Bernard Mourrain, Loïc Pottier

Un rapide tour d'horizon des systèmes de calcul formel montre que l'essentiel de l'énergie utilisée dans leur développement concerne l'implémentation de nouveaux algorithmes mathématiques, dans des domaines de plus en plus variés et spécialisés. L'effort de structuration de ces algorithmes en bibliothèque est faible en comparaison. Seul *Scratchpad II*, devenu *Axiom*, permet la constrution de structures mathématiques (groupes, anneaux, corps, modules, espaces vectoriels, etc) effectives, avec la possiblité d'écrire des algorithmes polymorphes. Toutefois ces structures ne sont pas complètes, puisque l'axiomatique et le traitement logique des expressions en sont absents.

Nous avons mené deux expériences par le passé, l'une avec *geo* qui permet la construction d'algèbres variées adaptées au calcul sur les invariants, avec des algorithmes construits automatiquement, et *athéna* (repris en C dans *télémaque*), où quelques constructeurs catégoriques (produit cartésien, exponentielle, quotient) permettent la construction de nombreuses structures algébriques avec leur axiomatique, et des algorithmes construits automatiquement, qui s'avèrent équivalents en efficacité avec ceux écrits manuellement.

Nous avons repris récemment *télémaque*, et initié parallèlement une bibliothèque d'algorithmes génériques en C++ sur les polynômes, dans le but de développer un module de construction automatique de structures mathématiques effectives, qui pourrait être utilisables dans les systèmes de calcul formel actuels via le Central Control. L'algorithmique de ce travail fait l'objet d'un stage de DEA qui a commencé en novembre.

3.1.6 A#

Participants : Bernard Mourrain, Noëlle Stolfi, Christèle Faure

En tant que site bêta-test de A# (successeur du langage *Scratchpad*), nous avons eu la joie (et la peine) de tester le compilateur de ce nouveau langage. Nous avons par exemple implanté des constructeurs d'objets génériques comme les listes ordonnées, les monoïdes, les modules et les algèbres ainsi que des algèbres de déterminants. Les calculs dans l'algèbre de Cayley (algèbre extérieure avec deux opérateurs somme et intersection) ont également été abordés. La programmation dans ce langage, pour l'instant encore mal documenté, s'est révélée difficile et un peu déroutante. Cela a permis d'exhiber quelques problèmes, critiques pour une utilisation effective de cet outil dans un futur proche.

3.1.7 Visualisation de courbes et de surfaces, IZIC

Participants : Robert Fournier, Bernard Mourrain, Norbert Kajler, Frédérique Peyrot

IZIC est un logiciel de visualisation 3D basé sur la bibliothèque ZICLIB permettant des rendus de très bonne qualité et relié, à travers une interface, à un interprète TCL. Son originalité réside dans la possibilité de le commander à distance à partir d'un processus externe. Il peut donc être connecté facilement à un ou plusieurs systèmes de calcul formel, ceux-ci étant alors utilisés pour le calcul des points d'un maillage, et pouvant collaborer sur la même visualisation. Parmi les systèmes déjà connectés, il y a *Maple* (manipulations étendues des surfaces et courbes), *Reduce* (K. Gaskell et W. Neun, ZIB, Berlin, pour inclusion dans la nouvelle distribution de *Reduce*), *Macsyma* (O. Bachmann, RIACA, Amsterdam). Le code a été amélioré pour aboutir à une version stable disponible en ftp. Un article présentant l'approche que nous avons suivie a été accepté pour publication dans le *Journal of Symbolic Computation*.

Un stage de DEA sur le tracé implicite de courbes et surfaces algébriques a été effectué par F. Peyrot sous la direction de J.C. Yakoubsohn (Toulouse) et B. Mourrain. La méthode utilisée, dite d'exclusion, consiste à enlever des cubes de plus en plus petits ne contenant pas la surface ou la courbe puis à construire un maillage, visualisable par IZIC. Cette méthode a l'avantage de décrire de manière précise et sûre la surface implicite contrairement aux méthodes utilisées actuellement dans les systèmes de calcul formel. Une implantation en *Maple* et C a été réalisée.

3.1.8 Hyperion

Participants : José Grimm

Le logiciel d'approximation rationnelle HYPERION, dont l'écriture a commencée l'année dernière, a été déposé à l'APP (Agence pour la protection des programmes). Ce logiciel est destiné à remplacer à terme le logiciel SISYPHE dont le développement au dessus de LELISP a été abandonné.

La première étape a été d'écrire un interprète Lisp en C avec des structures de données de base adaptées à nos objectifs (entiers courts 32 bits, entiers longs de taille arbitraire, flottants double et quadruple précision, polynômes et matrices de polynômes à coefficients réels ou complexes en double ou quadruple précision). Nous avons implémenté des fonctions de conversion entre tous ces objets, et une arithmétique générique au niveau Lisp. L'arithmétique sur les entiers longs et les nombres rationnels est l'ancienne arithmétique de SISYPHE, traduite dans le langage C ([16]). Nous avons également inclus l'algorithme de factorisation des entiers. Certains algorithmes ont été améliorés, c'est ainsi que le calcul des pgcd est trois fois plus rapide dans certains cas que l'ancienne version en Lisp. Ce logiciel a été porté sur station DEC Alpha. Nous n'avons pas encore utilisé les possibilités de l'architecture 64 bits, et nous espérons rendre le programme encore plus efficace.

La partie polynômes réels et complexes a été laissée inchangée par rapport à la version précédente de HYPERION. Le portage sur DEC Alpha nous a réservé quelques surprises, dans la mesure où la bibliothèque C fournie par SUN n'est pas strictement conforme à la norme ANSI. Du fait qu'il n'y a pas de quadruple précision sur cette machine, une grosse partie du logiciel est actuellement inopérante.

3.2 Algèbre, géométrie et complexité

3.2.1 Géométrie Algébrique, Robotique et Vision

Participants : Bernard Mourrain, Noëlle Stolfi

Nous avons poursuivi l'étude de problèmes issus de la robotique et de la vision par des méthodes d'algèbre effective. Le premier pas consiste à obtenir une énumération correcte des solutions des systèmes algébriques rencontrés. Ceci est souvent lié à une bonne analyse de la géométrie sous-jacente. Concernant les problèmes algébriques où interviennent des distances entre points, nous avons décrit l'anneau de Chow correspondant, et avons pu ainsi obtenir des formules d'énumérations pour une grande classe de problèmes rencontrés en robotique (visée de trois points, problèmes des quatre barres, triangle s'appuyant sur trois surfaces). Un article présenté à MEGA 94 ([11]) regroupe ces différents résultats. Ce travail devrait se poursuivre par l'obtention de formules du même type pour des problèmes de contraintes algébriques sur les déplacements.

3.2.2 Manipulation symbolique d'objets géométriques.

Participants : Bernard Mourrain, Noëlle Stolfi

Nous désirons manipuler directement des objets géométriques tels que des points, des droites, des cercles, des sphères ou des déplacements sans utiliser de système de coordonnées, mais en se basant sur les propriétés intrinsèques de ces objets. Pour cela, nous utilisons des outils comme les calculs d'intersection dans l'algèbre de Cayley, des lois de redressement dans des algèbres de déterminants, la structure d'algèbre de Clifford (qui permet de retrouver des calculs de quaternions) ainsi que l'espace des sphères. Ces techniques faisant appel à la théorie des invariants, permettent une représentation plus synthétique des calculs. Nous comptons appliquer ce traitement formel des objets géométriques dans des domaines comme la robotique et la vision. Un chapitre de l'ouvrage *Invariants Methods in Discrete and Computational Geometry* synthétise ces méthodes ([5]).

Un autre domaine où la théorie des invariants apparaît de manière surprenante est le traitement du signal. Ainsi, on peut s'intéresser à la recherche du nombre de sources se superposant dans un signal en utilisant les statistiques de ce signal. D'un point de vue algébrique, ce problème consiste à savoir sous quelles conditions un polynôme homogène de degré d en n variables est une somme de p puissances de formes linéaires. Un travail en collaboration avec P. Comon (Thomson et CNRS) a permis de tester quelques méthodes fournissant une telle décomposition et a conduit à un article résumant les différents cas que l'on connaissait déjà et présentant certains algorithmes de décomposition ([6]).

3.2.3 Résolution de systèmes polynômiaux

Participants : Bernard Mourrain, Noëlle Stolfi

Nous nous intéressons ici à la résolution numérique de systèmes de n polynômes $p_1, \ldots, p_n$ en n variables $x_1, \ldots, x_n$ ayant un nombre fini de zéros (intersection complète). Nous reprenons une méthode introduite par Bézout et étendue par Dixon (1908) pour calculer des résultants mais en l'abordant sous un angle nouveau. Les outils utilisés (les Bézoutiens) permettent en effet de construire les matrices de multiplication par x_l dans $B = \mathbf{C}[x_1, \ldots, x_n]/(p_1, \ldots, p_n)$ ou directement, par des calculs matriciels simples, un multiple de la forme de Chow associée à B. Cette approche, offrant une alternative aux incontournables (jusqu'à présent) calculs de base de Gröbner, semble très prometteuse pour la résolution efficace de systèmes dans le cas d'une intersection complète.

3.2.4 Bases de Gröbner toriques

Participants : Loïc Pottier, Catherine Moulinet-Ossola

L'étude des bases de Gröbner d'idéaux toriques s'est poursuivie dans deux directions : l'amélioration de l'algorithme de calcul et la recherche d'applications. L'algorithme, du type de celui de Buchberger, opère maintenant non plus sur des différences de monômes, mais sur des vecteurs entiers. Diverses améliorations ont été apportées. D'abord sur le calcul des paires critiques utiles, où l'utilisation d'un algorithme de géométrie algorithmique (sur une idée d'A. Galligo), s'est avérée très efficace (voir 3.2.5). Ensuite sur la méthode de calcul des formes normales, où diverses propriétés des enchaînements de réductions évitent de nombreux tests. Enfin, il semble que l'on peut éviter l'introduction d'une variable supplémentaire dans les générateurs de l'idéal de départ, et réduire ainsi le calcul des paires critiques à celui du calcul de certaines différences et sommes des vecteurs associés. Expérimentalement, cette conjecture est vérifiée, et on peut gagner jusqu'à un facteur 100 en temps de calcul. Il reste néanmoins à la démontrer. L'algorithme est implémenté dans le logiciel *bastat*, en C, connecté au Central Control, et disponible par FTP.

Les applications des bases de Gröbner toriques que nous avons envisagées sont de trois types. Le premier concerne la résolution de systèmes polynomiaux creux. Le théorème de Bernstein et Kouchnirenko, ainsi que les résultats de Sturmfels, Canny et Emiris, amènent à étudier une forme de Chow d'une variété torique, et à rechercher les points entiers d'un polytope. Calculer directement la base torique de l'idéal associé à la variété semble hors de portée (en temps), par contre la recherche des points entiers est plus accessible. Ce point semble plus prometteur, et regroupe le second type d'applications. La recherche des points entiers dans un polytope (qui est un sujet actif) est directement liée à la programmation linéaire entière, où les algorithmes existants ne sont satisfaisants que dans les situations génériques, car ils sont des adaptations de l'algorithme du simplexe, qui travaille sur les réels ou les rationnels. Les bases de Gröbner toriques permettent de résoudre élégamment ce problème en donnant une description synthétique des points entiers, par règles de réécriture, qui est directement utilisable. En pratique, des problèmes en dimension 25 peuvent être résolus de cette manière. Deux applications concrètes de la recherche de points entiers d'un polytope sont la parallélisation de programmes et le calcul des résultants creux, où nos techniques restent à tester. Le dernier type d'applications des bases de Gröbner toriques concerne les réseaux entiers, où elles permettent de calculer les plus petits vecteurs et les minimaux successifs (*cf* la section 3.2.5). Des applications sont possibles en théorie des codes. Citons enfin le fait que les bases de Gröbner toriques donnent un algorithme simple de calcul d'une triangulation de Delaunay de points entiers (cependant loin de concurrencer en pratique les algorithmes existants, bien que d'une complexité théorique équivalente).

3.2.5 Plus petits vecteurs d'un réseau, cas des réseaux de Craig

Participants : Catherine Moulinet-Ossola

Nous avons amélioré nos programmes de calculs de bases de Gröbner sur les binômes par l'adaptation de deux algorithmes de recherche de points minimaux pour sélectionner les paires critiques utiles. Un des critères permettant de juger de l'utilité ou non d'une paire critique est de ne conserver que celles associées à deux binômes dont le ppcm des termes de tête respectifs est minimal par rapport à tous les autres ppcm (minimal par rapport à l'ordre induit par la division euclidienne). Le premier algorithme utilisé est basé sur une technique de balayage des composantes, et le second sur la technique "diviser pour régner". Le premier améliore déjà nos temps de calculs, le second se montre encore plus performant.

Concernant le problème de recherche de vecteurs de norme minimale dans les réseaux arithmétiques, nous avions effectué l'année dernière de nombreux tests sur des réseaux générés aléatoirement ainsi que sur une famille particulière, définie par deux paramètres entiers n et m, les réseaux de Craig. Ces résultats expérimentaux présentaient des particularités intéressantes: certaines lois semblaient vérifiées par les plus petites normes euclidiennes et normes 1 en fonction du paramètre m pour un n fixé. Dans le cas des normes euclidiennes, nous avons pu prouver certaines de ces relations. Notre approche a consisté à chercher dans un premier temps les combinaisons linéaires qui engendraient un vecteur de norme minimale et à d'exhiber certaines récurrences entre deux combinaisons successives (passage du cas m au cas $m + 1$). Certaines autres récurrences ont été conjecturées. De ces résultats découlent les lois donnant les normes minimales en fonction des deux paramètres d'entrée.

Ce travail est décrit dans le mémoire de thèse de Catherine Moulinet-Ossola qui devrait être soutenu au début de l'année prochaine.

3.2.6 Borne inférieure temps-espace

Participants : José Grimm, Loïc Pottier, Nicole Rostaing-Schmidt

A partir d'un résultat algorithmique de A. Galligo et J. Abbott, nous avons obtenu une borne inférieure pour le produit Temps×Espace nécessaire à l'inversion d'une suite récurrente $x_{n+1} = f(x_n)$. En simplifiant, le résultat dit que le produit Temps×Espace pour inverser une suite de longueur p est minimum pour un espace égal à $\log p$ et est alors de l'ordre de $p \log(p)^2$. Ce résultat trouve une application dans la méthode inverse en différentiation automatique appliquée à une boucle. Plus généralement, nous comptons l'étendre au cas des arbres, avec en vue la recherche de compromis temps-espace en différentiation automatique.

3.2.7 Etude de la complexité et transformation symbolique de programmes de calcul de formes linéaires

Participants : Nathalie Eyssette

Nous étudions la complexité additive de systèmes de formes linéaires grâce à différents outils : le logiciel de manipulation de graphes que nous avons construit pour outil pratique, la géométrie projective, l'algèbre linéaire, une certaine famille de graphes de calcul et la complexité structurelle pour outils théoriques.

Chaque système de formes linéaires peut être représenté par non pas un, mais une famille de graphes de calcul linéaire. L'un de nos buts est de chercher un meilleur représentant de cette famille : optimal (en nombre d'additions) et de structure la plus simple possible. Il s'agit évidemment d'un problème difficile. En le restreignant au calcul de deux formes linéaires f et g à coefficients dans $Z/2Z$ à variables disjointes (f et g n'ont aucune variable commune), nous avons montré que tout algorithme de calcul optimal du système $\{f, g\}$ est la réunion de deux algorithmes optimaux disjoints, qui calculent respectivement f et g (équivalent linéaire de la conjecture forte de Strassen). Nous avons également généralisé le résultat aux systèmes de la forme $\{f_1, \ldots, f_n, g\}$, où les ensembles de variables de $\{f_1, \ldots, f_n\}$ et de g sont disjoints, aux systèmes de n formes linéaires à variables disjointes, et à tous les anneaux de coefficients. Ce travail a été effectué en collaboration avec John M. Robson, Professeur à l'Université de Canberra (Australie).

3.2.8 Algorithmes de calcul des réductions de Hermite et de Smith

Participants : Roger Marlin

Le but de ce travail est de donner, dans un anneau principal discret à division explicite, des algorithmes de calcul des réductions de Hermite et de Smith. Une revue des méthodes classiques a été faite. Certains de ces algorithmes ont été éclaircis, améliorés et généralisés. Des implémentations en divers langages, comme Lisp, C et Ada ont été réalisées. Ce travail s'effectue en collaboration avec H. Lombardi (de l'université de Besançon) et S. Labhalla (université de Marrakech) et fait l'objet d'un article soumis à *Journal of Symbolic Computation* (*Hermite Reduction Algorithms for Matrices with Polynomial Entries*).

3.2.9 Décomposition primaire d'idéaux

Participants : André Galligo, Alain Sausse

La décomposition d'un idéal en intersection d'idéaux primaires (qui généralisent la puissance d'un idéal premier) est un problème difficile algorithmiquement pour lequel on n'a pas encore de solution satisfaisante. Les algorithmes implémentés dans les systèmes de calcul formel, comme *Axiom* ou *Reduce*, ne permettent pas de travailler dans des anneaux de plus de trois ou quatre variables, pour des raisons de temps de calcul ou de place mémoire.

Nous avons repris un travail de Mike Stillman, David Bayer et André Galligo qui permet de définir une nouvelle famille d'algorithmes de décomposition primaire qui s'appuient sur des propriétés de platitude. Soit $I \subset k[x_1, ..., x_n] = A$ un idéal et V la variété algébrique associée. Pour calculer une décomposition primaire de I, nous commençons par projeter V sur K^{m+1} où $m = \dim A/I$ et K est la clôture algébrique de k (étape d'élimination). La factorisation du résultat (une hypersurface) en $P = \prod P_i$ nous donne les composantes maximales (pour la dimension).

Si la direction de projection est bien générique, en calculant les quotients itérés $(I : P_i)^*$ on efface ces composantes de la variété. Il ne reste plus qu'à recommencer le processus avec ces nouveaux idéaux. Malheureusement, mettre un idéal en position générique par une changement de coordonnées réclame en général des calculs trop coûteux. Notre approche est donc d'utiliser des projections bien choisies et des détecteurs de platitude pour accélérer le processus.

Pour l'implémentation de cet algorithme, nous utiliserons les possibilités du Central Control pour faire collaborer des systèmes efficaces, spécialisés en géométrie algébrique (*Macaulay*, GB, *Alpi*) et un système général, *Maple* en l'occurrence, qui fournit les opérations qui ne sont pas implémentées dans les systèmes précédents (comme la factorisation des polynômes). Le Central Control permettra aussi de distribuer ces calculs coûteux sur plusieurs machines afin d'exploiter une certaine forme de parallélisme présente dans nos algorithmes.

Cette année nous avons surtout expérimenté nos idées sur plusieurs exemples significatifs. Nous avons notamment travaillé sur un exemple difficile, provenant d'un problème de mécanique céleste.

3.2.10 Etude de la topologie des courbes planes

Participants : Jérôme De sousa, Frédéric Eyssette, Michel Merle

Dans son travail de thèse, en fin de rédaction, Jérôme De sousa étudie la topologie des courbes algébriques planes complexes. L'originalité de cette étude est l'utilisation complémentaire de techniques formelles et numériques. Une première étape est l'étude locale des singularités de la courbe. Au voisinage de chaque point critique nous calculons l'intersection de la courbe avec un polydisque centré en la singularité. L'étude de cette intersection, qui forme un entrelacs, se fait par l'intégration du système différentiel réel vérifié par l'intersection. Le résultat obtenu est une discrétisation de l'entrelacs. A partir de là, nous pouvons déterminer le nombre de composantes connexes en suivant un point le long du contour jusqu'à revenir en position initiale et en visitant tous les points. Les exposants ou paires de Puiseux sont déterminés en comptant le nombre de tours effectués par certains groupes de racines autour de leur barycentre. Les multiplicités d'intersections entre composantes sont déterminées par l'intersection d'une composante et d'une surface ayant pour bord l'autre composante. Ces opérations sont totalement topologiques contrairement aux méthodes habituelles qui, comme la désingularisation, sont algébriques. Nos algorithmes ont été implémentés en *Maple* avec génération de code C pour résoudre le système différentiel (en utilisant le programme macroC de P. Capolsini). Les algorithmes nécessaires à la détermination de la topologie sont écrits en C et utilisent des techniques d'infographie. Les entrelacs sont visualisés grâce au logiciel interactif *agat* de O. Arsac.

3.2.11 Singularités des vues de surfaces éclairées

Participants : Leonbattista Donati, Michel Merle

On cherche à classifier les singularités des vues d'une surface générique illuminée. Les jeux d'ombre et de lumière sur une surface fournissent des informations sur la *forme* de l'objet qui s'ajoutent à celles données par son contour apparent. Il est donc intéressant du point de vue de la reconnaissance de forme en vision automatique de tenir compte de deux nouvelles entités *morphogénétiques* qui sont la courbe d'ombre et la courbe d'ombre portée.

En continuant l'étude amorcée par J.P. Henry et M. Merle dans *Shade, Shadow and Shape*, nous appelons configurations *sol–y–sombra* les classes d'équivalence des ensembles de ces trois courbes selon l'équivalence analytique dans le plan visuel. Notre but est alors de fournir une classification des configurations sol–y–sombra *stables* (qui ne changent pas pour de petits mouvements de l'observateur ou de la source lumineuse) et de codimension un (qui correspondent aux événements de transition entre deux vues stables). Pour ce faire on introduit un concept d'équivalence infinitésimale qui donne des conditions nécessaires pour qu'une configuration sol–y–sombra soit stable: le module des déformations infinitésimalement triviales associé à f est un objet calculable qui contient les déformations h telles que $f+\varepsilon h$ est équivalente à f modulo ε^2. Nous dirons que f est infinitésimalement stable quand toute déformation infinitésimale est triviale. Il est alors possible de donner la liste complète des singularités infinitésimalement stables (il y en a six) et de codimension 1 (neuf singularités). Nous avons enfin montré que toute singularité sol–y–sombra infinitésimalement stable est effectivement stable.

Ce travail fait l'objet de la thèse de Leonbattista Donati qui sera soutenue à la fin de cette année ou au tout début de l'année prochaine.

3.3 Applications du calcul formel, liens formel-numérique

3.3.1 Utilisation du calcul formel pour la modélisation et la simulation de phénomènes physiques

Participants : Patrick Capolsini, Yves Papegay

L'utilisation du calcul formel dans le domaine général de la simulation apparaît comme un prolongement naturel des travaux sur l'utilisation du calcul formel en mécanique développés les années précédentes. Modéliser consiste à obtenir, à partir des caractéristiques d'un phénomène physique, et dans un certain cadre théorique, un système d'équations gouvernant son évolution. La simulation s'appuie sur l'analyse et la résolution numérique de ce système pour décrire quantitativement cette évolution. Dans ce cadre, l'utilisation du calcul formel doit permettre de générer automatiquement un modèle sous forme symbolique, de l'analyser, puis de produire automatiquement un code de résolution numérique spécialisé. Nous avons identifié, à partir de l'étude de problèmes industriels concrets dans trois domaines d'applications – dynamique des systèmes polyarticulés, planification de tâches d'assemblage en robotique, et calibration d'instruments optiques – des fonctionnalités générales nécessaires à une telle utilisation. Parallèlement, nous avons mené une réflexion sur la méthodologie de la modélisation dans le but d'intégrer ces outils dans un environnement général de calcul scientifique pour la modélisation et la simulation. Nous avons commencé le développement des outils logiciels correspondant.

Dans la thèse qu'il a soutenu en début d'année ([1]), Patrick Capolsini présente le logiciel SYGMMAE, développé au dessus du système de calcul formel *Maple*. A partir de la description d'un mécanisme arborescent, SYGMMAE en calcule symboliquement les équations de la dynamique (en utilisant le formalisme de Kane). Il permet leur manipulation et produit automatiquement un code efficace de simulation numérique (en C ou FORTRAN). Ce travail a fait l'objet d'un contrat avec le CNES.

La spécificité de la réalisation de tâches d'assemblage en robotique est de nécessiter un mouvement de l'effecteur qui conserve le contact entre l'objet manipulé et l'obstacle. La planification de ce mouvement requiert le calcul exact de l'image de l'obstacle dans l'espace de configuration de l'objet. Un calcul complet est impraticable dans le cas tridimensionnel d'un environnement polyhédral non convexe. Nous avons développé une méthode paresseuse de calcul de cette image, prenant en compte progressivement la non-convexité de l'obstacle ([9], [10]). Cette méthode et les outils nécessaires ont été implémentés dans le système de calcul formel *Mathematica*. Ce travail a fait l'objet d'une collaboration avec des chercheurs de ETL (Electro Technical Laboratory, à Tsukuba, Japon) et H. Hirukawa à l'université de Stanford.

La calibration d'instruments optiques embarqués sur satellites pour satisfaire une mission d'observation s'appuie sur un modèle complexe, faisant intervenir un grand nombre de paramètres. Nous nous sommes intéressés à la production d'outils symboliques permettant de vérifier la cohérence du modèle en tenant compte des contraintes de dimensionnement, d'analyser la sensibilité des fonctions de performances à la variation de divers paramètres et de simuler le comportement de l'instrument. Ce travail fait l'objet d'un contrat avec l'Aérospatiale (établissement de Cannes) qui devrait se prolonger l'année prochaine.

3.3.2 Dérivation automatique

Participants : Frédéric Eyssette, Christèle Faure, André Galligo, Eric Hassold, Nicole Rostaing-Schmidt

Nous avons poursuivi l'étude de la différentiation automatique et le développement du logiciel *Odyssée* de différentiation de programmes FORTRAN.

Afin d'améliorer les performances des programmes engendrés par *Odyssée*, C. Faure a enrichi ses fonctionnalités de manipulations symboliques d'expressions. Une phase de prétraitement permet d'isoler les sous-expressions constantes (au sens de la dérivation) en introduisant de nouvelles affectations. De même, les arguments des appels de fonction et de procédure sont également affectés à des variables intermédiaires pour réduire le nombre d'opérations effectuées dans l'application de la formule de dérivation des fonctions composées. Ce travail a eu lieu en collaboration avec le LEAP, (ENSIGC, Toulouse) dans le cadre d'un contrat avec ELF. Outre ce prétraitement minimal, il est maintenant aussi possible de découper toutes les expressions algébriques en opérations élémentaires. Ceci permet de se rapprocher de la borne de complexité du mode inverse (résultat théorique de Baur-Strassen et Morgenstern).

N. Rostaing-Schmidt a développé un mode inverse qui ne requiert pas d'hypothèses sur la structure du code. L'algorithme inclut des instructions de sauvegarde dans la phase de calcul de la fonction initiale et restitue les valeurs nécessaires lors du calcul des formes linéaires transposées. Nous avons ainsi étendu la classe de programmes acceptée en mode inverse aux boucles qui terminent avec un test d'arrêt dépendant de l'exécution (boucles WHILE). Ce nouveau module est en cours de validation sur des programmes développés dans le projet SINUS.

Dans le cadre de la collaboration avec le LEAP, nous avons étendu les fonctionnalités de la différentiation en mode direct. E. Hassold a développé le calcul de la matrice jacobienne complète. Cela nous a conduit à mettre en œuvre un calcul de graphe de dépendances des variables (au sens de la dérivation).

Enfin, nous avons mis au point une première version d'*Odyssée* avec un langage de commande et une interface graphique (stage de DEA de V. Vyskocil) permettant d'utiliser le système indépendamment de l'interprète du langage CAML-LIGHT (dans lequel *Odyssée* est écrit).

Aussi bien pour réduire le nombre de dérivations inutiles dans le calcul de la matrice jacobienne que pour diminuer la place mémoire requise dans le mode inverse (par une délimitation plus précise des parties de tableaux modifiées par une instruction), il est apparu qu'une analyse fine des dépendances entre les variables non scalaires était indispensable. Cependant cette analyse est complexe car elle doit tenir compte du sens de parcours des tableaux dans les différentes instructions. En collaboration avec L. Hascoët (projet SINUS), nous étudions un algorithme et des structures de données permettant un calcul de dépendances fines entre les variables. La mise en œuvre de l'algorithme obtenu utilisera à la fois des outils d'*Odyssée* et ceux du logiciel FORESYS (SIMULOG).

Un travail est aussi en cours pour parvenir au traitement global d'un graphe d'appels. Dans la version actuelle du système, les procédures appelées par une procédure à dériver ne sont pas dérivées automatiquement. L'automatisation complète du processus de dérivation nous a donc amené à spécifier les informations nécessaires à un tel traitement global et à étendre la base d'informations d'*Odyssée* (qui les stocke). Un contrôle de cohérence entre les procédures engendrées et leurs appels pourra ainsi être effectué. La base d'information doit aussi permettre de traiter les appels de procédures dont le code source n'est pas disponible et d'appliquer à une suite d'instructions possédant une sémantique mathématique connue, une méthode de dérivation adaptée.

3.3.3 Approximation de fonctions de transfert

Participants : José Grimm, Juliette Leblond, Pascale Fulcheri

L'objectif de nos travaux est d'écrire un algorithme qui trouve les paramètres d'une fonction de transfert rationnelle f_0 à partir de mesures fréquentielles $f_0(i\omega)$.

Une première transformation consiste à écrire $f_0(i\omega) = f_1(z)$, où z est dans le cercle unité, et f_1 est une fonction de transfert discrète, une fraction rationnelle de même degré que f_0. L'algorithme utilisé procède en deux étapes : nous cherchons d'abord le meilleur approximant de f_1 dans l'espace de Hardy du disque, puis le meilleur approximant rationnel de cette fonction. Notant g_1 ce résultat, il est facile d'en déduire le résultat final g_0. Nous avons rajouté à HYPERION le code qui permet d'obtenir les coefficients, pôles et zéros de g_0 à partir de g_1.

La théorie implémentée l'année dernière supposait f_0 scalaire et strictement propre. Nous avons tenté de lever ces deux restrictions. Le cas non scalaire a été implémenté dans Basile par P. Fulcheri dans le cas réel, en utilisant le paramétrisation de Schur et un Hessien estimé. Les résultats sont encourageants, mais le programme est très lent. Nous avons écrit dans HYPERION toutes les primitives nécessaires sur les matrices de polynômes pour traiter le cas complexe avec Hessien calculé explicitement ; l'implémentation de l'algorithme lui-même est en cours.

Dans le cas où la fonction f_0 n'est pas strictement propre, elle est somme d'une partie strictement propre et d'une constante c qu'il faut estimer ou calculer. Dans le premier jeu de données fournies par le CNES, l'estimation est facile, dans le second, on a été obligé de la calculer par programme. On est donc amené à un problème du type : minimiser la norme de $f_1 - g_1 - c\xi$ sur I, en bornant la norme de g_1 sur J, où g_1 est dans H_2. Il s'agit d'un cas particulier d'optimisation dans un espace de Hilbert, où I et J sont deux sous-espaces orthogonaux, et ξ est fixé a priori. Ce problème admet une solution unique sous certaines conditions, et la solution est donnée par $g_1 = T(f_1 - c\xi)$, pour un certain opérateur T. La quantité c s'exprime simplement en fonction de Tf_1 et $T\xi$. Comme l'espace que l'on considère est de dimension infinie, ces formules théoriques sont entachées d'erreur ; en particulier, dans la base choisie, la base des coefficients de Fourier, la fonction ξ se laisse mal approcher. Nous étudions diverses possibilités pour minimiser cette erreur, par exemple en cherchant des formules équivalentes pour T, ou en prenant des bases plus adaptées. Nous essayons aussi d'étudier l'erreur globale en fonction des paramètre suivants : le bruit de mesure sur f_1, l'erreur de troncature due à la dimension finie, et l'erreur numérique intervenant lors de l'inversion de la matrice qui apparaît dans l'opérateur T. Plusieurs implémentations de cet algorithme ont été écrites, mais, pour l'instant, aucune ne possède la stabilité numérique suffisante pour traiter des données réelles (i.e. bruitées).

4 Actions industrielles

4.1 Hyperfréquences

Participants : José Grimm

La collaboration avec le CNES, qui a commencé il y a quelques années s'est poursuivie, avec la signature d'un nouveau contrat, en collaboration avec le projet MIAOU. Il s'agit essentiellement d'étudier le problème d'approximation décrit dans la section 3.3.3, dans le cas de deux entrées, deux sorties, en utilisant soit la paramétrisation de Schur, soit une paramétrisation spécifique au cas 2-2, qui utilise le théorème de factorisation spectrale.

4.2 GENIE

Participants : Christèle Faure, Nicole Rostaing-Schmidt

Nous avons commencé à travailler sur le contrat GENIE (Dassault-INRIA) en collaboration avec le projet SINUS. Nous devons y appliquer nos outils de différentiation automatique (*Odyssée*) à des problèmes d'optimisation de formes en aérodynamique. Nicole Rostaing-Schmidt sera ingénieur-expert dans le cadre de ce contrat.

4.3 Chimie

Participants : Christèle Faure, Eric Hassold, Nicole Rostaing-Schmidt, André Galligo, Frédéric Eyssette

Nous avons conclu un contrat de recherche d'un an avec ELF, en collaboration avec le laboratoire d'analyse fonctionnelle des procédés (LEAP) de l'ENSIGC de Toulouse. Le sujet en était l'utilisation de la différentiation automatique pour la modélisation, la simulation et l'optimisation des procédés. Christèle Faure est ingénieur-expert dans le cadre de ce contrat. La société ELF s'étant déclarée safisfaite des résultats obtenus, ce contrat devrait être renouvelé l'année prochaine.

4.4 CNES

Participants : Frédéric Eyssette

Nous avons conclu un contrat de neuf mois avec le CNES sur l'étude de la sensibilité aux erreurs numériques de programmes FORTRAN. Nous devons étendre les fonctionnalités d'*Odyssée* afin de pouvoir appliquer divers critères (Miller, Larson-Sameh) à des zones de code (régions à entrée et sortie unique) choisies par l'utilisateur.

4.5 EDF

Participants : Frédéric Eyssette, Nicole Rostaing-Schmidt, Christèle Faure,

Un contrat de dix mois a été conclu avec EDF, en collaboration avec le projet PROMATH (Jean-Charles Gilbert). Le but est d'appliquer nos techniques de différentiation automatique à une maquette de logiciel de thermodynamique diphasique. Après une première étape de vérification de l'applicabilité d'*Odyssée*, nous devons comparer les résultats d'une étude de sensibilité réalisée à l'aide des dérivées évaluées en modes direct et inverse avec une approche par différences finies.

4.6 Optique

Participants : Yves Papegay

Nous avons conclu un premier contrat d'étude à la fin de l'année avec l'Aérospatiale (établissement de Cannes) pour étudier l'apport du calcul formel dans la conception de systèmes optiques embarqués sur satellites. Ce contrat devrait être renouvelé l'année prochaine.

5 Actions nationales et internationales

5.1 Actions internationales

5.1.1 Invitations

Nous avons reçu Allan Borodin (professeur à l'université de Toronto) et Joos Heinz (professeur à l'université de Santander) pendant trois semaines en avril, Ashutosh Rege, doctorant à l'université de Californie à Berkeley pendant trois jours en août, Andreas Strotmann, un doctorant de l'université de Cologne pendant deux mois, Hannes Schönmann de l'université de Kaiserlautern pendant une semaine, Roberto Deias de l'université de Pise pour trois jours.

5.1.2 PoSSo

Notre équipe participe au projet européen ESPRIT/BRA PoSSo (Polynomial Systems Solving, projet numéro 6846), qui a commencé en octobre 92. PoSSo rassemble une dizaine d'équipes universitaires. Nos contributions concernent l'architecture (le Central Control), les interfaces, l'étude des systèmes creux et des variétés toriques et de la décomposition primaire. Le projet PoSSo devrait se terminer à l'automme 1995.

5.1.3 Réseau HCM SAC

L'équipe SAFIR est le nœud français du réseau *Capital Humain et Mobilité* SAC consacré au calcul formel et qui a vu le jour officiellement cette année. Ce réseau regroupe huit nœuds (CAN/RIACA en Hollande qui est le nœud principal, Bath en Grande-Bretagne, Linz en Autriche, Genova en Italie, Paderborn en Allemagne, l'université de Cantabria en Espagne et l'ETH à Zurich).

5.1.4 Réseau HCM SOL

De par notre activité sur les bases de Gröbner toriques, nous participons au réseau *Capital Humain et Mobilité* SOL sur la résolution des équations diophantiennes linéaires. Nous sommes rattachés au nœud français du réseau qui est à l'INRIA Nancy. Catherine Moulinet-Ossola a participé au deuxième workshop SOL à Orsay en novembre.

5.1.5 RIACA

Norbert Kajler collabore activement avec RIACA (Research Institute for Applications of Computer Algebra, à Amsterdam) sur le thème des interfaces homme-machine pour le calcul formel. Il participe à la préparation du colloque HISC 96, la rédaction d'un livre collectif sur ce sujet et co-éditera un numéro spécial du *Journal of Symbolic Computation*.

5.1.6 Océanographie

Dans le cadre de notre activité en différentiation automatique, un travail commun a été réalisé avec la société CETIS et le Max Planck Institut für Meteorologie (Hambourg) sur la génération automatique d'un code adjoint. Ce travail a été exposé au *second workshop on Adjoint Applications in Dynamic Meteorology* (*Writing the code of a primitive equation GCM with an automatic adjoint generator*). Cette collaboration devrait se poursuivre l'année prochaine.

5.1.7 OpenMath

Le projet SAFIR participe à l'effort international *OpenMath*. Le but en est la définition d'un protocole de communication d'expressions mathématiques entre applications. Ces applications pouvant être des systèmes de calcul formel, des interfaces graphiques, des bases de données ou des systèmes de traitement de textes. Nous avons accueilli Andreas Strotmann qui travaille sur ce sujet pendant deux mois dans notre projet. Stéphane Dalmas, Marc Gaëtano et Yves Papegay ont assisté au second atelier OpenMath qui s'est tenu au collège Sainte-Catherine à Oxford en juillet. Marc Gaëtano y a présenté un exposé sur ASAP et le Central Control.

5.2 Actions nationales

5.2.1 Invitations

Nous avons reçu Jean-Charles Faugère, chargé de recherche au CNRS (institut Blaise Pascal) pendant trois jours et Emmanuel Saint-James, maître de conférences à Paris 6 pour la même durée (au mois d'octobre).

5.2.2 PRC et GDR

Le projet SAFIR participe aux PRC et GDR qui couvrent le domaine du calcul formel, le GDR Medicis et le PRC Math-Info.

Par son activité sur la différentiation automatique, le projet participe à deux GDR: le GDR *Méthodes variationnelles en météorologie et océanographie* et le GDR *Optimisation de formes.*

5.2.3 Organisations de séminaires

Le projet organise le séminaire *Mathématiques et informatique* qui a accueilli cette année une douzaine de conférenciers.

Loïc Pottier a co-organisé avec Bruno Martin la journé I3S/INRIA sur la complexité structurelle (quatre exposés, le 14 avril).

Bernard Mourrain organise le groupe de travail *Formes et Formules* qui se réunit régulièrement au département de mathématiques de l'université de Nice (une douzaine de séances cette année).

6 Diffusion des résultats

6.1 Diffusion de produits

Une première version de notre système d'animation d'algorithmes *agat* est distribuée par FTP anonyme ainsi que le logiciel de visualisation IZIC.

B. Mourrain a mis un place un serveur WWW *hypermathématèque* orienté vers le calcul formel. Ce service, créé il y a déjà un an, s'est étoffé cette année par une description systématique des logiciels développés dans le projet, par une aide concernant les systèmes de calculs formels utilisés à l'INRIA et par une présentation des séminaires que nous organisons ainsi que des colloques liés à notre activité.

6.2 Formation

6.2.1 Enseignement universitaire

Plusieurs membres du projet participent à l'option *Calcul et Déduction* du DEA de mathématiques de l'Université de Nice-Sophia Antipolis (organisée par M. Gaëtano et R. Marlin). Bernard Mourrain donne un cours intitulé *Problèmes combinatoires, aspects algébriques. applications* (une trentaine d'heures). Stéphane Dalmas donne une dizaine d'heures de cours sur des techniques d'analyses de programmes et de compilation, Marc Gaëtano une vingtaine d'heures sur les l'architecture et la conception des systèmes de calcul formel, Nicole Rostaing-Schmidt un cours de quatre heures sur la différentiation automatique.

En plus de ces enseignements de troisième cycle, les membres de notre projet appartenant à l'université de Nice-Sophia Antipolis assurent des cours en premier et deuxième cycles et à l'ESSI (école d'ingénieurs de l'université de Nice).

6.2.2 Thèses

Le projet est équipe d'accueil de doctorants des formations doctorales SPI et mathématiques, de l'université de Nice-Sophia Antipolis.

Patrick Capolsini a soutenu sa thèse intitulée *Le Logiciel formel de simulation de mécanismes spatiaux* SYGMMAE en février.

Nathalie Eyssette a soutenu sa thèse *Etude de la complexité et transformation symbolique des programmes de calculs de formes linéaires* en décembre.

André Galligo a été président du jury de la thèse de Nathalie Eyssette et de celle de Stéphane Le Ménec, intitulée *Théorie des jeux et programmation avancée appliquées au duel aérien a moyenne distance* (université de Nice-Sophia Antipolis). Il a également participé en tant qu'examinateur au jury de la thèse de François Boulier *Etude et implantation de quelques algorithmes en algèbre différentielle* (thèse d'informatique à université des sciences et technologies de Lille).

6.2.3 Stages

Le projet a accueilli deux stagiaires de seconde année d'ESSI pendant le mois d'août. Stéphane Lavirotte a enrichi les possibilités graphiques de notre système d'animation d'algorithmes *agat* et Stéphane Chasez a travaillé sur l'éditeur de formule mathématiques (mise en place de mécanisme de couper-coller entre applications).

Frédérique Peyrot, stagiaire du DEA robotique et vision a travaillé pendant quatre mois sur le tracé implicite de courbes et de surfaces par une méthode d'exclusion (encadrement commun de B. Mourrain et J.C. Yakoubsohn, professeur à Toulouse).

6.3 Participation à des manifestations

Stéphane Dalmas, Christèle Faure, Marc Gaëtano et Yves Papegay ont assisté à la conférence ISSAC (International Symposium on Symbolic and Algebraic Computation) à Oxford. Olivier Arsac, Stéphane Dalmas et Yves Papegay ont assisté au colloque *Human Interaction for Symbolic Computation* (Amsterdam), où O. Arsac a fait un exposé sur le système d'animation d'algorithmes *agat*.

Jérôme De sousa, Leonbattista Donati, Nicole Rostaing-Schmidt et Bernard Mourrain ont fait un exposé à l'Ecole Polytechnique (au séminaire Aleph et Geode, organisé par Marc Giusti). Jérôme De sousa a présenté un exposé au séminaire Calcul Formel et Calcul Numérique à l'université de Limoges. B. Mourrain a présenté une communication au congrès MEGA (méthodes effectives en géométrie algébriques) à Santander (Espagne) et au colloque *Invariants Methods in Discrete and Computational Geometry* à Curaçao (Antilles néerlandaises). Loïc Pottier a assisté à MEGA et aux journées du PRC Math-Info à Luminy.

A. Galligo, C. Faure, N. Rostaing-Schmidt et E. Hassold ont participé aux journées thématiques de Calcul Formel du département de Mathématiques de Limoges dont le thème étaient *Différentiation automatique et manipulation symbolique de programmes*. A. Galligo et E. Hassold ont présenté des exposés. Nicole Rostaing-Schmidt a participé au *second workshop on Adjoint Applications in Dynamic Meteorology* (à Visegrad, Hongrie). E. Hassold a présenté un exposé aux septièmes journées des thésards en mathématiques appliquées ELF-SNEA et aux journées ELF de mathématiques appliquées.

Yves Papegay a assisté au *Rhine Workshop on Computer Algebra* (Karlsruhe), à trois conférences : *IEEE International Conference on Robotics and Automation* (San Diego), *20th International Conference on Industrial Electronics, Control and Instrumentation* (Bologne, Italie), et *IEEE/RSJ/GI International Conference on Intelligent Robots and Systems* (Munich). Il a également assisté à la conférence européenne des utilisateurs de *Mathematica* (*1994 European Mathematica Conference for Advanced Users*) à Oxford et au *Symbolic Computation in Optics workshop* à Amsterdam.

7 Publications

Thèses

[1] P. CAPOLSINI, *Le Logiciel formel de simulation de mécanismes spatiaux SYGMMAE*, thèse de doctorat, Université de Nice-Sophia Antipolis, février 1994.

[2] N. DELAMARE-EYSSETTE, *Etude de la complexité et transformation symbolique des programmes de calculs de formes linéaires*, thèse de doctorat, Université de Nice-Sophia Antipolis, décembre 1994.

Articles et chapitres de livre

[3] J. DAVENPORT, C. FAURE, «The "Unknown" in Computer Algebra», *Programmirovanie*, 1, 1994.

[4] N. KAJLER, «IZIC 1.0.: an Overview for the User», *Computer Algebra Nederland (CAN) Newsletter*, 13, décembre 1994.

[5] B. MOURRAIN, N. STOLFI, *Invariants methods in Discrete and Computational Geometry*, Kluwer acad. pub., 1994, ch. Computational Symbolic Geometry.

Communications à des congrès, colloques, etc.

[6] P. COMON, B. MOURRAIN, «Decomposition of quantics in sums of powers», *in : Advanced. Signal Processing: Algorithms,Architectures and Implementations*, SPIE, San Diego, 1994.

[7] S. DALMAS, M. GAËTANO, A. SAUSSE, «Distributed Computer Algebra: the Central Control approach», *in : PASCO'94*, World Scientific, p. 104–113, Linz, Austria, septembre 1994.

[8] H. HIRUKAWA, Y. PAPEGAY, T. MATSUI, «A Motion Planning Algorithm for Convex Polyhedra in Contact under Translation and Rotation», *in : proceedings of the 1994 IEEE International Conference on Robotics and Automation*, IEEE Robotics and Automation Society, p. 3020–3028, San Diego, California, mai 1994.

[9] H. HIRUKAWA, Y. PAPEGAY, H. TSUKUNE, «A Motion Planning Algorithm of Polyhedra in Contact for Mechanical Assembly», *in : proceedings of the 20th International Conference on Industrial Electronics Control and Instrumentation*, IEEE Industrial Electronics Society, p. 924–929, Bologna, Italia, septembre 1994.

[10] H. Hirukawa, Y. Papegay, « A Lazy Algorithm for Planning Motions in Contact », *in : proceedings of the 1994 IEEE/RSJ/GI International Conference on Intelligent Robots and Systems*, IEEE Industrial Electronics Society, p. 2152–2159, Munich, Germany, septembre 1994.

[11] B. Mourrain, « Enumeration problems in Geometry, Robotics and Vision », *in : Proceedings of MEGA-94*, Santander (Spain), avril 1994.

[12] N. Rostaing-Schmidt, E. Hassold, « Basic functional representation of programs for automatic differentiation in the Odyssée system », *in : Proceedings of the workshop on High Performance Computing in the Geosciences*, F. L. Dimet (réd.), Kluwer Academic Publishers, NATO ASIE SERIES, Les Houches (France), 1994.

Rapports de recherche et publications internes

[13] R. Avitzur, O. Bachmann, N. Kajler, « From Honest to Intelligent Plotting », *rapport de recherche n° 4*, RIACA, Amsterdam, décembre 1994.

[14] S. Dalmas, M. Gaëtano, A. Sausse, « ASAP : a protocol for symbolic computation systems », *Rapport Technique n° 162*, INRIA, mars 1994.

[15] C. Faure, « Objets Conditionnels et objets inconnus », *Rapport de recherche n° 2298*, INRIA, juillet 1994.

[16] J. Grimm, « Implémentation en C de l'arithmétique de Sisyphe », *Rapport Technique n° 168*, INRIA, novembre 1994.

[17] F. Peyrot, *Représentation explicite des courbes et surfaces implicites*, Mémoire, Université de Nice-Sophia Antipolis, 1994.

[18] L. Pottier, « Gröbner bases of toric ideals », *Rapport de recherche n° 2224*, INRIA, mars 1994.

8 Abstract

The SAFIR project is a joint project between INRIA, the university of Nice-Sophia Antipolis and CNRS. Our research theme is symbolic and algebraic computation. We are mainly interested in the following aspects of computer algebra: the design of symbolic computation systems, algebraic algorithms and their complexities and applications of symbolic computation (robotics, computer vision, design and simulation).

The main research topics of this year are distributed computer algebra (with the Central Control), fully correct computations with mathematical expressions, high-quality graphical interfaces, the visualisation of surfaces, the design of a specialized computer algebra system for

control theory (HYPERION), the theory and practice of symbolic computations in geometry and polynomial systems solving (with applications to computer vision and robotics), the study of toric Gröbner basis and their applications to lattices and polytopes, automatic differentiation (of FORTRAN programs) with the *Odyssée* system and the application of computer algebra to the study and simulation of various processes.

Table des matières

1 Composition de l'équipe 1

2 Présentation du projet 2

3 Actions de recherche 4

3.1 Aspects logiciels du calcul formel 4

3.1.1 Calcul formel distribué 4

3.1.2 Expressions et formules 5

3.1.3 Interfaces 7

3.1.4 Reconnaissance de formules planes 8

3.1.5 Bibliothèque et constructeurs de structures mathématiques effectives 9

3.1.6 A# 9

3.1.7 Visualisation de courbes et de surfaces, IZIC . . . 10

3.1.8 Hyperion 11

3.2 Algèbre, géométrie et complexité 12

3.2.1 Géométrie Algébrique, Robotique et Vision 12

3.2.2 Manipulation symbolique d'objets géométriques. . 12

3.2.3 Résolution de systèmes polynômiaux 13

3.2.4 Bases de Gröbner toriques 13

3.2.5 Plus petits vecteurs d'un réseau, cas des réseaux de Craig 15

3.2.6 Borne inférieure temps-espace 16

3.2.7 Etude de la complexité et transformation symbolique de programmes de calcul de formes linéaires 16

3.2.8 Algorithmes de calcul des réductions de Hermite et de Smith 17

3.2.9 Décomposition primaire d'idéaux 17

3.2.10 Etude de la topologie des courbes planes 18

3.2.11 Singularités des vues de surfaces éclairées 19

3.3 Applications du calcul formel, liens formel-numérique . . . 20
3.3.1 Utilisation du calcul formel pour la modélisation et la simulation de phénomènes physiques 20
3.3.2 Dérivation automatique 21
3.3.3 Approximation de fonctions de transfert 23

4 Actions industrielles 24
4.1 Hyperfréquences 24
4.2 GENIE 24
4.3 Chimie 25
4.4 CNES 25
4.5 EDF 25
4.6 Optique 25

5 Actions nationales et internationales 26
5.1 Actions internationales 26
5.1.1 Invitations 26
5.1.2 PoSSo 26
5.1.3 Réseau HCM SAC 26
5.1.4 Réseau HCM SOL 26
5.1.5 RIACA 27
5.1.6 Océanographie 27
5.1.7 OpenMath 27
5.2 Actions nationales 27
5.2.1 Invitations 27
5.2.2 PRC et GDR 28
5.2.3 Organisations de séminaires 28

6 Diffusion des résultats 28
6.1 Diffusion de produits 28
6.2 Formation 28
6.2.1 Enseignement universitaire 28

6.2.2 Thèses 29
6.2.3 Stages 29
6.3 Participation à des manifestations 30

7 Publications **31**

8 Abstract **32**