

MAXIMA et les logiciels libres de calcul formel

MAXIMA and the free softwares for symbolic computation

François OLLIVIER

LIX UMR CNRS-Ecole polytechnique 7161, 91128 Palaiseau CEDEX

Mots-clefs : Calcul formel, calcul symbolique, logiciel libre, interface formel-numérique

Key-words: Computer algebra, symbolic computation, free software, symbolic-numerical interface

MAXIMA est un système de calcul formel généraliste sous licence GPL qui permet de résoudre simplement des problèmes de modélisation en vue de la simulation numérique.

L'essentiel

Le calcul formel permet de manipuler des formules et non simplement des nombres, ce qui en fait un outil puissant pour de nombreux secteurs scientifique et technique : théorie des codes, automatique, théorie du signal, modélisation, etc.

Maxima est un système de calcul formel généraliste rustique mais stable et offrant au modélisateur des capacités amplement suffisantes pour produire à peu de frais du code numérique. Sa disponibilité sous licence GPL permet de l'utiliser commodément dans des projets logiciels à long terme.

Synopsis

Computer algebra allows the manipulation of formulas and not only of numbers; it is thus a powerful tool in various areas of science and technology: coding theory, control theory, signal theory, modeling, and so on.

Maxima is a general purpose computer algebra system, rustic but stable and offering for modeling far enough tools to produce with little effort numerical code. Its availability under GNU license allows an easy use in long term programming projects.

Résumé

Le calcul formel manipule des formules ou des nombres de taille arbitraires. Ceci le distingue du calcul numérique traditionnel. L'utilisation conjointe du calcul formel et du calcul numérique permet de développer des outils logiciels particulièrement puissants, utiles dans de nombreux domaines scientifiques et techniques.

Pour de tels projets, les codes propriétaires sont souvent un obstacle, tant pour l'écriture des programmes, leur diffusion auprès de partenaires que pour la maintenance à long terme.

Le choix d'un logiciel libre est donc un atout important. Maxima est le seul système généraliste de calcul formel libre disposant d'une communauté crédible d'utilisateurs qui puisse en assurer la pérennité. Après avoir dressé un panorama historique des principaux logiciels de calculs formel libres et/ou gratuits, nous illustrerons son utilisation pour traiter un problème de modélisation.

Abstract

Computer algebra manipulates formula and numbers of arbitrary size. This is the great difference with classical numerical computation. The joint use of computer algebra and numerical computation allows the development of powerful software, useful in various areas of science and engineering. For such projects, proprietary software is a drawback for writing the programs, sharing programs with partners and for maintaining them at long term. The choice of free software is thus a great advantage. Maxima is the only all purpose computer algebra system with a big enough users community. After an historical survey of the main free computer algebra systems, we will illustrate its use for solving a modeling problem.

L'auteur

François OLLIVIER, polytechnicien, débute au Laboratoire Laurent Schwartz, avant de rejoindre le LIX à sa création. Il soutient sa thèse en 1990, sur la résolution de systèmes algébro-différentiels intervenant en automatique théorique Entré au CNRS en 1991, il a participé à l'Unité Mixte de Service CNRS-X MEDICIS, puis rejoint le LIX, après la disparition de cette structure. Il est membre du projet INRIA-Futurs ALIEN et préside l'association GAGE, destinée à promouvoir le calcul formel.

1. Introduction

Le calcul formel, ou calcul symbolique, est un champ scientifique pluridisciplinaire à la frontière entre les mathématiques effectives et l'informatique. Son objet est la manipulation sur ordinateur de formules mathématiques, de taille arbitraire, et non uniquement de nombres entiers ou flottants de taille fixe, comme le fait le calcul numérique traditionnel.

Il s'est développé depuis les années 1960 pour répondre aux besoins des mathématiques, mais aussi de la physique, en profitant du développement des ordinateurs. Selon les pays, son développement fut assuré plutôt par des mathématiciens, des informaticiens ou des physiciens théoriciens. On distingue des systèmes généralistes, qui tentent de couvrir avec plus ou moins de succès, l'essentiel du champ de la discipline : résolution de systèmes algébriques, intégration en formules closes (c'est-à-dire à l'aide d'expressions utilisant les fonctions élémentaires classiques : exponentielle, logarithmes, fonctions trigonométriques, ...), factorisation, etc. et des systèmes spécialisés, qui se proposent d'aborder avec plus d'efficacité un domaine précis.

Les principaux systèmes généralistes actuellement disponibles sont Maple et Mathematica. Il s'agit de systèmes commerciaux. De nouvelles versions peuvent inclure certains progrès des algorithmes, mais outre le coût des licences, l'absence d'accès aux sources et l'instabilité des implantations successives peuvent compromettre leur utilisation dans le cadre du développement de projets logiciels, par exemple si l'on souhaite utiliser les fonctionnalités du calcul formel pour produire automatiquement du code numérique en C ou en FORTRAN.

Pour cela, on pourra préférer un logiciel libre, gratuit, mais surtout dont les sources sont accessibles, ce qui garantit la pérennité des investissements de ses utilisateurs.

2. Panorama historique de Maxima et du calcul formel libre.

Ce système a été développé au MIT, dans le cadre du projet MAC (Machine Aided Cognition), de la fin de 1967 à 1982, sous le nom de Macsyma, grâce aux contributions du DOE (Department of Energy), de la NASA, de l'ARPA (Advanced Research Project Agency) et de divers contributeurs industriels. A l'époque, ce projet relevait de « l'intelligence artificielle ». Il a été le système de référence des années 1970 et inspiré, tant par ses possibilités que par ses limites, les concepteurs des systèmes de la génération suivante, comme Maple ou Mathematica, créés dans les années 1980. Développé en LISP, il n'était disponible que sur des ordinateurs puissants et peu répandus, et accessible à une communauté restreinte d'utilisateurs via le précurseur d'Internet : Arpanet. En 1982, le MIT vend les droits de Macsyma, qui devient une marque déposée par Symbolics. Toutefois, le DOE, l'un des principaux contributeurs publics, impose le dépôt d'une copie du code source dans un centre de distribution gouvernemental états-unien : le NESCS (National Energy Software Center) : c'est le DOE-Macsyma.

Maxima est une variante développée et maintenue par William Frederick Schelter à partir de 1982. En 1998, il obtient du DOE la permission de le distribuer sous licence GPL (GNU General Public Licence). Depuis sa mort en 2001, le logiciel est maintenu par un groupe d'utilisateurs et de développeurs. On peut trouver plus de détails historiques et une réflexion de fond sur les avantages et inconvénients de Macsyma dans le texte de Fateman [3].

Parmi les systèmes de cette génération, on peut citer REDUCE, particulièrement orienté vers les besoins de la physique et qui est toujours disponible en version commerciale. MuPAD (MultiProcessing Algebra Data), développé en Allemagne à l'université de Paderborn depuis 1989 n'est pas un logiciel libre. Les organismes de recherche et d'enseignement à but non lucratif peuvent toutefois disposer d'une licence gratuite.

Il existe enfin un système généraliste original, AXIOM, disponible depuis peu en version libre, sous une version modifiée de la licence BSD (Berkley System Distribution), non « copyleftée ». Les différentes parties du code faisant parfois l'objet de restrictions particulières, on ne peut qu'inciter les utilisateurs à les étudier de près pour un usage professionnel (voir <http://savannah.nongnu.org/projects/axiom>). Ce système a été développé par IBM à partir de 1971, sous le nom de Scratchpad. Il s'agit donc d'un système ancien, mais c'est en même temps, dans sa conception, un système très novateur. L'utilisation du typage permet l'écriture d'algorithmes, non pour un type particulier d'objets mathématiques (par exemple : entiers relatifs ou polynômes) mais pour toute une catégorie (par exemple : anneaux). Cette conception puissante et ambitieuse en rend le maniement difficile. En 1990, IBM a vendu les droits de Scratchpad à NAG (Numerical Algorithms Group), fournisseur bien connu de programmes numériques qui souhaitait compléter son offre par un logiciel formel. Faute de succès commercial, AXIOM a été retiré du marché en 2001 et NAG en a autorisé la diffusion comme logiciel libre.

Signalons enfin l'existence de logiciels libres spécialisés de grande qualité : GP/Pari, conçu pour des calculs rapides en arithmétique (factorisations, théorie algébrique des nombres, courbes elliptiques... d'une grande importance en théorie des codes notamment), développé à l'université de Bordeaux et disponible sous licence GPL ; Macaulay 2, spécialisé dans la résolution de systèmes algébriques, également sous GPL, etc.

Une liste de logiciel de calculs formel « open source » est accessible sur : http://web.usna.navy.mil/~wdj/opensource_math.html On peut aussi profiter de la synthèse [1] pour en savoir davantage sur le calcul formel libre.

L'essentiel des logiciels de calcul formel est encore accessible gratuitement, sur simple demande, pour des travaux de recherche via le centre de calcul MEDICIS (<http://medicis.polytechnique.fr/medicis>), dont la pérennité n'est pas certaine. Il faut vite en profiter !

3. Caractéristiques et possibilités de Maxima.

Maxima peut être obtenu à l'adresse <http://maxima.sourceforge.net/> . La version courante est la 5.9.1 qui est disponible sous Windows et Linux. Outre la documentation standard en anglais, on trouve à cette adresse une documentation et un tutorial en espagnol. Signalons aussi l'intéressante documentation en français accessible à l'adresse <http://www.maxima.fr.st/>

La priorité pour le développement est de supprimer les bugs (à partir de la version 5.9.3). Ceci peut sembler troublant, mais il faut garder à l'esprit que le développement a été figé durant près de 20 ans, William Schelter n'ayant pu que maintenir vivant le code hérité du DOE. À bien des égards, Maxima est rustique. Il ne distingue pas (en principe !) majuscules et minuscules, un inconvénient qui devrait disparaître avec la version 5.9.2. À plus long terme,

une révision complète de la documentation est prévue avant la 6.0 (donc au cours des versions 5.9.x) et une amélioration des routines graphiques avec la 6.1.

Maxima est implanté en LISP. Autrefois, Macsyma reposait sur MAC Lisp. Le travail de William Schelter a principalement consisté en un portage en AKCL (Austin Kyoto Common Lisp), devenu GCL (GNU Common Lisp). Les utilisateurs les plus compétents peuvent utiliser cette particularité pour écrire directement leur code en LISP. Mais MAXIMA possède aussi son propre langage, dont la structure n'est pas très différente de celle de Maple. Ce petit exemple (voir fig. 1) montre que dans la version 5.9.1 la distinction majuscule minuscule peut avoir lieu de façon baroque et qu'il faut être prudent. Les réponses sont appelées dans l'ordre %o1, %o2, etc. ce qui permet de les réutiliser dans de nouvelles expressions.

```
(%i1) Pt(q) := q^2;
(%o1)          PT(q) := q2
(%i2) pt(2);
(%o2)          4
(%i3) p(q) := q^2;
(%o3)          p(q) := q2
(%i4) P(2);
(%o4)          P(2)
(%i5) p(2);
(%o5)          4
(%i6) %o5^2;
(%o6)          16
```

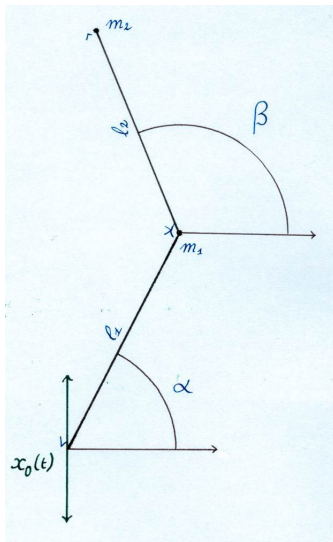
Un extrait d'une session Maxima
Fig. 1

Maxima est par ailleurs un peu lent. Une raison supplémentaire de bien réfléchir préalablement à la complexité des calculs envisagés. Le temps des calculs, souvent imposé par la taille même du résultat, ou des résultats intermédiaires, est la grande contrainte du calcul formel. Bien qu'il n'ait pas bénéficié des progrès algorithmiques les plus récents, ses possibilités sont amplement suffisantes pour des travaux de modélisation et de simulation numérique.

4. Production de code numérique grâce à MAXIMA.

Le logiciel CARINS développé avec la participation d'APPEDGE pour le compte du CNES et de l'ONERA est une illustration réussie de l'emploi de CARINS dans le cadre d'un projet logiciel ambitieux de modélisation et de simulation numérique. Nous n'insisterons pas sur cet exemple développé dans un autre article de ce numéro (voir aussi [4]). Nous allons en revanche illustrer la manière dont MAXIMA peut être utilisé comme calculette formelle par l'ingénieur, pour produire les équations de ses modèles physiques en vue de la simulation, en nous donnant pour règle de n'utiliser que des logiciels libres.

Nous considérons un exemple mécanique élémentaire, mais dont les équations sont néanmoins difficiles à calculer à la main : le double pendule de Kapitza. Un pendule en position verticale, mais dont le point d'attache est affecté d'un mouvement vibratoire vertical à une fréquence assez élevée devient stable. Ce phénomène, remarqué par Van der Pol en 1925, a été élégamment décrit et expérimentalement démontré par Pyotr Leonidovich Kapitza en 1950. On peut généraliser le phénomène avec un pendule double (voir fig. 2).



Le plus simple pour écrire les équations est d'utiliser le formalisme du Lagrangien. Mais, à la main, c'est fastidieux pour un calculateur du 21^e siècle. Maxima nous permet de résoudre le problème en quelques lignes (voir fig. 3). Il faut ensuite traduire le résultat pour le réutiliser en Scilab (voir l'article de Claude Gomez dans ce numéro). Ceci peut être fait grâce à Emacs, l'éditeur fétiche sous licence GPL. La tâche principale est de remplacer les majuscules par des minuscules, Scilab faisant lui la distinction. Une subtilité : « beta » est réservé pour l'implantation en Scilab de la fonction β . On l'a donc remplacé par « Beta ».

Fig. 2 : Le double pendule de Kapitsa

Code formel en MAXIMA :

```
DT(exp):=diff(exp,t)+diff(exp,alpha)*alpha1+diff(exp,alpha1)*alpha2+
diff(exp,beta)*beta1+diff(exp,beta1)*beta2;

m1:3; m2:1; l1:2; l2:1; g:9.81; x0:0.0;y0:0.1*cos(100*t);
x1:x0+l1*cos(alpha); x2:x1+l2*cos(beta); y1:y0+l1*sin(alpha); y2:y1+l2*sin(beta);

LA1:(m1*(DT(x1)^2+DT(y1)^2)+m2*(DT(x2)^2+DT(y2)^2))/2-g*(m1*y1+m2*y2);

display2d:false;
solve([DT(diff(LA1,alpha))=diff(LA1,alpha),DT(diff(LA1,beta1))=
diff(LA1,beta)], [alpha2,beta2]);
```

Résultat traduit grâce à Emacs et incorporé dans un fichier Scilab :

```
function derivees=DERIVEES(t,etat)
alpha=etat(1);
alpha1=etat(2);
Beta=etat(3);
Beta1=etat(4);
alpha2=((40000*cos(alpha)*sin(Beta)^2-10000*sin(alpha)*cos(Beta)*sin(Beta)
+30000*cos(alpha)*cos(Beta)^2*cos(100*t)+(100*cos(alpha)*sin(Beta)^3-
100*sin(alpha)*cos(Beta)*sin(Beta)^2+100*cos(alpha)*cos(Beta)^2*sin(Beta)-
100*sin(alpha)*cos(Beta)^3)*Beta1^2+(200*cos(alpha)*sin(alpha)*alpha1^2-3924*cos(alpha))*sin(Beta)^2+(-
200*sin(alpha)^2*alpha1^2+200*cos(alpha)^2*alpha1^2+981*sin(alpha))*cos(Beta)*sin(Beta)+(-
200*cos(alpha)*sin(alpha)*alpha1^2-2943*cos(alpha))*cos(Beta)^2)/((600*sin(alpha)^2+800*cos(alpha)^2)*sin(Beta)^2-
400*cos(alpha)*sin(alpha)*cos(Beta)*sin(Beta)+(800*sin(alpha)^2+600*cos(alpha)^2)*cos(Beta)^2);
Beta2=-((10000*cos(alpha)*sin(alpha)*sin(Beta)-10000*sin(alpha)^2*cos(Beta))*cos(100*t)
+(25*cos(alpha)*sin(alpha)*sin(Beta)^2+(25*cos(alpha)^2-25*sin(alpha)^2)*cos(Beta)*sin(Beta)-
25*cos(alpha)*sin(alpha)*cos(Beta)^2)*Beta1^2+(200*cos(alpha)*sin(alpha)^2*alpha1^2+200*cos(alpha)^3*alpha1^2-
981*cos(alpha)*sin(alpha))*sin(Beta)+(-200*sin(alpha)^3*alpha1^2-
200*cos(alpha)^2*sin(alpha)*alpha1^2+981*sin(alpha)^2*cos(Beta))/((75*sin(alpha)^2+100*cos(alpha)^2)*sin(Beta)^2-
50*cos(alpha)*sin(alpha)*cos(Beta)*sin(Beta)+(100*sin(alpha)^2+75*cos(alpha)^2)*cos(Beta)^2);
derivees=[alpha1;alpha2;Beta1;Beta2];
endfunction;

TFIN=10;
TIMES=[0:0.01:TFIN];
resu=ode([%pi/2+0.05;0.0;%pi/2+0.01;0.0],0,TIMES,DERIVEES);
plot2d(TIMES,[resu(1,:)' resu(3,:)],[1,2],"111", "alpha@Beta",[0,1.4,TFIN,1.7]);
```

Figure 3 : Code numérique complexe obtenu à partir de quelques commandes Maxima

On obtient ainsi les courbes suivantes (voir fig. 4), qui montrent que les pendules restent en position verticale. Ce petit exemple donne un aperçu des vastes possibilités offertes dans le cadre d'une utilisation manuelle de type « super calculette ». On devine les difficultés qu'il faut vaincre pour le rendre le processus entièrement automatique dans le cadre d'un projet comme CARINS (choix des lois physiques, production rapide des équations, traduction en code numérique).

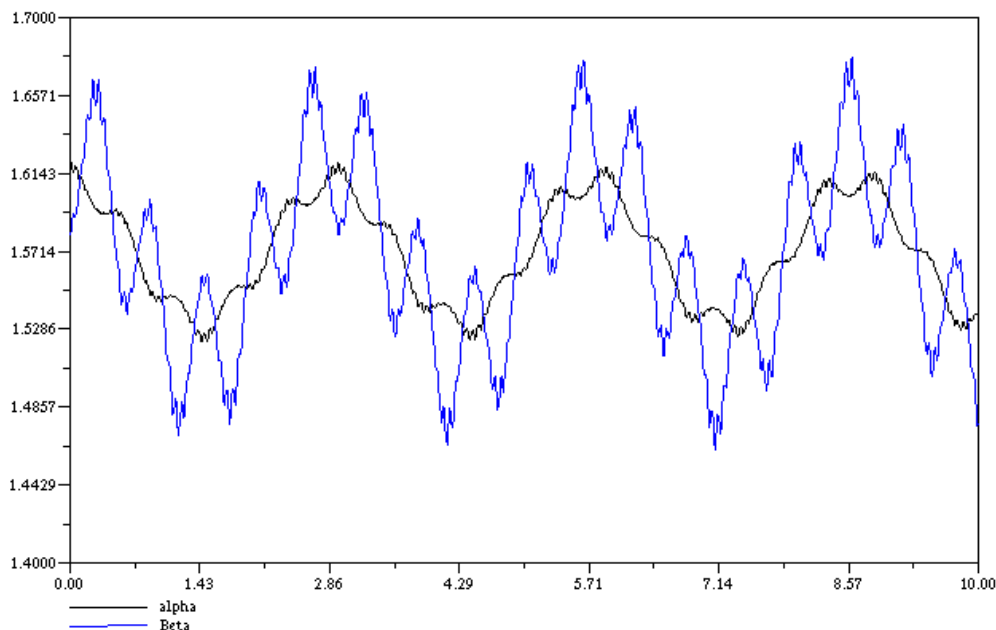


Fig. 4 : Courbes d'évolution temporelle des angles α et β

5. Conclusions

MAXIMA est un logiciel libre de calcul formel bénéficiant d'une communauté d'utilisateurs suffisante pour en assurer la pérennité, le développement ainsi que l'aide à l'utilisateur débutant ou confirmé via les forums de discussion. Son apprentissage peut être rapide et nécessite surtout, comme pour tout logiciel formel de comprendre un peu la logique de ce type de calcul. « Quelle sera la taille du résultat ? » est une question de base. Que pourrait-on faire en effet d'un polynôme de degré 100 en 10 variables qui compte 51540966982791 monômes et ne saurait tenir sur aucun disque dur ? Pour la modélisation, un réflexe de base est de passer au plus vite au numérique, plus rapide et plus concis, et donc de limiter le recours au calcul formel à ce qui le rend véritablement indispensable.

Les exemples considérés ici ne peuvent couvrir toute la variété des possibilités offertes par les programmes généralistes ou spécialisés. Il devient en fait de plus en plus difficile de faire une synthèse « du » calcul formel. L'ouvrage [2] permet néanmoins de se faire une idée des méthodes et outils de base, même s'il a vieilli et si réécrire un équivalent à jour serait une entreprise téméraire. Le Web est l'outil par excellence de documentation dans ce domaine mouvant. Si les principaux systèmes ont été développés en Amérique du nord, l'Europe a

largement contribué à l'écriture des packages qu'il contiennent et la France possède une école très active, bien qu'il n'existe plus de structure académique permettant de fédérer la discipline, répartie à entre l'INRIA et deux commissions du CNRS. L'association GAGE se propose, entre autre, d'être un lieu de contact dans ce domaine entre la recherche académique et l'industrie. (GAGE ; 11, rue Crozatier ; 75012 Paris ; Courriel : g.a.ge@free.fr)

Références

[1] « Logiciels libres et calcul formel », document collectif disponible à l'url <http://www.lapcs.univ-lyon1.fr/~nthiery/CalculFormelLibre/>

[2] James DAVENPORT, Yvon SIRET, Evelyne TOURNIER, *Le calcul formel*, Masson, 1987.

[3] Richard J. FATEMAN, *An overview of Macsyma*, 1982-1984, légèrement révisé en 2001, <http://www.cs.berkeley.edu/~fateman/papers/mac82b.pdf>

[4] Benjamin LEGRAND, John MASSE, Vincent LEUDIERE, Gérard ALBANO, *CARINS : un logiciel de modélisation et de simulation pour des procédés industriels complexes*, Journée LMCS (Logiciel de Modélisation et de Calcul Scientifique), 24 novembre 2004, http://aldebaran.devinci.fr/cs/lmcs2004/ENLIGNE/Legrand_resume.pdf