Calcul Formel : Tendances et progrès récents

Daniel Lazard
LIP6, Université Paris VI, 75252 Paris CEDEX 05
Email : Daniel.Lazard@lip6.fr

RÉSUMÉ. *Les tendances de l'évolution des logiciels de calcul formel sont présentées, telles qu'elles sont ressenties par l'auteur à l'aube de l'an 2000. Deux points sont paticulièrement développés : La multiplications des logiciels spécialisés performants et la résolution des systèmes d'équations polynomiales, où des progrès remarquables sont en cours.*

ABSTRACT. *Tendencies of software development in symbolic computation (computer algebra) are presented as viewed by the author around year 2000. Two points are especially developed : The multiplicity of efficient specialized software, and polynomial system solving were a breakthrough is in course.*

MOTS-CLÉS : *calcul formel, système de calcul formel, systemes d'équations polynomiales, systèmes polynomiaux, équations polynomiales, inéquations polynomiales.*

KEY WORDS : *computer algebra, symbolic computation, algebraic computation, polynomial system solving, polynomial equations, polynomial inéquations.*

1. Introduction.

La grande diffusion des logiciels MATHEMATICA [17][1] et MAPLE [16] a fortement popularisé le calcul formel. Ces progiciels associent un langage de programmation, une technique de représentation des données et une bibliothèque d'algorithmes, en vue de la manipulation des expressions mathématiques. Contrairement au calcul numérique, en calcul formel, les variables et paramètres ne sont pas nécessairement remplacés par des valeurs numériques, et la plupart des calculs sont exacts (i. e. sans approximation numérique). Ainsi, en calcul formel, le nombre π est un symbole auquel sont associés un certain nombre de propriétés de simplification (par exemple $\sin(\pi) = 0$) et un programme de calcul d'un nombre arbitraire de décimales ; le remplacement de π par une valeur approchée ne se fait qu'à la demande explicite de l'utilisateur.

Depuis quelques années ces logiciels sont devenus irremplaçables pour l'expérimentation mathématique. Pour chercher quelle est la meilleure forme pour exprimer une loi physique, comment l'adapter au contexte dans lequel on travaille, quelles re-

1. La plupart de nos références bibliographiques sont des adresses de sites web. C'est inhabituel, mais apparaît comme la meilleure solution pour les logiciels.

1e soumission à TSI, version 2 du 23 septembre 1999.

lations sont susceptibles d'être vérifiées entre les variables, ..., les ingénieurs et chercheurs sont de plus en plus en plus nombreux à utiliser quotidiennement des outils de calcul formel.

Ces systèmes de calcul formel, dits généralistes, continuent à évoluer par l'introduction de nouveaux algorithmes augmentant les possibilités ou améliorant les performances. Cependant, on peut considérer qu'ils sont aboutis et qu'ils n'évolueront plus guère : Ils sont parfaitement adaptés à l'expérimentation et aux calculs pour lesquels l'utilisateur ne connaît pas exactement la forme que doit avoir le résultat. Mais l'extrême diversité des formes que peut prendre une même expression mathématique rend impossible la compilation des programmes des utilisateurs, ce qui rend ces logiciels inadaptés pour le calculs "d'exploitation" où on connaît la forme du résultat et où le temps de calcul devient un paramètre important. Pour la même raison, ils sont mal adaptés à de nombreux algorithmes récents : on a pu constater des rapports de temps de calcul supérieur à 10 000 entre un calcul de base de Gröbner effectué en Maple et le même calcul effectué avec un programme écrit en C. Mais cette relative lenteur semble actuellement inévitable, si l'on veut conserver l'énorme variété des possibilités et des domaines d'application qu'offrent des logiciels comme Maple et Mathematica.

L'objectif de cet article est de montrer quelques tendances de la recherche actuelle en calcul formel, et plus spécialement dans les aspects logiciels de ce domaine. Il se divise en deux parties : La première est consacrée aux systèmes de calcul formels et notamment à la multiplicationgrande variété des systèmes spécialisés et aux problèmes d'interface que cela pose. La deuxième est consacrée à la résolution des systèmes d'équations polynomiales, un domaine où des progrès importants, tant logiciels qu'algorithmiques, sont en cours de développement en France. Ces progrès rendent accessibles, voire faciles, des problèmes qu'il n'était même pas envisageable d'aborder avec les outils antérieurs. Ceci constitue donc une avancée scientifique majeure, selon l'avis, peut-être partial, de l'auteur.

Le lecteur souhaitant des informations détaillées pourra utilement consulter Journal of Symbolic Computation, la principale revue du domaine, qui publie régulièrement des numéros spéciaux consacrés à des sujets spécialisés (le dernier est consacré aux systèmes polynomiaux). Pour compléter le panorama de la recherche en calcul formel, il faut ajouter les actes publiés par l'ACM du principal colloque régulier du domaine, ISSAC (International Symposium on Symbolic and Algebraic Computation). Cependant de nombreux articles de calcul formel sont publiés dans des revues variées, ainsi que dans les actes de divers colloques, souvent spécialisés dans un aspect particulier du calcul formel.

2. Systèmes de calcul formel spécialisés.

A côté des systèmes généralistes dont il a été parlé plus haut, de nombreux systèmes spécialisés ont été développés depuis les origines du calcul formel. Pendant longtemps, ce n'étaient que des outils d'expérimentation qui n'étaient guère diffusés en dehors des équipes qui les développaient. Mais depuis quelques années, un grand

nombre de ces logiciels ont atteint le stade d'une large diffusion internationale.

Ces systèmes sont généralement spécialisés dans une branche particulière des mathématiques, et obtiennent des performances remarquables en associant des techniques fines de programmation (le plus souvent en C ou C++) à des outils mathématiques mettant en œuvre toutes les connaissances du domaine considéré.

Bien que les énumérations soient souvent indigestes, il n'est pas inutile de citer quelques uns de ces systèmes spécialisés, ne serait-ce que pour montrer la variété des domaines mathématiques concernés.

— Théorie des groupes : C'est dans ce domaine que sont apparus les premiers systèmes spécialisés. Ceux qui dominent le "marché" actuel sont GAP [8] et MAGMA [15]. En raison de la multiplicité des approches utilisées en théorie des groupes et des nombreux domaines utilisant cette théorie, ces deux systèmes ont plus ou moins tendance à devenir généralistes.

— Théorie des nombres et arithmétique : Le système français GP/PARI [10] et le système allemand KANT/KASH [11], plus récent, dominent ce domaine.

— Combinatoire : Le système français ACE [1] traite de différents aspects de la combinatoire qui ne sont pas traités par GAP et MAGMA.

— Géométrie algébrique : Dans ce domaine, le système le plus connu est MACAULAY [13] et son successeur plus moderne et plus convivial MACAULAY2 [14]. Mais il faut citer aussi le système allemand SINGULAR [22], plus spécialisé en géométrie analytique et théorie des singularités et le système italien COCOA [5], plus orienté vers l'algèbre commutative.

— Topologie algébrique : Dans ce domaine extrêmement spécialisé, le logiciel grenoblois KENZO [12] est remarquable car, en alliant une programmation fonctionnelle élaborée (écrite en LISP) et une réécriture constructive presque complète de cette théorie mathématique, il permet des calculs qui semblaient impossibles aux spécialistes.

— Systèmes d'équations polynomiales : Nous reviendrons plus loin sur cette question, où les logiciels leaders sont GB [9] et REALSOLVING [20] et leurs successeurs FGB et RS.

Le lecteur averti remarquera qu'il y a un domaine important pour lequel aucun système spécialisé n'a été cité : Il s'agit des équations et systèmes différentiels et aux dérivées partielles. Dans ce domaine il y a eu ces dernières années de nombreux progrès très importants, mais la grande variété des méthodes d'approche et la grande variété des formes d'équation différentielles considérées font, qu'à l'heure actuelle, aucun système (spécialisé ou généraliste) ne regroupe l'ensemble de ces progrès ; tous ces progrès mériteraient un autre article en complément de celui-ci.

La variété des notions mathématiques, souvent sophistiquées, impliquées par tous ces différent logiciels crée un effet "tour de Babel" : Il n'est plus possible pour un individu de maîtriser l'ensemble du calcul formel.

Par ailleurs, chacun des logiciels considérés possède sa propre interface utilisateur, ce qui pose des problèmes sérieux quand on a besoin de fonctionnalités relevant de plusieurs de ces systèmes (pour résoudre un problème particulier, l'auteur de ces

lignes a eu de besoin de cinq systèmes différents) : Pour transférer des données d'un système à l'autre, les utilisateurs doivent, soit les convertir à la main, soit programmer eux même un traducteur.

Pour résoudre ce problème de communication entre logiciels, il n'est pas possible de recourir à une norme uniforme de représentation des données, comme cela a été fait pour les nombres flottants (norme IEEE 754) : La complexité des algorithmes est trop dépendante de la représentation des données pour qu'une seule représentation puisse être adaptée à toutes les situations. Une autre approche est en train de voir le jour : il s'agit du protocole standardisé de transmission de données OPENMATH [19]. C'est un projet très prometteur, mais trop jeune pour qu'on sache si cette solution s'imposera à long terme.

Il faut citer également deux systèmes généralistes qui, par des moyens différents tentent de résoudre ces problèmes de communication : Le système AXIOM [3] est un système généraliste basé sur un langage de programmation fortement typé, permettant une réelle compilation et un contrôle de la représentation des données par l'utilisateur. Son nouveau compilateur ALDOR permet la génération de modules autonomes et l'importation de modules écrits dans d'autres langages. Cette approche est prometteuse ; malheureusement, ces produits tardent à acquérir une efficacité et une ergonomie susceptibles d'entraîner l'adhésion de la communauté des utilisateurs et des développeurs.

Le système généraliste allemand MUPAD [18] a une syntaxe très voisine de celle de MAPLE tout en étant gratuit pour l'utilisation non commerciale, ce qui lui assure un développement rapide. Sa conception permet d'inclure des modules écrits dans d'autres langages, ce qui lui permet d'incorporer partiellement ou totalement plusieurs des logiciels spécialisés cités.

3. Systèmes d'équations polynomiales.

Dans ce foisonnement de progrès récents, nous avons choisi de développer plus spécialement le domaine de la résolution des systèmes d'équations (et inéquations) polynomiales pour plusieurs raisons.

D'abord, c'est un problème fondamental qui a des applications dans tous les domaines, et à ce titre devrait être une fonctionnalité indispensable à tout système généraliste de calcul formel. Mais il s'agit d'un problème difficile tant au plan mathématique que du point de vue de la théorie de la complexité, pour lequel il n'y avait pas de solution satisfaisante, jusqu'à tout récemment.

Ensuite, des progrès spectaculaires sont en cours : Il y a une dizaine d'années, les logiciels disponibles ne permettaient de résoudre que des problèmes académiques de très petite taille presque tous accessibles au calcul manuel ; au contraire, les logiciels récents, et surtout les prototypes en cours de développement FGB [9] de J.-C. Faugère et RS [20] de F. Rouillier permettent d'aborder des problèmes réels de taille importante ; ces logiciels ont déjà permis de résoudre des problèmes auparavant inaccessibles dans des domaines aussi variés que la compression d'image, la robotique, la

cryptographie ou la mécanique céleste.

Avant d'esquisser une description de ces progrès, il n'est pas inutile de préciser le problème considéré. Une équation polynomiale est une équation de la forme $P = 0$ où P est un polynôme à plusieurs variables à coefficients entiers ou rationnels, par exemple $x^2+y^2-1 = 0$; le plus souvent, on omet le "$= 0$", et on considère P comme une équation. Une inéquation polynomiale est de la forme $P > 0$, $P < 0$, $P \geq 0$ ou $P \leq 0$. Un système d'équations (et inéquations) est une conjonction d'équations (et inéquations) dont on recherche les solutions simultanées. Quand on considère un système d'équations, on en recherche généralement toutes les solutions complexes, mais quand des inéquations apparaissent, on se limite implicitement aux solutions réelles.

Il faut noter que la limitation aux coefficients rationnels est beaucoup moins restrictive qu'il peut sembler : $\sqrt{2}$ peut être remplacé par l'inconnue r vérifiant $r^2-2 = 0$ et $r > 0$. De même, de nombreux systèmes trigonométriques se ramènent immédiatement à des systèmes polynomiaux : si x est un angle inconnu, il suffit de remplacer $\cos(x)$ et $\sin(x)$ par deux inconnues c et s vérifiant $c^2 + s^2 = 1$.

Ayant spécifié l'entrée du problème (un système d'équations et d'inéquations), il faut maintenant définir la sortie (les solutions recherchées). Ces solutions sont des listes de nombres réels ou complexes (un par inconnue), et peuvent être approchées par des listes de "nombres flottants". Si cette formulation des solutions est souvent nécessaire in fine, elle est très souvent inadaptée pour plusieurs raisons :

— Dans de nombreux cas, l'ensemble des solutions est infini, comme dans le système réduit à $x^2 + y^2 - 1 = 0$.

— Même si l'ensemble des solutions est fini, il est souvent trop grand pour être pratiquement utilisable sous forme d'approximation flottante. Ainsi, le système

$$\begin{gathered}
a+b+c+d+e+f+g=0\\
ab+bc+cd+de+ef+fg+ga=0\\
abc+bcd+cde+def+efg+fga+gab=0\\
abcd+bcde+cdef+defg+efga+fgab+gabc=0\\
abcde+bcdef+cdefg+defga+efgab+fgabc+gabcd=0\\
abcdef+bcdefg+cdefga+defgab+efgabc+fgabcd+gabcde=0\\
abcdefg=1.
\end{gathered}$$

a 924 solutions. Un autre inconvénient des approximations flottantes est qu'elles font disparaître des informations structurelles qui peuvent être indispensables.

Aussi, l'ensemble des solutions d'un système est généralement présenté comme la réunion des solutions d'un ou plusieurs systèmes "plus simples" en ce sens que l'information pertinente en est facilement extractible.

Ainsi, la base de Gröbner d'une famille de polynômes est une autre famille (généralement beaucoup plus grosse) qui a les mêmes solutions, et dont on peut déduire facilement le nombre de solutions (si elle sont en nombre fini) ou la dimension de l'espace qu'elles forment (s'il y en a une infinité).

Ce n'est pas le lieu de décrire plus précisément ici ce qu'est une base de Gröbner. Il suffit de dire que le calcul de bases de Gröbner est un module indispensable dans

tout logiciel de résolution de systèmes d'équations polynomiales, et de renvoyer le lecteur intéressé à [2], [4] ou [6]. On peut résumer les progrès accomplis au cours de la dernière décennie avec l'exemple en 7 variables ci-dessus : Ce fut un grand succès, en 1991, que d'obtenir en 110 heures de calcul la preuve que ce système possède 924 solutions. Auparavant, le calcul de la base de Gröbner du système analogue en 6 variables avait nécessité 82 heures en 1987. Le logiciel GB de 1994 de J.-C. Faugère nécessite respectivement 5 heures et 3 secondes pour effectuer ces calculs. Le tout récent logiciel FGB du même auteur calcule ces mêmes bases de Gröbner en 40 et 0,3 secondes respectivement ; il calcule également (en 18 jours CPU) la base de Gröbner du système analogue en 9 variables, base qui est constituée de 800 polynômes ayant environ 1000 termes et des coefficients d'environ 1000 chiffres décimaux ; la taille de ce résultat (1,6 Goctet) le rend à peu près inexploitable, et montre que l'efficacité actuelle du calcul des bases de Gröbner est proche de l'optimal.

Tant par leur taille que par la difficulté d'en extraire les valeurs numériques des solutions, les bases de Gröbner ne constituent pas une sortie satisfaisante pour un solveur d'équations polynomiales. L'expression des solutions qui s'avère la plus satisfaisante consiste en ce que l'on appelle des systèmes triangulaires. Il s'agit de systèmes de la forme

$$\begin{aligned}
&f_1(X_1,\ldots,X_{k_1}) = 0\\
&f_2(X_1,\ldots,X_{k_1}\ldots,X_{k_2}) = 0\\
&\ldots\\
&f_h(X_1,\ldots,X_{k_1},\ldots,X_{k_2},\ldots,X_{k_h}) = 0
\end{aligned}$$

où les f_i sont des polynômes qui ne dépendent que des inconnues apparaissant entre parenthèses et tels que $k_1 < k_2 < \cdots < k_h$. Dans le cas où l'ensemble des solutions est fini, on a $k_i = i$; dans le cas contraire, les variables autres que les X_{k_i} sont considérées comme des paramètres arbitraires (sous une certaine condition trop technique pour être décrite ici).

Ces systèmes triangulaires sont bien adaptés au calcul numérique des solutions, car, une fois les paramètres fixés, il suffit de résoudre successivement h équations en une inconnue. Très souvent, tous les f_i sauf f_1 sont linéaires en X_{k_i}, ce qui fait qu'il n'y a qu'une seule équation non linéaire à résoudre ; si ce n'est pas le cas on peut très souvent s'y ramener par un changement (aléatoire) linéaire de variables.

Exprimer les solutions d'un système polynomial sous forme de systèmes triangulaires est une opération qui peut se faire soit directement, soit en passant par les bases de Gröbner. Dans les deux cas, il y a eu d'importants progrès au cours des trois dernières années, et il est maintenant assez courant d'obtenir des systèmes triangulaires tels que f_1 soit un polynôme de degré supérieur à 500, avec des coefficients de plusieurs dizaines de chiffres. Néanmoins, la mise en œuvre logicielle de ces méthodes n'a pas encore rattrapé l'efficacité atteinte par les calculs de base de Gröbner, et il reste beaucoup à faire pour obtenir un solveur combinant efficacement les deux approches.

Dans beaucoup d'applications, le nombre des solutions est fini, et il est nécessaire de calculer numériquement les solutions réelles satisfaisant certaines conditions d'inégalité (la plupart des variables de la physique, telles les distances, les masses, l'énergie, ..., doivent être positives). En raison du haut degré et de la grande taille des

coefficients des équations qui apparaissent dans les systèmes triangulaires obtenus, les méthodes numériques traditionnelles ne permettent généralement pas de calculer ces solution réelles. Aussi, a-t-il fallu développer de nouvelles méthodes pour déterminer le nombre de solution réelles, des encadrements certifiés de ces solutions et les signes que prennent certaines expressions quand les valeurs des solutions y sont substituées. La réalisation informatique la plus avancée dans ce domaine est constituée par les logiciels RealSolving et RS [20] de F. Rouillier.

Quand un système d'équations n'a qu'un nombre fini de solutions complexes, on dispose ainsi de moyens efficaces pour déterminer toutes les solutions réelles ainsi que leurs signes. Dans le cas d'une infinité de solutions, ces problèmes ont permis de résoudre certains problèmes notamment en compression d'image [7]. Cependant, bien d'autres problèmes, dont certains ont été résolus à la main, résistent encore aux algorithmes et implantations actuellement disponibles.

Le plus célèbre de ces problèmes est le « piano movers problem » qui consiste à déterminer si un piano à queue peut être déménagé à travers une cage d'escalier. Il s'agit bien de résolution de systèmes polynomiaux : Si la forme du piano et de la cage d'escalier sont décrites par des fonctions polynomiales, une position admissible du piano est une solution d'un certain système d'équations et d'inéquations. Le déménagement est possible si la position de départ et la position d'arrivée sont dans la même composante connexe de l'ensemble des solutions. En théorie il existe des algorithmes pour les problèmes de ce type [21]. En pratique, les algorithmes existants ne sont pas assez efficaces pour résoudre le problème plan (passage à plat dans un couloir) pourtant beaucoup plus simple.

Il reste donc beaucoup à faire!

Références

[1] ACE : http://phalanstere.univ-mlv.fr/~ace/

[2] ADAMS, W. W., et LOUSTAUNAU, P., *An introduction to Gröbner bases.* Graduate Studies in Mathematics, 3. American Mathematical Society, Providence, RI, 1994. xiv+289 pp. ISBN: 0-8218-3804-0

[3] AXIOM : http://www.nag.co.uk/symbolic_software_more.asp

[4] BECKER, T. et WEISPFENNING, V. , *Gröbner bases. A computational approach to commutative algebra.* In cooperation with Heinz Kredel. Graduate Texts in Mathematics, 141. Springer-Verlag, New York, 1993. xxii+574 pp. ISBN: 0-387-97971-9

[5] COCOA : http://cocoa.dima.unige.it/system/

[6] COX, D., LITTLE, J. et O'SHEA, D., *Ideals, varieties, and algorithms. An introduction to computational algebraic geometry and commutative algebra.* Second

edition. Undergraduate Texts in Mathematics. Springer-Verlag, New York, 1997, xiv+536 pp. ISBN: 0-387-94680-2.

[7] FAUGÈRE, J.-C., MOREAU DE SAINT MARTIN, F. et ROUILLIER, F. « Design of regular non separable bidimensional wawelets using Gröbner basis techniques ». *IEEE SP Transactions Special Issue on Theory and applications of Filter banks and wawelets*, SP:30, 1997.

[8] GAP : http://www-gap.dcs.st-and.ac.uk/~gap/

[9] GB and FGB : http://posso.lip6.fr/Gb/

[10] GP/PARI : http://www.math.u-bordeaux.fr/A2X/Logiciels.html

[11] KANT/KASH : http://www.math.TU-Berlin.DE/~kant/

[12] KENZO : http://www-fourier.ujf-grenoble.fr/~sergerar/Kenzo/

[13] MACAULAY : http://www.math.columbia.edu/~bayer/Macaulay.html

[14] MACAULAY2 : http://www.math.uiuc.edu/Macaulay2/

[15] MAGMA : http://www.maths.usyd.edu.au:8000/u/magma/

[16] MAPLE : http://www.maplesoft.com/

[17] MATHEMATICA : http://www.mathematica.com/

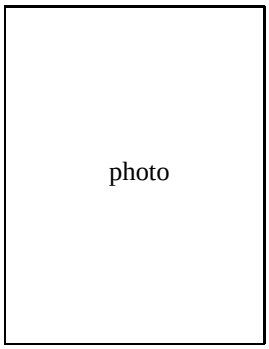
[18] MUPAD : http://www.mupad.de/

[19] OPENMATH : http://www.openmath.org/

[20] REALSOLVING and RS : http://www.loria.fr/~rouillie/software.html

[21] SCHWARTZ, J. T. et SHARIR, M., « On the "piano movers" problem. II. General techniques for computing topological properties of real algebraic manifolds ». *Adv. in Appl. Math.* vol. 4, nº 3, p. 298–351, 1983.

[22] SINGULAR : http://www.mathematik.uni-kl.de/~zca/Singular/



Daniel Lazard *est directeur du LIP6 (Laboratoire d'Informatique de Paris 6 (UMR 7606). Après un début de carrière comme mathématicien (algèbre commutative) il a commencé à utiliser le calcul formel vers 1970 (logiciel* FORMAC*) et a progressivement orienté la totalité de son activité de recherche dans ce domaine, ce qui l'a amené à changer de discipline : Après avoir été professeur de mathématiques à Poitiers de 1972 à 1983, il est, depuis, professeur d'informatique à l'Université Pierre et Marie Curie (Paris VI). Par la mise sur pied et la direction du* GRECO *de Calcul Formel de 1982 à 1990, il a certainement contribué a faire de la France une des nations dominantes pour la recherche en Calcul Formel. Bien que s'interessant à de nombreux aspects du calcul formel, son activité de recherche est, depuis 20 ans, principalement consacrée aux systèmes d'équations polynomiales sous leurs divers aspects et à leurs applications variées*