Séminaire de Théorie des Nombres de Bordeaux
Année 1986-1987 - Exposé n° 17 le 6 février 1987

D5 : UN SYSTEME DE CALCUL FORMEL AVEC DES NOMBRES ALGEBRIQUES

par

Dominique DUVAL

Il s'agit ici de décrire un système permettant de calculer (sur ordinateur) de manière exacte avec des nombres algébriques. Ce système, baptisé D5, a été mis au point et implanté avec Claire Dicrescenzo sur le système de calcul formel Reduce.

Il n'a pas été conçu *a priori* pour les arithméticiens, mais pour les utilisateurs du calcul formel. En effet ceux-ci, en traitant des problèmes très variés (intégration, équations différentielles, résolution de systèmes polynomiaux, géométrie,...) rencontrent, bien malgré eux, des nombres algébriques lors des calculs. Ces nombres algébriques apparaissent la plupart du temps sous la forme suivante :

- on calcule dans $\mathbb{Q}$,
- apparaît un polynôme P_1 à coefficients dans $\mathbb{Q}$,
- on poursuit les calculs dans $\mathbb{Q}(\alpha_1)$, où α_1 désigne une racine *quelconque* de P_1,
- apparaît un polynôme P_2 à coefficients dans $\mathbb{Q}(\alpha_1)$,
- on poursuit les calculs dans $\mathbb{Q}(\alpha_2, \alpha_2)$, où α_2 désigne une racine *quelconque* de P_2,
- etc.

A la fin de l'algorithme, les calculs se font dans $\mathbb{Q}(\alpha_1, \alpha_2, \ldots, \alpha_n)$ pour un certain entier n, appelé le *niveau* de l'extension. Nous verrons que cet entier, sans grande signification mathématique (d'après le théorème de l'élément primitif), est en pratique fort important.

Remarquons que

* Les calculs doivent être faits pour *chaque* racine α_1 de P_1, α_2 de P_2, etc.

* Les calculs comportent au minimum les *opérations de corps* (+,-,×,/ et =), et nous nous limitons ici à ces opérations, ce qui permet déjà de traiter de nombreuses applications.

* Le polynôme P_i n'est *pas* supposé *irréductible* sur $\mathbb{Q}(\alpha_1,\ldots,\alpha_{i-1})$.

La méthode généralement proposée pour effectuer ces calculs consiste à se ramener au cas où chaque P_i est irréductible sur $\mathbb{Q}(\alpha_1,\ldots,\alpha_{i-1})$, en factorisant P_1 sur $\mathbb{Q}$, puis P_2 sur $\mathbb{Q}[X]/(\prod_1)$ pour chaque facteur irréductible $\prod_1$ de P_1, etc.

Toutes les opérations se ramènent ensuite à des calculs "classiques" sur les polynômes à coefficients rationnels (essentiellement l'algorithme d'Euclide).

Mais cette méthode présente un gros inconvénient : le coût de la factorisation des polynômes. En effet, les algorithmes de factorisation sont compliqués (donc les programmes correspondants prennent beaucoup de place en mémoire), et leur exécution est très longue. De plus, cette complexité augmente beaucoup avec le niveau n, que l'on se ramène ou non au niveau 1 par calcul d'un élément primitif [Lo,A-B-D,B-H-P-S].

Le principe du système D5 est très simple, et *évite toute factorisation*. C'est D. Lazard qui nous a suggéré l'idée, et nous avons commencé à l'étudier avec J. Della Dora, ce qui explique son nom [DDDDD].

Expliquons ce principe au niveau 1, le cas d'un niveau quelconque s'y ramenant facilement par récursivité.

Soit donc à calculer dans $\mathbb{Q}(\alpha)$, où α désigne une racine quelconque d'un polynôme P de $\mathbb{Q}[X]$. Quitte à remplacer P par $P/pgcd(P,P')$, on peut supposer que P n'a que des facteurs simples, disons $P=Q_1Q_2\ldots Q_r$ avec les Q_i irréductibles sur $\mathbb{Q}$ et deux à deux non proportionnels.

Alors le théorème chinois dit que l'application canonique

$$\mathbb{Q}[X]/(P) \to \prod_{i=1}^{r} \mathbb{Q}[X]/(Q_i)$$

est un isomorphisme d'anneaux. Or chaque $\mathbb{Q}[X]/(Q_i)$ est un corps, et on veut calculer dans tous ces corps. Il suffit donc de calculer dans $\mathbb{Q}[X]/(P)$ tant que seules les *opérations d'anneau* (+,-,×) sont utilisées. Par contre ce n'est plus possible en général pour

les inversions et les tests d'égalité :

Par exemple, si $P=X^3-X$, est-ce que $\alpha=1$?

Commençons par les tests d'égalité. Il est toujours possible, par soustraction, de se ramener à un test de la forme

$$A(\alpha)=0 \text{ ? où } A\in\mathbb{Q}[X].$$

On calcule alors

$$D=\text{pgcd}(P,A) \text{ et } E=P/D.$$

La paire (D,E), qui forme une factorisation partielle de P sur Q, est appelée un *scindage* de P. Le théorème chinois prouve alors l'isomorphisme

$$\mathbb{Q}[X]/(P) \longrightarrow \mathbb{Q}[X]/(D)\times\mathbb{Q}[X]/(E).$$

Dans le premier facteur, la réponse au test "$A(\alpha)=0$?" est OUI, ce qui signifie que $A(\alpha)$ est nul pour toute racine α de D, et dans le deuxième facteur la réponse est NON, autrement dit $A(\alpha)$ n'est nul pour aucune racine α de E.

Les calculs sont alors poursuivis "en parallèle" dans $\mathbb{Q}[X]/(D)$ et dans $\mathbb{Q}[X]/(E)$, et le polynôme P est oublié. Bien sûr, des *raffinements* de ce scindage peuvent apparaître par la suite, si de nouveaux tests d'égalité entre nombres algébriques sont rencontrés.

Quant aux inversions, elles ne posent en fait aucun problème si on a pris soin de tester auparavant que le nombre algébrique à inverser n'est pas nul :

En effet, soit à calculer $1/A(\alpha)$ après avoir testé $A(\alpha)\neq 0$. On a obtenu un scindage de P en (D,E) et on sait que le test n'est vérifié que dans la deuxième *branche* $\mathbb{Q}[X]/(E)$. Mais alors A et E sont premiers entre eux, et l'algorithme d'Euclide étendu permet de calculer les coefficients de l'égalité de Bezout entre A et E, en particulier $B\in\mathbb{Q}[X]$ tel que $AB\equiv 1 (\text{mod. } E)$. Alors

$$1/A(\alpha)=B(\alpha) \text{ pour toute racine } \alpha \text{ de } E.$$

On voit donc que le système D5 est simple *et* élémentaire, et qu'il n'utilise que l'algorithme d'Euclide.

Cependant, ce système présente *a priori* un gros inconvénient : il complique l'écriture des programmes, puisque chaque test d'égalité entre nombres algébriques doit être remplacé par un scindage et suivi d'une boucle englobant toute la suite du programme et permettant de parcourir toutes les branches du scindage.

Quelques exemples suffisent pour se persuader que cela rend le programme illisible et la correction des erreurs très difficile. Par exemple, pour calculer le degré d'un polynôme $\Phi(T)=a_1T+a_0$, on a envie d'appliquer le "programme" suivant :

```
Programme I.
début
si a1≠0 alors d ← 1
   sinon si a0≠0 alors d ← 0
            sinon d ← + ∞ ;
retourner d ;
fin.
```

Mais si a_1 et a_0 sont des nombres algébriques que l'on veut traiter avec le système D5, le programme qu'il faut écrire est le suivant (dans lequel ld désigne un ensemble qui groupera les résultats obtenus dans les diverses branches, et *extension* est une caractérisation de la branche dans laquelle on est, c'est-à-dire de l'anneau dans lequel on est en train de calculer) :

```
Programme T.
début
ld ← ∅ ;
scinder par rapport à a1 ;
si a1≠0 alors ld ← U{(0,extension)}
   sinon scinder par rapport à a0 ;
         si a0≠0 alors ld ← ld U {(0,extension)}
            sinon ld ← ld U {(+∞,extension)} ;
retourner ld ;
fin.
```

Notons que les "vrais" programmes sont essentiellement les mêmes que ceux-ci, mais leur écriture est un peu plus technique.

Si maintenant le programme T est appliqué au polynôme

$$\Phi = (\alpha^4-\alpha^3-4\alpha^2+4\alpha)T+(\alpha^2-1)$$

où α désigne une racine quelconque de

$$P=X^4-2X^3-X^2+2X$$

le résultat obtenu est

$$\{(1,X+1),(0,X^2-2X),(+\infty,X-1)\},$$

ce qui signifie que le degré de Φ est

* 1 si α est racine de $X+1$,
* 0 si α est racine de X^2-2X,
* $+\infty$ si α est racine de $X-1$.

Remarquons que, dans cet exemple, la méthode "classique" aurait d'abord factorisé

$$P=X(X-1)(X+1)(X-2)$$

puis fait 4 calculs, obtenant

* 1 si α est racine de $X+1$,
* 0 si α est racine de $X-2$,
* 0 si α est racine de X,
* $+\infty$ si α est racine de $X-1$.

Ce problème de programmation a été résolu en fixant des règles de "transformation de programme" qui permettent d'écrire un programme *initial* sans se préoccuper des scindages -c'est le programme I ci-dessus-. Un programme annexe, écrit par J.L. Roch, analyse ce programme initial, et lui applique les règles de transformation. Il renvoie le programme *transformé* -le programme T ci-dessus-, qui est maintenant exécutable par le système D5.

Grâce à cette transformation automatique de programmes, l'utilisation de D5 est très simple, et l'utilisateur peut se consacrer à *son* problème, sans avoir à se préoccuper de la manière dont les nombres algébriques sont traités. Comme il a été dit dans l'introduction, c'est en général ce que cherche l'utilisateur, que les nombres algébriques en eux-mêmes n'intéressent pas.

En conclusion, D5 apporte pour la première fois une réponse simple et efficace à un problème fondamental du calcul formel, et qui en limitait sévèrement la portée.

Ce système a déjà été utilisé pour calculer des formes de Jordan de matrices [Oz], résoudre des équations différentielles

[Ba], calculer des développements de Puiseux de courbes planes [Du-2].
Il est clair que le corps $\mathbb{Q}$ pourrait être remplacé par n'importe quel corps, avec quelques modifications en caractéristique finie. Malheureusement un système comme Reduce se prête mal à ces généralisations. Une implantation prévue sur le système de calcul formel **Scratchpad II** devrait permettre de choisir n'importe quel corps de base "calculable".

Par ailleurs P. Ozello a remarqué que **D5** traite le nombre algébrique α racine de P comme un paramètre, soumis à la contrainte $P(\alpha)=0$; et que **D5** peut être généralisé en un système de discussion automatique de problèmes avec paramètres. Il travaille actuellement à l'implantation d'une telle généralisation.

REFERENCES

[A-B-D] **J.A. ABBOT, R.J. BRADFORD, J.H. DAVENPORT.-** *The Bath algebraic number package,* Symsac'86, p. 250-253, B.W. Char ed., Univ. of Waterloo, 1986.

[B-H-P-S] **R.J. BRADFORD, A.C. HEARN, J.A. PADGET, E. SCHRUEFER.-** *Enlarging the* **REDUCE** *domain of computation,* Symsac'86, p. 100-106, B.W. Char ed., Univ. of Waterloo, 1986.

[Lo] • **R. LOOS.-** *Computing in algebraic extensions,* Computer algebra, symbolic and algebraic computation, p. 173-187, Springer-Verlag, 1983.

Sur le système D5 :

[DDDDD] **J. DELLA DORA, C. DISCRESCENZO, D. DUVAL.-** *About a new method for computing in algebraic number fields,* Lecture Notes in Computer Science, Springer Verlag, 204 (1985), 289-290.

[D-D-1] **C. DISCRESCENZO, D. DUVAL.-** *Calculs algébriques avec des nombres algébriques : exemples,* Calsyf 4, éd. M. Mignotte, Univ. de Strasbourg, 1985.

[D-D-2] C. DISCRESCENZO, D. DUVAL.- *Algebraic computations on algebraic numbers,* Informatique et calcul, p. 54-61, Masson-Wiley, 1986.

[Du-1] D. DUVAL.- *Diverses questions relatives au calcul formel avec des nombres algébriques,* Thèse d'Etat, Univ. de Grenoble 1, parties 1 (et 4), 1987.

Sur diverses applications :

[Ba] A. BARKATOU.- *Résolution formelle des équations différentielles linéaires d'ordre 2,* Rapport de recherche TIM3, Univ. de Grenoble 1, 1987.

[Du-2] D. DUVAL.- *Diverses questions relatives au calcul formel avec des nombres algébriques,* Thèse d'Etat, Univ. de Grenoble 1, parties 2 (et 3), 1987.

[Oz] P. OZELLO.- *Calcul des formes de Frobenius et de Jordan d'une matrice,* Thèse, Univ. de Grenoble 1, 1987.

(Texte reçu le 8 juillet 1987)

Dominique DUVAL
Université de Grenoble I
Institut Fourier
Laboratoire de Mathématiques
associé au CNRS
B.P. 74
38402-St-Martin-d'hères