Introduction au Calcul formel — Algorithmes et complexité

Frédéric Chyzak, Novembre 2006

CHAPITRE 1

Présentation du cours

1. Le calcul formel

Le calcul formel — *computer algebra* en anglais — a pour objet d'étude la manipulation symbolique effective d'objets mathématiques. Par ses méthodes, il procède en formalisant et en donnant une nature algorithmique au calcul sur des représentations informatiques finies d'objets mathématiques. Calculer, c'est en fait construire une telle représentation, le plus souvent algébrique, en respectant une mécanique fixée à l'avance et uniforme sur toute une classe d'objets. Par delà les questions de faisabilité, une évolution récente de la discipline est de revisiter des problèmes anciens du point de vue de la complexité.

L'objectif principal de cette introduction au calcul formel sera d'en montrer les algorithmes de base, avec, à chaque fois, des exemples d'applications. Les algorithmes étudiés permettront par exemple de répondre à des questions telles :

– Quelle est l'enveloppe des cordes d'une ellipse vues sous un angle droit depuis un point donné ?

(Indication : faire pivoter une paire de droites orthogonales autour du point fixe et obtenir une paramétrisation du point variable.)

– Comment se poursuit la suite $(3, 0, 4, 2, 3, 0, \dots)$ d'entiers modulo 5 ?

(Indication : calculer un approximant de Padé-Hermite.)

– Est-il possible de trouver des entiers $e_1, \dots, e_5$ tels que $12e_1+43e_2+189e_3+19e_4+289e_5 = 220$?

(Indication : utiliser un algorithme de réduction de réseau.)

– L'intégrale de $x^{1/2}(lnx)^5/(1-x)^5$ de 0 à l'infini est difficilement calculable sous forme explicite ; une évaluation numérique donne la valeur approchée $V = -16.699473719229070496187243400 7\dots$ On a de bonnes raisons de penser que ce nombre s'exprime aisément comme combinaison linéaire des puissances de $\pi = 3.14159\dots$ avec des coefficients rationnels. Peut-on trouver s'il y a une telle relation entre V et π ?

(Indication : considérer des approximations rationnelles de V et des puissances de π et utiliser l'algorithme LLL.)

– Dans un modèle simplifié de l'interaction d'une mailloche frappant la membrane d'une timbale circulaire, la détermination des modes propres de la vibration amène à calculer l'intégrale entre -1 et $+1$ de $f_n(p, x) = \exp(-px)T_n(x)/(1-x^2)^{1/2}$. Que vaut cette intégrale ?

(Indication : écrire un système d'équations différentielles et de récurrences vérifiées par la fonction $f_n(p, x)$ et se débarrasser de x.)

– Quelles sont les conformations géométriques possibles en trois dimensions du cyclohexane C_6H_{12} ?

(Indication : on sait que les angles entre les liaisons carbone-carbone successives ont pour cosinus $-1/3$ et que les atomes de carbone sont libres de leur rotation autour des liaisons. On pourra effectuer plusieurs éliminations polynomiales.)

Par les techniques utilisées (souvent, de l'algèbre) et les objectifs visés (algorithmiser des questions mathématiques), le cours s'adresse à un public d'informaticiens mathématiquement réceptifs.

2. Contenu du cours

Le cours abordera les sujets et algorithmes suivants, traitant tantôt de structures de données de bas niveau, tantôt de représentations d'objets mathématiques de haut niveau. (L'ordre qui suit n'est pas significatif.)

(1) entiers, polynômes, séries, algorithme d'Euclide, résultants ;

Les opérations élémentaires, somme, produit et division d'entiers et de polynômes, sont au centre de la plupart des algorithmes de calcul formel. On présentera quelques solutions classiques pour la multiplication rapide, culminant dans les algorithmes de type FFT (Fast Fourier Transform). On verra ensuite un premier algorithme d'élimination, l'algorithme d'Euclide, qui s'applique aussi bien aux entiers qu'aux polynômes. Ses applications sont nombreuses, on verra notamment les approximants de Padé et les calculs de résultants.

(2) algèbre linéaire (produit de matrices, algorithme de Gauss et optimisations) ;

Les questions relatives aux opérations matricielles sont extrêmement variées. D'une part, on montrera comment la multiplication, l'inversion, le calcul du polynôme caractéristique, sont de complexité équivalente, lorsqu'on traite des matrices à coefficients dans un corps. On introduira la notion d'exposant de l'algèbre linéaire, qui vient mesurer cette complexité. On évoquera ensuite l'algorithmique de l'algèbre linéaire creuse, qui permet de traiter des matrices de taille très conséquente, à plusieurs centaines de milliers d'entrées. Enfin, on traitera des questions relatives aux matrices ayant d'autres types de coefficients, notamment des entrées entières, des séries formelles, voire des nombres flottants.

(3) bases de Gröbner et algorithme de Buchberger, résolution de systèmes polynomiaux ;

Les bases de Gröbner fournissent une extension de l'algorithme d'Euclide pour le cas de polynômes en plusieurs variables. L'algorithme de Buchberger pour le calcul de bases de Gröbner fournit une procédure d'élimination pour le cas de plusieurs variables. Cette approche algorithmique importante a de nombreuses applications en géométrie (notamment en géométrie algébrique), par exemple à la démonstration semi-automatique de théorèmes de géométrie et à la recherche d'équations implicites de courbes algébriques paramétrées. Une autre application très importante est l'utilisation des bases de Gröbner pour la résolution de systèmes de polynômes en plusieurs variables. La contrepartie de la large applicabilité de cette théorie est sa complexité élevée, doublement exponentielle, au moins dans le cas le pire.

(4) algorithme de réduction de réseaux (LLL) ;

Étant donnés des vecteurs V_1, …, V_m dans l'espace $\mathbb{Z}^n$, on appelle réseau engendré par V_1, …, V_m l'ensemble de leurs combinaisons linéaires à coefficients entiers. Le problème de trouver le plus court vecteur dans un réseau est NP-dur, il est donc peu vraisemblable que l'on dispose prochainement d'un algorithme efficace pour le traiter. Si l'on relâche la contrainte d'optimalité, il devient possible de trouver un vecteur « assez » court en temps polynomial, en utilisant l'algorithme LLL (pour « Lenstra, Lenstra, Lovasz »). Ses applications sont extrêmement nombreuses et vont de la théorie des nombres aux problèmes de sac à dos, en passant par la factorisation des polynômes.

(5) intégration symbolique (des fractions rationnelles à l'algorithme de Risch) ;

À la main, le calcul d'une intégrale définie passe souvent par la recherche d'une primitive. Il en est de même en calcul formel, on l'on recherche de plus une forme explicite. Le cas de base, l'intégration des fractions rationnelles, sera traité en détail, puis l'algorithme de Risch sera abordé dans ses grandes lignes.

(6) équations différentielles linéaires (solutions polynomiales, rationnelles, exponentielles, calculs sur leurs solutions).

Le calcul formel s'intéresse aussi à la résolution en formules explicites d'équations différentielles ou à l'inverse, à la manipulation de solutions données implicitement comme solutions d'équations différentielles. On s'intéressera dans ce cours tout particulièrement au cas des équations différentielles linéaires. On verra tout d'abord comment en rechercher les solutions polynomiales, rationnelles

et exponentielles, puis des méthodes de manipulation permettant le calcul d'intégrales définies paramétrées.

Remerciements. Mes remerciements vont à Philippe Dumas qui a relu certains des chapitres de ce cours et contribué à améliorer l'exposition. Je remercie aussi Éric Schost avec lequel nous avons monté ce cours de 2003 à 2005, avant que je ne le reprenne seul.

CHAPITRE 2

Algorithmes naïfs sur les polynômes : Multiplication, division, algorithme d'Euclide et applications

Le sujet du produit d'entiers et de polynômes, de leurs division et p. g. c. d. peut sembler banal. Il l'est tant qu'on ne s'intéresse pas aux questions de temps de calcul. Dans ce chapitre, nous analysons les algorithmes naïfs qui serviront de base fiable pour les analyses des algorithmes optimisés au chapitre 3. Nous donnons aussi des applications non triviales de la division euclidienne et de l'algorithme d'Euclide.

1. Rappels minimaux et succincts d'algèbre : groupes, anneaux, corps

Cette section peut être sautée dès la première lecture par le lecteur qui se souvient de ses bases élémentaires d'algèbre.

Un *groupe* est un ensemble muni d'une loi interne associative, munie d'un élément neutre et pour laquelle tout élement admet un inverse (à gauche et à droite, le même). Si la loi est commutative, on dit que le groupe est *commutatif* ou encore *abélien*. La loi est alors le plus souvent notée additivement. Si c'est le cas, le neutre est noté 0. Des exemples de groupes commutatifs sont l'ensemble $\mathbb{Z}$ des entiers relatifs vu avec son addition et l'ensemble $\mathbb{Q}^\times$ des nombres fractionnaires non nuls vu avec son produit et son neutre 1. Un exemple non commutatif est donné par tout groupe de permutations d'un ensemble d'au moins quatre éléments.

Un *anneau* est un groupe commutatif noté additivement muni d'une seconde loi associative et distributive sur l'addition. Cette seconde loi est notée multiplicativement. Lorsqu'elle est commutative, l'anneau est dit *anneau commutatif*. Parmi les exemples que nous rencontrerons, nous avons les anneaux de nombres $\mathbb{Z}$, $\mathbb{Q}$, $\mathbb{R}$, $\mathbb{C}$, les anneaux de polynômes $A[X]$, de fractions rationnelles $A(X)$, de séries formelles $A[[X]]$, de matrices $M_n(A)$. Parmi ceux-ci, seul le cas des matrices fournit un anneau non commutatif. Notons que nos anneaux auront toujours un neutre, noté 1, pour la multiplication, même si le calcul formel s'intéresse parfois à des anneaux sans neutre.

Un *corps* est un anneau commutatif possédant un élément neutre 1, dans lequel tout élément non nul est inversible pour la loi multiplicative. La *caractéristique* d'un corps est le plus petit entier naturel non nul n tel que $n \times 1$ soit nul dans le corps. Par convention, on dit qu'un corps donné est *de caractéristique nulle* lorsqu'un tel entier n'existe pas. Les exemples les plus courants sont $\mathbb{Q}$, $\mathbb{R}$, $\mathbb{C}$ et $\mathbb{Z}/p\mathbb{Z}$, le corps des entiers vu modulo un nombre premier p. Les premiers exemples sont de caractéristique nulle ; le dernier de caractéristique p.

D'autres rappels seront faits au gré des besoins.

2. Algorithmes algébriques, complexité algébrique

Dans ce cours, on souhaite répondre de façon satisfaisante et scientifiquement argumentée à des questions comme : quels sont les coûts de la multiplication et de la division d'un polynôme de degré n par un autre de degré m ? quel est le coût de la résolution d'un système linéaire donné par une matrice de taille $n \times m$? Par « quel est le coût de », on entend ici « en combien de temps se réalise ». Les réponses qui seront données nécessitent un modèle des algorithmes de calcul algébrique qui passe par la théorie de la complexité algorithmique.

Pour définir un algorithme algébrique, on considère un anneau (commutatif) A ou une ou plusieurs structures algébriques au dessus de A (polynômes, fractions rationnelles, séries, matrices ou autres, à coefficients dans A). On dira qu'un algorithme est un *algorithme algébrique* s'il prend

en entrée, manipule et renvoie un ou plusieurs éléments de ces structures et si ces manipulations se résolvent ultimement en des opérations dans A.

Pour mesurer l'efficacité d'un algorithme algébrique, on souhaite définir une notion de complexité arithmétique d'un algorithme algébrique. On attribue pour cela aux éléments de ces structures algébriques une notion de taille, selon plusieurs modèles possibles. Pour une entrée polynomiale par exemple, la taille est généralement le degré du polynôme (modèle dense), mais il peut être le nombre de termes non nuls de son écriture développée et réduite, ou une mesure de la taille d'un programme qui en calcule une évaluation en un élément donné de A.

Pour un modèle de taille fixé, la *complexité arithmétique* d'un algorithme algébrique est le nombre maximal $C(n)$ d'opérations dans A effectuées sur une entrée de taille n, où l'on prendra en compte :

- les additions, soustractions et multiplications entre éléments de A,
- si l'anneau A est un corps, les divisions,
- les tests d'égalité entre deux éléments de A, en particulier les tests à zéro,
- si l'anneau A est ordonné, les tests d'ordre entre deux éléments de A, en particulier les tests de positivité et de négativité.

Comme on accède rarement exactement aux complexités arithmétiques, on se contentera le plus souvent d'indiquer une classe asymptotique dans laquelle tombe une complexité arithmétique. En pratique, pour une fonction f donnant un régime polynomial, $f(n) = n^\alpha$, exponentiel, $f(n) = \exp(\alpha n)$, ou doublement exponentiel, $f(n) = \exp\exp(\alpha n)$, on écrira $C \in O(f)$ pour dire que la complexité est au plus polynomiale, exponentielle, doublement exponentielle.

3. Motivations de l'algorithme d'Euclide

3.1. Suites linéairement récurrentes. Le type de problèmes suivant est posé à la petite école ou par les psychologues pour mesurer l'intelligence des enfants. On en donne ici des instances typiques et de plus difficiles. Il consiste à prolonger une suite de nombres donnés. Par exemple :

- 1, 1, 1, 1, 1, 1, 1, 1, ...
- 5, 9, 13, 17, 21, 25, 29, 33, ...
- 0, 1, 1, 2, 3, 5, 8, 13, ...
- 12, 134, 222, 21, −3898, −40039, −347154, −2929918, −24657854, ...

Les réponses aux trois premières sont certainement immédiates pour le lecteur. S'il s'interroge pour déterminer comment il a trouvé la solution, le lecteur se rendra compte que, plus ou moins consciemment, il a réalisé que la première suite est constante — $u_n = u_{n-1}$ —, que la différence entre deux termes de la deuxième est constante — $u_n - u_{n-1} = 4$ — et qu'un terme de la troisième s'obtient en sommant les deux précédents — $u_n = u_{n-1} + u_{n-2}$. Dans chaque cas, la suite vérifie une récurrence linéaire à coefficients constants et, si l'on normalise la deuxième sous la forme $u_n - 2u_{n-1} + u_{n-2} = 0$, toutes vérifient des récurrences *homogènes* (c'est-à-dire « sans second membre »).

La réponse pour la quatrième suite est −207605083. Pour cela, il « suffit » d'observer que la suite vérifie la récurrence $u_n = 12u_{n-1} - 33u_{n-2} + 22u_{n-3} + 19u_{n-4}$. Un humain a peu de chances d'obtenir ce résultat de tête, mais le principe reste le même.

Toutes les suites de ces problèmes sont dites *suites linéairement récurrentes*. Formellement, une suite u de $V^{\mathbb{N}}$ pour un espace vectoriel V sur un corps k est linéairement récurrente lorsqu'il existe des coefficients $a_0, \ldots, a_r$ de k, avec a_r non nul, tels que

$$a_r u_{n+r} + \cdots + a_0 u_n = 0 \tag{1}$$

pour tout entier positif n. Plusieurs questions algorithmiques se posent immédiatement à leur sujet :

- Comment calculer efficacement pour N grand un terme u_N de la suite à partir de la récurrence ?
- Comment calculer efficacement pour N grand les N premiers termes de la suite à partir de la récurrence ?
- Comment retrouver la récurrence à partir de suffisamment de termes ?
- Combien de termes sont nécessaires pour établir la récurrence ?

L'algorithme d'Euclide de ce chapitre fournit une réponse à la troisième question. L'opérateur de Newton du chapitre **??** donnera une solution pour la première. Les autres questions relèvent bien du calcul formel, mais ne sont pas traitées dans ce cours.

3.2. Entiers modulo m, polynômes modulo f. Les deux cadres algébriques suivants, entiers modulo m et polynômes modulo f, ont des propriétés analogues. L'intuition du cas naturel des entiers doit soutenir l'explication du cas polynomial.

Pour un entier strictement positif m, l'anneau $\mathbb{Z}/m\mathbb{Z}$ est l'anneau des entiers modulo m : deux entiers de $\mathbb{Z}$ y sont identifiés dès lors qu'ils diffèrent d'un multiple (pour un coefficient entier) de m. Chaque élément admet un unique représentant entre 0 et $m-1$, appelé *représentant principal*.

Considérons un anneau de polynômes $k[X]$ sur un corps k. Pour un polynôme non nul f de $k[X]$, de degré d, l'anneau $k[X]/(f)$ est l'anneau des polynômes modulo l'idéal (f) : deux polynômes de $k[X]$ y sont identifiés dès lors qu'ils diffèrent d'un multiple (pour un coefficient polynomial) de f. Chaque élément admet un unique représentant de degré strictement inférieur à d, appelé *représentant principal*.

Dans ces deux cas d'anneaux quotients d'un anneau euclidien, les opérations d'anneau — addition, soustraction, produit — peuvent s'effectuer en réalisant d'abord l'opération sur les représentants principaux, avant de réduire le résultat modulo m ou f, respectivement. Pour le produit, la division euclidienne permet la réduction. L'inversion d'un élément dont le représentant principal ne divise pas m ou f, respectivement, et plus généralement l'opération partielle de division reposent sur l'algorithme d'Euclide.

Dans ce qui suit, nous ne traitons pas le cas des entiers aussi systématiquement que le cas des polynômes. En effet, bien qu'historiquement, le premier ait le plus souvent été traité avant le dernier, les phénomènes de retenues dans le produit d'entiers complique considérablement les analyses. On pourra considérer, en première analyse, que les résultats énoncés sur les polynômes se transcrivent aux entiers en remplaçant le degré par la valeur absolue. Nous énoncerons pourtant les quelques résultats sur les entiers qui s'obtiennent à peu de frais.

4. Complexité des algorithmes naïfs de multiplication et de division euclidienne

L'algorithme de multiplication naïve de polynômes $a = a_0 + \cdots + a_n X^n$ et $b = b_0 + \cdots + b_m X^m$ à coefficients dans un anneau A calcule simplement le coefficient c_i de leur produit $ab = c_0 + \cdots + c_{n+m} X^{n+m}$ par la formule $c_i = a_i b_0 + \cdots + a_0 b_i$.

Algorithme (Multiplication naïve).

Entrée : un anneau intègre A et deux polynômes a et b de $A[X]$ de degrés respectifs n et m avec $n \geq m$

Sortie : le produit de a par b

Complexité arithmétique : $O(n^2)$

(1) Pour i de 0 à $n+m$, faire $c_i = 0$

(2) Pour i de 0 à n et j de 0 à m, ajouter $a_i b_j$ à c_{i+j}

(3) Renvoyer $c_0 + \cdots + c_{n+m} X^{n+m}$

Le lemme suivant fournit une estimation plus précise que celle annoncée ci-dessus. Elle resservira dans des analyses ultérieures.

Lemme 1. Le produit complet requiert $2(n+1)(m+1)$ additions et multiplications dans A.

Démonstration. Seule l'étape (2) réalise des calculs dans A. Il s'agit d'une boucle sur i itérée sur $n+1$ valeurs, exécutant une boucle sur j itérée sur $m+1$ valeurs. □

Présentons l'algorithme naïf pour la division euclidienne (en place) en conservant les mêmes notations.

ALGORITHME (Division euclidienne).

ENTRÉE : un anneau intègre A et deux polynômes non nuls a et b de $A[X]$ de degrés respectifs n et m avec $n \geq m$

SORTIE : le reste de la division de a par b

COMPLEXITÉ ARITHMÉTIQUE : $O(n^2)$

(1) Pour k décroissant à partir de n tant que $k \geq m$:
   (a) déterminer u et v dans A vérifiant $ua_k = vb_m$
   (b) remplacer chaque a_i par ua_i pour $0 \leq i < k$
   (c) retirer vb_i de a_{i+k-m} pour $0 \leq i < m$
(2) Renvoyer $a_0 + \cdots + a_{m-1}X^{m-1}$

Pour estimer le coût de l'algorithme ci-dessus, nous faisons l'hypothèse que l'étape (1)(a) peut se faire en un nombre κ d'opérations dans A qui ne dépend que de A, et non des valeurs prises par a_k et b_m. Cette hypothèse est vérifiée si A est un corps ou en faisant naïvement $u = b_m$ et $v = -a_k$, mais elle ne l'est pas, par exemple, dans le cas $A = \mathbb{Z}$ lorsque l'on cherche des cofacteurs u et v minimaux. Le lemme suivant fournit une estimation plus fine de la complexité que celle annoncée ci-dessus. Elle resservira dans l'analyse en complexité de l'algorithme d'Euclide en section 6.1.

LEMME 2. La division euclidienne d'un polynôme de degré n par un polynôme de degré m pour $n \geq m$ se fait en moins de $(n-m+1)(3n+\kappa)$ opérations arithmétiques pour une constante κ qui ne dépend que de l'anneau de coefficients.

DÉMONSTRATION. En tenant compte de l'hypothèse sur l'étape (1)(a), une exécution de l'étape (1) nécessite moins de $k + 2m + \kappa$ opérations arithmétiques et la boucle complète, moins de $n(n+1)/2 - m(m-1)/2 + (n-m+1)(2m+\kappa)$, soit encore moins de $(n-m+1)(3n+\kappa)$. □

5. Théorie du p. g. c. d. d'un anneau principal

En toute généralité, un *plus grand commun diviseur*, ou p. g. c. d., de deux éléments a et b d'un anneau A est un élément g qui divise a et b et tel que tout diviseur commun h de a et de b divise g.

Ce chapitre traite essentiellement des anneaux $\mathbb{Z}$ et $k[X]$ pour un corps k, dont la propriété principale est d'être des *anneaux principaux* : tout idéal peut se voir comme engendré par un unique générateur. Dans ce cadre, un p. g. c. d. de deux éléments a et b existe toujours et a la propriété caractéristique d'être un générateur de l'idéal engendré par a et b. Ceci se traduit par l'égalité

$$\{ ra + sb : r \in A, s \in A \} = \{ rg : r \in A \}.$$

Deux p. g. c. d. g et h devant donc se diviser l'un l'autre, ils s'obtiennent tous les uns des autres par multiplication par une unité (un élément inversible) de A : $g = uh$ pour un inversible u de A.

On décide donc d'un mode de normalisation des éléments de A pour pouvoir parler *du* p. g. c. d. unique. Dans le cas des entiers, on choisit le p. g. c. d. positif ; dans le cas des polynômes, le p. g. c. d. unitaire (de coefficient de plus haut degré 1).

Ainsi, le p. g. c. d. dans $\mathbb{Q}[X]$ de $a = 24X^4 + 17X^3 + 8X^2 + X - 14$ et de $b = 16X^4 + 46X^3 - 51X^2 + 5X - 70$ est $g = X^2 + 3X/8 + 7/8$. Les p. g. c. d. de ces mêmes polynômes sont de la forme αg pour α de $\mathbb{Q}$. Si ces deux polynômes avaient été vus dans $\mathbb{Z}/71\,\mathbb{Z}$, leur p. g. c. d. aurait été $X^2 + 27X + 63$, donnant lieu à 69 autres p. g. c. d. non unitaires par multiplication par les éléments de $\mathbb{Z}/71\,\mathbb{Z}$ autres que 0 et 1.

EXERCICE. Vérifier ces calculs en Magma.

Du point de vue de la notation, on écrit aussi $(a_1, \ldots, a_k)$ ou $Aa_1 + \cdots + Aa_k$ pour désigner l'idéal $\{ r_1a_1 + \cdots + r_ka_k : r_1 \in A, \ldots, r_k \in A \}$. Ainsi, dire que g est le p. g. c. d. de a et de b s'exprime aussi par les relations $(a, b) = (g)$ ou $Aa + Ab = Ag$.

De la relation $(a,b)=(g)$ découle l'existence de *cofacteurs* r et s tels que g s'écrive $ra+sb$. Dans le cas polynomial, quitte à remplacer r par le reste de sa division euclidienne par le polynôme b/g et à ajuster s en conséquence, on peut supposer sans perte de généralité la relation $\deg r < \deg b - \deg g$. Ceci impose la relation $\deg s < \deg a - \deg g$. Une paire (r,s) qui vérifie ces contraintes de degré est nécessairement unique. Dans le cas des entiers, le même raisonnement aboutit à l'existence et à l'unicité de cofacteurs vérifiant les contraintes $|r| < |b/g|$ et $|s| < |a/g|$.

La relation $g = ra+sb$ s'appelle une *relation de Bézout*. Dans le cas où le p. g. c. d. vaut 1, une telle relation permet d'inverser b modulo a (et a modulo b). En effet, on a alors les congruences $sb \equiv bs \equiv 1 \mod (a)$. Un exemple non trivial est donné par la recherche d'une équation différentielle linéaire vérifiée par une fonction algébrique f, que nous illustrons dans le cas où f donne le graphe de la strophoïde particulière d'équation cartésienne

$$x(x^2+y^2)+4x^2-4y^2+6xy=0$$

par $(x,y)=\big(t,f(t)\big)$. Pour tout t dans un domaine de définition que nous n'explicitons pas, nous avons

$$(t-4)f(t)^2+6tf(t)+(t+4)t^2=0.$$

Par dérivation, nous obtenons

$$2\big((t-4)f(t)+3t\big)f'(t)+f(t)^2+6f(t)+(3t+8)t=0,$$

c'est-à-dire qu'hors du lieu d'annulation de $(t-4)f(t)+3t$, nous obtenons $f'(t)$ par une fraction rationnelle en t et $f(t)$. Pour récrire cette dérivée sans plus faire apparaître f au dénominateur, introduisons l'anneau $A=\mathbb{Q}(t)[X]$ et les polynômes $a=(t-4)X^2+6tX+(t+4)t^2$, $b=2\big((t-4)X+3t\big)$ et $c=X^2+6X+(3t+8)t$, de sorte à obtenir les relations :

$$a\big(t,f(t)\big)=0, \qquad b\big(t,f(t)\big)f'(t)=-c\big(t,f(t)\big).$$

Le p. g. c. d. de a et b est 1 et nous avons la relation de Bézout

$$1=ra+sb \qquad \text{pour} \qquad r=\frac{t-4}{(t+5)(t-5)t^2} \quad \text{et} \quad s=-\frac{(t-4)X+3t}{2(t+5)(t-5)t^2}.$$

En conséquence, la dérivée $f'(t)$ est égale à $-s\big(t,f(t)\big)c\big(t,f(t)\big)$. Une expression équivalente s'obtient en tenant compte de l'annulation de a sur $f(t)$ et en considérant le reste de la division euclidienne de $-sb$ par a. On aboutit finalement à la relation :

$$f'(t)=\frac{(t^3-8t^2+100)f(t)+3t^3}{(t-5)(t-4)t(t+5)}.$$

Exercice. Vérifier ces calculs en Magma.

La notion de *plus petit commun multiple*, ou p. p. c. m., a un traitement formel analogue de celui du p. g. c. d. : un p. p. c. m. de deux éléments a et b d'un anneau A est un élément p que divisent a et b et tel que tout multiple commun h de a et b est nécessairement divisé par p. On montre que s'il existe un p. p. c. m. p de a et b, alors p divise ab dans A et ab/p est un p. g. c. d. de a et b. L'existence de p. p. c. m. est assurée dans le cas des anneaux principaux qui nous préoccupent. Ici encore, on choisit une certaine normalisation pour distinguer un p. p. c. m. particulier que l'on désignera comme *le* p. p. c. m. Il est commode de distinguer comme le p. p. c. m. d'éléments a et b de l'anneau celui qui s'exprime ab/g où g est le p. g. c. d. (distingué).

Pour prouver toutes les affirmations de cette section, il est utile d'observer qu'un p. g. c. d. est un générateur d'un idéal principal minimal parmi ceux contenant $(a)+(b)$ alors qu'un p. p. c. m. est un générateur d'un idéal principal maximal parmi ceux contenant $(a)\cup(b)$.

Exercice. Vérifier ces interprétations en termes d'idéaux et préciser les preuves de cette section.

6. Algorithme d'Euclide

6.1. Algorithme de base. L'algorithme d'Euclide qui suit s'appuie sur la propriété immédiate selon laquelle le p. g. c. d. de deux polynômes f et g est le même que celui de $f+qg$ et g pour tout polynôme q. Il ne reste plus qu'à organiser les calculs pour faire décroître quelque chose, et c'est ici le degré maximal de la paire (f,g) que l'algorithme fait décroître jusqu'à son minimum. La version décrite ici permet des choix de normalisation au cours du calcul. Les premiers exemples ne montreront que des normalisation triviales, avec $s_i = 1$, et nous reviendrons sur la question d'autres normalisations en section 6.3.

Algorithme (d'Euclide).

Entrée : un anneau intègre A et deux polynômes non nuls f et g de $A[X]$ de degrés respectifs n et m avec $n \geq m$

Sortie : un p. g. c. d. pas forcément unitaire de f et g

Complexité arithmétique : $O(n^2)$

(1) Initialiser r_0 à f et r_1 à g

(2) Pour i entier à partir de $i = 0$ et tant que r_{i+1} est non nul, déterminer par pseudo-division euclidienne un coefficient non nul $s_i \in A$, un polynôme q_i et un polynôme r_{i+2} de degré plus petit que celui de r_{i+1} tels que

$$s_i r_i = q_i r_{i+1} + r_{i+2}$$

(3) Renvoyer r_i

Preuve de l'algorithme. Notons δ_i le degré du polynôme r_i et e la valeur finale de i. On a donc les relations $\delta_i > \delta_{i+1}$, sauf peut-être en $i = 0$ où l'égalité est aussi possible, $r_e \neq 0$, $r_{e+1} = 0$ et $e \leq n$. Par le lemme 2, l'itération i de la boucle (2) utilise au plus $(\delta_i - \delta_{i+1}+1)(3\delta_i+\kappa)$ opérations arithmétiques. Au total, l'algorithme en utilise donc moins de $(\delta_0 - \delta_e + e)(3n+1)$, et donc moins de $2(3n+\kappa)n$. □

Prenons l'exemple aléatoire suivant, où l'on a uniquement contraint les degrés des polynômes initiaux r_0 et r_1 ainsi que le nombre de chiffres de leurs coefficients, et dans lequel on s'impose la normalisation $s_i = 1$ tout au long de l'algorithme :

$$\begin{aligned}
r_0 &= 72X^3 + 37X^2 - 23X + 87,\\
r_1 &= 43X^2 + 29X + 98,\\
r_2 &= -\frac{331522}{1849}X + \frac{209569}{1849},\\
r_3 &= \frac{14674219389677}{109906836484},\\
r_4 &= 0.
\end{aligned}$$

Une première remarque est que même partant de coefficients entiers, des dénominateurs apparaissent. La nécessité de deux étapes dans la division euclidienne de r_0 par r_1 pour obtenir le reste r_2 explique le dénominateur 1849, qui n'est autre que 43^2. De même, 109906836484 n'est autre que 331522^2.

Prenons un autre exemple, avec des polynômes initiaux à coefficients rationnels, pour préciser les tailles de nombres qui apparaissent, ici encore avec $s_i = 1$:

$$
\begin{aligned}
r_0 &= \frac{72}{23}X^4 + \frac{37}{71}X^3 - \frac{23}{35}X^2 + \frac{87}{66}X + \frac{56}{17},\\
r_1 &= \frac{43}{13}X^3 + \frac{29}{48}X^2 + \frac{98}{11}X + \frac{25}{51},\\
r_2 &= -\frac{1013256055661}{111597652320}X^2 + \frac{1118774451}{1129261958}X + \frac{59813911}{18116502},\\
r_3 &= \frac{43351434833721674347787197604 1}{42429928129508304558381804167}X + \frac{190851433005134787689817680}{22689801138774494416246954 1},\\
r_4 &= \frac{22018632994671032429083032424380045408004634250446121422178337}{69711674759513147990326431379826432008316794788185423104478 34},\\
r_5 &= 0.
\end{aligned}
$$

Après le calcul de r_2, une étape de mise en route, nous pouvons très nettement voir que le nombre de chiffres décimaux des numérateurs et des dénominateurs des calculs fait un peu plus que doubler à chaque étape.

Exercice. Reprendre ces calculs où d'autres exemples aléatoires en Magma pour se convaincre que ces phénomènes sont génériques.

Ce phénomène est général et relativement indépendant de l'anneau de coefficients, dès lors que celui-ci n'est pas fini. Pire, pour des calculs impliquant un paramètre, c'est-à-dire pour $A = \mathbb{Q}(T)$, par exemple, non seulement le degré en T des coefficients va doubler à chaque étape, mais le nombre de chiffres des coefficients dans $\mathbb{Q}$ va faire de même. À supposer que les algorithmiques entre entiers, entre polynômes en T et entre polynômes en X soient toutes naïves, on peut s'attendre à ce que chaque opération arithmétique dans $\mathbb{Q}(T)$ soit environ 16 fois plus coûteuse à une étape de l'algorithme d'Euclide dans $\mathbb{Q}(T)[X]$ qu'à la précédente. Globalement, le temps d'exécution devrait être proportionnel à $\sum_{i=0}^{n}(n-i)\,16^i$, c'est-à-dire à 16^n.

Ce cours a entre autres objectifs de donner au chapitre 3 des algorithmes essentiellements linéaires pour la multiplication et le p. g. c. d. d'entiers et de polynômes.

6.2. Algorithme d'Euclide étendu. Une version étendue de l'algorithme d'Euclide tient à jour tout au long du calcul des cofacteurs a_i et b_i qui expriment le reste r_i sous la forme $r_i = a_i r_0 + b_i r_1$. Ainsi, cet algorithme délivre dans le même temps qu'un p. g. c. d. une relation de Bézout $\operatorname{pgcd}(a_0, a_1) = a_e f + b_e g$ et un p. p. c. m. $a_{e+1} f = -b_{e+1} g$ de son entrée.

L'algorithme devient le suivant :

ALGORITHME (Algorithme d'Euclide étendu).

ENTRÉE : un anneau intègre A et deux polynômes non nuls f et g de $A[X]$ de degrés respectifs n et m avec $n \geq m$

SORTIE : les coefficients (q_i, r_i, s_i) de toutes les pseudo-division euclidienne effectuées au cours de l'algorithme d'Euclide classique ainsi que les cofacteurs (a_i, b_i) donnant chacun des restes intermédiaires

COMPLEXITÉ ARITHMÉTIQUE : $O(n^2)$

(1) Initialiser r_0 à f, r_1 à g, (a_0, b_0) à $(1, 0)$ et (a_1, b_1) à $(0, 1)$.

(2) Pour i entier à partir de $i = 0$ et tant que r_{i+1} est non nul :

(a) déterminer par pseudo-division euclidienne un coefficient non nul $s_i \in A$, un polynôme q_i et un polynôme r_{i+2} de degré plus petit que celui de r_{i+1} tels que

$$s_i r_i = q_i r_{i+1} + r_{i+2}$$

(b) faire

$$(a_{i+2}, b_{i+2}) = s_i(a_i, b_i) - q_i(a_{i+1}, b_{i+1})$$

(3) Renvoyer $(q_i, r_i, s_i, a_i, b_i)$ pour i entre 0 et e, la dernière valeur prise par i à l'étape (2)

PREUVE DE L'ALGORITHME. La preuve de correction de cet algorithme découle de l'invariant $r_i = a_i r_0 + b_i r_1$. Celui-ci est imposé pour $i = 0$ et $i = 1$ par l'initialisation (1), puis maintenu au cours de l'algorithme par (2)(b). □

On ne détaillera pas ici d'argument pour justifier que la complexité de cet algorithme est du même ordre que celle de l'algorithme de base.

EXERCICE. Justifier cette complexité.

Le résultat suivant précise l'évolution des degrés des restes successifs et sera utile pour arrêter l'algorithme à mi-parcours, dans les applications.

LEMME 3. Avec les notations de l'algorithme d'Euclide étendu, on a les relations

$$\begin{aligned} \deg a_{i+1} &= \deg g - \deg r_i, && \text{pour } i \geq 1, \\ \deg b_{i+1} &= \deg f - \deg r_i, && \text{pour } i \geq 0. \end{aligned}$$

DÉMONSTRATION. Les q_i sont de degré strictement positif à partir de $i = 1$. De $a_0 = 1$ et $a_1 = 0$, il vient donc de proche en proche que les degrés des a_i croissent strictement à partir de $i = 1$, avec $\deg a_{i+2} - \deg a_{i+1} = \deg q_i = \deg r_i - \deg r_{i+1}$ pour $i \geq 1$. La première identité annoncée s'obtient maintenant par sommation, en notant que r_1 n'est autre que g. La seconde par une preuve analogue. □

6.3. Formes normales. L'algorithme d'Euclide de la section 6.1 et sa version étendue en section 6.2 permettent divers choix de normalisation par le biais des variables s_i. Ces normalisations permettent d'éviter l'introduction de fractions au cours du calcul et de limiter la croissance de la taille des coefficients des résultats intermédiaires. Dans un cas comme dans l'autre, un bon choix de normalisation permet d'abaisser le temps de calcul, mais une analyse précise de complexité sort des objectifs de ce cours.

7. Approximants rationnels

Un besoin fréquent dans les applications est de pouvoir identifier une fraction rationnelle dont on ne connaît que les premiers termes du développement en série. Plus généralement, ce problème se situe dans le cadre de l'approximation d'une série par une fraction rationnelle. Il s'agit alors de substituer à une donnée infinie un nombre fini de coefficients. L'outil adapté est l'approximant de Padé que nous allons étudier. Cette problématique apparaît par exemple en traitement du signal, où la technique des approximants de Padé permet la synthèse de filtres numériques : numériquement, un approximant rationnel « colle » mieux à une fonction donnée ayant des singularités polaires qu'un approximant de Taylor.

7.1. Approximants de Padé classiques. Pour une série $f = f_0 + f_1 X + \cdots$ à coefficients dans un corps k et des entiers naturels n et m, un *approximant de Padé* de type (m, n) de f est la donnée de polynômes u et v de $k[X]$ satisfaisant aux relations suivantes :

- u a degré inférieur ou égal à m ;
- v a degré inférieur ou égal à n ;
- la relation

$$vf - u \in O(X^{m+n+1}). \tag{2}$$

Ici, $O(X^l)$ ne représente que l'idéal engendré par X^l dans l'anneau des séries, $k[[X]]$. On le distingue ainsi de son homologue polynomial, (X^l).

Observons déjà que $u = f_0 + \cdots + f_d X^d$ et $v = 1$ est un approximant de type $(d, 0)$ de f et qu'en présence d'un approximant de type (m, n), le développement en série de u/v coïncide avec f à l'ordre $m + n + 1 - k$ où k est la valuation de v en X, c'est-à-dire le plus grand exposant k tel que X^k divise v.

Existence. La définition se reformule en termes d'algèbre linéaire pour permettre de montrer l'existence d'approximants de tous types. La relation de définition (2) met en place $m+n+1$ contraintes linéaires. Parmi celles-ci, les n contraintes correspondant aux coefficients des degrés $m+1$ à $m+n$ inclus ne font intervenir que les $n+1$ coefficients de v et fournissent donc toujours au moins une solution v non nulle. Une fois une telle solution v fixée, le polynôme u est donné de façon unique par troncature à l'ordre $m+1$ du produit vf.

Un exemple est donné par la variante

$$x \mapsto \frac{\sin\sqrt{x}}{\sqrt{x}}$$

du sinus cardinal. Sa série de Taylor est donnée à l'ordre 6 par

$$f = 1 - \frac{1}{6}X + \frac{1}{120}X^2 - \frac{1}{5040}X^3 + \frac{1}{362880}X^4 - \frac{1}{39916800}X^5.$$

Un approximant de type $(2, 3)$ fournit la fraction rationnelle

$$\frac{u}{v} = \frac{42(4363920 - 567120X + 12671X^2)}{183284640 + 6728400X + 126210X^2 + 1331X^3)}.$$

Cette fraction fournit un approximant numérique à moins de 10^{-5} près sur tout l'intervalle $[\,0, 6\,]$.

Exercice. Valider ces dires en Magma.

Non-unicité. Concernant l'unicité des approximants de type donné, rien n'impose que le système définissant v dans la preuve d'existence ait un noyau de dimension 1. Ainsi, l'unicité n'est pas garantie, même à multiplication près par une constante non nulle. On se convainc aisément que tout problème d'approximants de type élevé lorsque f est une série rationnelle fournit un exemple de non-unicité. Par exemple, la série $s = 1+2X+2X^2+\cdots$ du développement de $(1+X)/(1-X)$ vérifie $(1-X)s = 1+X$, ce qui fournit l'approximant $(1-X, 1+X)$ de type $(1, 1)$. Il en découle que pour tout λ et μ,

$$u = (\lambda + \mu X)(1 - X) \qquad \text{et} \qquad v = (\lambda + \mu X)(1 + X)$$

donnent un approximant de type $(2, 2)$.

7.2. Approximants de Baker. La définition de la section 7.1 est la plus classique et est dûe à Frobenius et Padé. Elle assure l'existence d'approximants de tous types, mais pas leur unicité, et la fraction u/v n'approxime pas toujours bien la série f. Par exemple, lorsque f est de la forme $1 + X^{11} + \cdots$, un approximant de type $(10, 10)$ est $(u, v) = (X^{10}, X^{10})$, pour lequel $u/v = 1$ n'approxime f qu'à l'ordre 1.

Une définition différente, dûe à Baker, assure une bonne approximation rationnelle et une forme d'unicité : celle de la fraction rationnelle u/v. Mais elle n'assure pas l'existence d'un approximant. À cette fin, au lieu de contraindre $vf - u$, la définition de Baker demande que $f - u/v$ soit de la forme $X^{m+n+1}r$ pour une série r.

En reprenant l'exemple de non-unicité du cas classique en type $(2, 2)$, on voit que le rapport u/v vaut constamment $(1 + X)/(1 - X)$. De manière générale, supposons que des polynômes u_1 et u_2, de degré borné par m, et v_1 et v_2, de degré borné par n, soient tels que $f - u_1/v_1$ et $f - u_2/v_2$ soient tous deux des séries multiples de X^{m+n+1}. Alors il en est de même de la différence $u_1/v_1 - u_2/v_2$, puis, après multiplication par les dénominateurs, de $u_1v_2 - u_2v_1$. Comme ce dernier polynôme est de degré inférieur ou égal à $m + n$, il est nul. D'où l'égalité des rapports u_1/v_1 et u_2/v_2.

Revenons maintenant sur l'exemple $f = 1+X^{11}+\cdots$. Un approximant de type $(10, 10)$ au sens de Baker doit fournir une relation de la forme $f - u/v \in O(X^{21})$, donc la relation $v(1 + X^{11}) - u \in O(X^{21})$. Comme u et v ont degré borné par 10, il n'y a aucune solution non nulle et donc, pas d'approximant au sens de Baker. De manière générale, les approximants de Baker font défaut lorsque les approximants classiques ont le coefficient constant de leur v égal à zéro.

Pour finir, une définition équivalente de la définition de Baker revient à imposer l'inégalité $v(0) \neq 0$ dans la définition classique. En effet, un approximant de Baker peut toujours est supposé donné sous forme u/v irréductible. Comme la série initiale n'a pas d'exposant négatif, le dénominateur v n'a pas X en facteur. La paire (u, v) est alors un approximant classique avec la contrainte $v(0) \neq 0$. Réciproquement, pour un approximant classique avec $v(0) \neq 0$ on a $vf - u \in O(X^{m+n+1})$. Comme v est une série inversible, on a encore $f - u/v \in O(X^{m+n+1})$, et donc un approximant de Baker.

7.3. Calcul d'approximants par l'algorithme d'Euclide. La relation $vf - u = X^{m+n+1}r$ qui définit un approximant classique et se retrouve pour un approximant de Baker implique une relation analogue au niveau des polynômes. Celle-ci est de la forme

$$u = -\bar{r}X^{m+n+1} + v\bar{f}, \tag{3}$$

où $\bar{f}$ est le polynôme obtenu en tronquant la série f à l'ordre $m + n + 1$,

$$\bar{f} = f_0 + \cdots + f_{m+n}X^{m+n},$$

et où $\bar{r}$ est un polynôme troncature de r. La relation (3) exprime que le numérateur u est dans l'idéal *polynomial* engendré par X^{m+n+1} et $\bar{f}$.

Considérons l'algorithme d'Euclide étendu appliqué à X^{m+n+1} et $\bar{f}$ et arrêtons-nous au premier indice i tel que r_i a degré inférieur ou égal à m. On a alors

$$r_i = a_iX^{m+n+1} + b_i\bar{f}$$

avec, par le lemme 3, $\deg a_i = \deg \bar{f} - \deg r_{i-1} < n$. La paire (r_i, a_i) est donc un approximant de Padé.

7.4. Séries génératrices rationnelles. Nous revenons finalement sur les suites linéairement récurrentes.

Pour une suite solution de (1), introduisons la *série génératrice* $f = \sum_{n\geq 0} u_nX^n$ pour montrer que celle-ci est rationnelle, donc reconnaissable par un calcul d'approximant de Padé. Faisons ici l'hypothèse que (1) est d'ordre minimal parmi les relations de récurrence vérifiées par la suite u. Après multiplication par X^{n+r} et sommation sur n, (1) devient

$$a_r(f - u_0 - \cdots - u_{r-1}X^{r-1}) + a_{r-1}(f - u_0)X + \cdots + a_0fX^r = 0.$$

Autrement dit, $(a_r+\cdots+a_0X^r)f$ s'avère être un polynôme de degré au plus $r-1$, et f, une fraction rationnelle. Les numérateurs et dénominateurs de cette dernière constituent un approximant de Padé de type $(r - 1, r)$ de f et peuvent donc être calculés par l'algorithme de la section 7.3.

Exercice. Retrouver les récurrences vérifiées par les suites de nombres de la section 3.1 de façon à prolonger ces suites.

Problème (Suite de Conway). Partant de « 1 », on écrit une suite de mots sur l'alphabet des chiffres obtenus en comptant les chiffres successifs du mot précédent. Le deuxième mot est « 11 » parce qu'il y a un « 1 » dans « 1 ». Ensuite il y a deux « 1 », donc le troisième mot est « 21 », et ainsi de suite. Les premiers mots sont donc

$$1, 11, 21, 1211, 111221, 312211, 13112221, \ldots$$

On considère ensuite la suite des longueurs de ces mots : 1, 2, 2, 4, 6, 6, 8, ... Un joli théorème dû à Conway montre que cette suite des longueurs vérifie une récurrence linéaire à coefficients constants d'ordre 72.

Le problème consiste à retrouver cette récurrence linéaire.

CHAPITRE 3

Algorithmes rapides pour les polynômes et les séries

Pour un système de calcul formel, il est crucial d'optimiser les opérations arithmétiques sur les objets de base — nombres, polynômes, séries, ... En effet, ces opérations élémentaires interviennent ultimement dans la plupart des calculs plus élaborés et consomment bien souvent la majeure partie du temps de calcul. Une accélération des algorithmes de base améliore donc les performances de tout le système.

Ce chapitre présente des algorithmes de meilleure complexité que celle de l'algorithme naïf quadratique (en $O(n^2)$) pour le produit de polynômes. Ainsi, l'algorithme de Karatsuba de la section 1 a pour complexité $O(n^{\log_2 3})$, où $\log_2 3$ vaut à peu près $1,595$; l'algorithme par transformation de Fourier rapide (FFT) de la section 2 a pour complexité $O(n \log n)$, c'est-à-dire une complexité quasi-linéaire. Par ailleurs, la méthode de Newton de la section 4 ramène le coût d'une division de polynômes à celui d'un nombre borné de multiplications, ce quel que soit l'algorithme de multiplication employé.

Comme pour l'algorithme naïf, tous ces algorithmes s'adaptent à d'autres objets comme les entiers, quoiqu'avec une difficulté technique supplémentaire dans les analyses en complexité pour tenir compte des propagations de retenues. De même que dans le chapitre 2, nous nous cantonerons à cette remarque.

1. Algorithme de Karatsuba pour le produit

2. Produit par transformation de Fourier rapide

3. Les fonctions $M(\cdot)$

Les algorithmes de multiplication précédents partagent un certain nombre de propriétés. En notant $M(n)$ la complexité arithmétique dans le pire cas de l'algorithme considéré sur des entrées de taille au plus n, on observe pour chacun de ces algorithmes :

- que la fonction M est croissante ;
- qu'il existe une constante C propre à l'algorithme telle que l'inégalité $M(n) \leq CM(2n)$ pour tout n, la constante 4 pouvant être utilisée pour tous les algorithmes ;
- que la fonction M est sous-additive, au sens où la relation

$$M(n) + M(m) \leq M(n+m)$$

est vérifiée, avec pour conséquence immédiate les inégalités

$$kM(n) \leq M(kn).$$

La notation $M(n)$ a un autre avantage : celui de permettre de donner des complexités en termes d'équivalent en nombre de multiplication de taille n.

4. Itération de Newton pour l'inversion de séries et la division de polynômes

La méthode Newton vise à résoudre une équation $f(x) = 0$ en produisant une suite de valeurs convergeant vers une solution. Plus généralement, on considère une équation de la forme $f_a(x) = 0$ dont on recherche une solution $x = b$, liant ainsi a et b. Pour de bons choix de f, on peut alors exprimer que b est l'inverse de a, sa racine carrée, etc. Par suite, une troncature d'un quotient de séries a/b se calcule par multiplication d'une troncature de a et d'une troncature de $1/b$. Un algorithme rapide de division euclidienne avec reste découle enfin d'un changement de variables entre l'origine et l'infini.

4.1. Inversion de séries. Dans notre contexte, a sera un polynôme de $\mathbb{Q}[X]$ que nous interpréterons comme la troncature d'une série de puissances croissantes, et la convergence à laquelle il a été fait allusion sera une convergence au sens des séries de $\mathbb{Q}[[X]]$; dans le même esprit, la solution b recherchée est une série dont on n'obtiendra qu'une troncature. Pour le calcul de l'inverse de a, la fonction f naturelle est donnée par $f_a(b) = ab - 1$.

Pour une fonction f générale, étant donnée une première approximation b_0 de b, un développement de Taylor de f au voisinage de b_0 s'écrit

$$0 = f_a(b) = f_a(b_0) + f'_a(b_0)(b - b_0) + O\big((b - b_0)^2\big).$$

Heuristiquement, on néglige le terme quadratique en imaginant que b_0 est déjà proche de b, ce qui fournit une nouvelle approximation de b,

$$b_1 = b_0 - f'_a(b_0)^{-1} f_a(b_0).$$

En itérant le processus — en construisant b_2 à partir de b_1 de façon analogue, et ainsi de suite —, la suite des b_n semble pouvoir converger vers b.

Dans le cas de l'inversion donnée par $f_a(b) = ab - 1$, la formule d'itération est $b_{n+1} = b_n - a^{-1}(ab_n - 1)$, où l'on voit qu'elle fait intervenir a^{-1}, c'est-à-dire précisément l'inverse b que l'on veut calculer. L'approximation de a^{-1} par b_n donne finalement l'itération

$$b_{n+1} = b_n - b_n(ab_n - 1).$$

En notant ϵ_n l'erreur $b_n - b$, le calcul donne

$$\epsilon_{n+1} = (b + \epsilon_n)(2 - a(b + \epsilon_n)) - b = -a\epsilon_n^2.$$

Dans le cas numérique, une étude élémentaire montre soit que l'erreur décroît quadratiquement vers 0, soit qu'elle explose quadratiquement, selon que $\big|a(b_0 - b)\big|$ est inférieur ou supérieur à 1. Quant il y a convergence, le nombre de chiffres corrects de l'approximation double donc à chaque étape.

Exercice. Observer numériquement l'itération pour $a = 5$ et b_0 valant tantôt 1 et 0,1.

Dans le cas des séries, la topologie adaptée à ce problème est celle donnée par la norme qui à une série de valuation v associe la valeur 2^{-v}. (La valuation d'une série s est le plus petit entier v tel que s soit dans $O(X^v)$.) Sous l'hypothèse que a a un coefficient constant non nul, deux cas sont alors possibles. Soit b_0 a pour un coefficient constant autre que $1/a(0)$ et c'est alors le cas pour aucun des b_n, si bien que pour tout n, b_n et b sont à distance 1, au sens des séries. Dans le cas contraire, $\epsilon_0 = b - b_0$ est dans $O(X)$; il s'ensuit par récurrence que ϵ_n est dans $O(X^{2^n})$, donc de norme 4^{-n}. Le point important est que le nombre de coefficients corrects de l'approximation double à chaque étape.

Exercice. Observer l'itération pour $a = 5 + 2X$ et b_0 valant tantôt 1, 1/5 et 1/10.

Une conséquence de cette dernière observation est que l'itération se comporte aussi bien si l'on perturbe légèrement les b_n après chaque itération, pour autant qu'on ne touche pas à ses 2^n premiers termes. En particulier, pour le calcul de b_{n+1} à partir de b_n, il est possible de changer a sans atteinte à la convergence tant qu'on ne touche pas aux 2^n premiers termes de a.

Évaluons maintenant le nombre d'opérations arithmétiques nécessaires dans le cas de séries pour réaliser k itérations à partir de $a_0 = 1/b(0)$. Le calcul de b_1 requiert une multiplication en taille 1, une addition en taille 2, puis une multiplication en taille 2, au total un coût arithmétique inférieur à $3M(2)$. De la même façon, le calcul de b_{n+1} à partir de b_n consomme moins de $3M(2^{n+1})$ opérations arithmétiques. Ainsi, les k premières itérations ont un coût arithmétique borné supérieurement par

$$\beta = 3\big(M(2) + M(4) + \cdots + M(2^{k+1})\big).$$

Par sous-additivité, cette borne vérifie

$$\beta \leq 3\left(\frac{1}{2^k} + \cdots + 1\right) M(2^{k+1}).$$

En arrêtant l'itération dès que b_k a précision suffisante, c'est-à-dire, lorsque l'on demande n termes de $b = 1/a$, dès que l'encadrement $n \leq 2^{k+1} < 2n$ a lieu, et donc l'inégalité $2^k < n$, le coût arithmétique est donc borné par $24M(n)$.

[INTRODUIRE L'ALGO D'INVERSION]

4.2. Application à la division euclidienne. Considérons maintenant deux polynômes a et b dont on veut réaliser la division euclidienne. Supposons que a ait un degré d supérieur au degré d' de b et que la division soit posée sous la forme $a = qb + r$, avec r de degré strictement inférieur à d'. On sait que le développement asymptotique à l'infini de la fraction r/b, interprétée comme fonction rationnelle, ne fait intervenir que des puissances négatives de la variable. En posant $X = U^{-1}$, on déduit

$$\frac{a(U^{-1})}{b(U^{-1})} = q(U^{-1}) + \frac{r(U^{-1})}{b(U^{-1})} \in q(U^{-1}) + O(1),$$

où $O(U^k)$ désigne ici l'idéal des séries en U engendré par U^k. Une multiplication par $U^{d-d'}$ fait disparaître les polynômes en U^{-1} :

$$\frac{U^d a(U^{-1})}{U^{d'} b(U^{-1})} \in U^{d-d'} q(U^{-1}) + O(U^{d-d'}),$$

où $\tilde{a} = U^d a(U^{-1})$, $\tilde{b} = U^{d'} b(U^{-1})$ et $\tilde{q} = U^{d-d'} q(U^{-1})$ sont des polynômes. Ce dernier s'interprète comme une troncature à l'ordre $d - d'$ du développement en série en U du quotient $\tilde{a}/\tilde{b}$.

[INTRODUIRE L'ALGO DE DIVISION EUCLIDIENNE]

5. Algorithme rapide pour le p. g. c. d.

6. Algorithme rapide pour le résultant

CHAPITRE 4

Algèbre linéaire

L'algorithme naïf de multiplication de deux matrices $n \times n$ effectue une quantité cubique de multiplications de scalaires ainsi qu'une quantité cubique d'additions de scalaires. Deux directions d'optimisations distinctes sont possibles. Soit, pour des valeurs de n restreintes, chaque multiplication de scalaires coûte significativement plus que chaque addition de scalaires, et il y a un intérêt à remplacer une proportion, même fixe avec n, de multiplications en additions. C'est ce qui est entrepris en section 1. Soit n prend une valeur asymptotique qui interdit un calcul en complexité cubique. L'objectif est alors d'abaisser autant que possible l'exposant ω de la complexité, sachant que celui-ci ne saurait être inférieur à 2, puisqu'il faut bien écrire le résultat. L'algorithme de Strassen de la section 2 est un cas typique d'algorithme du type « diviser pour régner », même s'il ne fournit pas la valeur ω la plus petite connue à ce jour.

La notion d'exposant de l'algèbre linéaire découle de ce que les principaux problèmes d'algèbre linéaire ont une complexité arithmétique asymptotiquement proportionnelle à n^ω pour le même exposant ω. Nous donnons en section 3 un exemple de ces propriétés par l'équivalence en complexité des problèmes de multiplication et d'inversion de matrices.

Une autre forme d'algèbre linéaire n'est pas traitée dans ce cours : l'algèbre linéaire creuse. Dans cet autre modèle, le nombre de coefficients non nuls des matrices de taille n considérées est supposé infiniment petit devant $n^{\omega/2}$, par exemple, de l'ordre de $O(n)$. Les algorithmes de ce chapitre sont alors inefficaces et une jolie algorithmique dédiée a été mise au point pour s'y substituer.

1. Gagner sur la constante de temps : l'algorithme de Winograd

L'algorithme naïf pour le produit de matrices $A = (a_{i,j})_{i,j=1}^n$ et $B = (b_{i,j})_{i,j=1}^n$ calcule le coefficient (i, j) du produit $C = AB$ par la formule

$$c_{i,j} = \sum_{k=1}^{n} a_{i,k} b_{k,j}. \tag{4}$$

Il utilise donc n^3 multiplications et $(n-1)n^2$ additions. Dans les contextes où une multiplication prend plus de temps qu'une addition — c'est le plus souvent le cas —, il est naturel de tenter de réduire le nombre de multiplications, quitte à demander plus d'additions. C'est ce que réalise l'algorithme de Winograd.

L'idée de l'algorithme peut déjà être comprise sur le produit de matrices 2×2 : soit à calculer le produit

$$\begin{pmatrix} a & b \\ c & d \end{pmatrix} \begin{pmatrix} A & B \\ C & D \end{pmatrix} = \begin{pmatrix} aA + bC & aB + bD \\ cA + dC & cB + dD \end{pmatrix}.$$

La méthode consiste à recombiner des coefficients de la même ligne de la première matrice, m, avec des coefficients de la même colonne de la seconde, M. Par exemple, à partir de la première ligne de m et de la première colonne de M, on observe la relation

$$(a + C)(b + A) = (aA + bC) + (ab + AC) \tag{5}$$

qui fait apparaître le terme habituel, « bilinéaire » en les entrées (m, M), ainsi qu'un terme quadratique en celles de m et qu'un terme quadratique en celles de M. Il reste à se débarrasser des termes quadratiques par une correction ad hoc.

L'observation clé est que le précalcul des produits ab et AC permet de réduire les deux multiplications intervenant dans le calcul direct de $aA+bC$ par une seule multiplication, $(a+C)(b+A)$, au prix de quatre additions/soustractions au lieu d'une.

La formule (5) se plonge sans peine dans le calcul du coefficient $c_{i,j}$ du cas général de dimension n : il suffit pour cela de regrouper deux à deux les termes de la somme (4), en appariant par exemple chaque k impair avec le k pair qui le suit. La formule qui remplace (5) s'écrit alors

$$a_{i,2k-1}b_{2k-1,j} + a_{i,2k}b_{2k,j} = (a_{i,2k-1} + b_{2k,j})(a_{i,2k} + b_{2k-1,j}) - a_{i,2k-1}a_{i,2k} - b_{2k-1,j}b_{2k,j}.$$

Lorsque n est pair, l'algorithme de Winograd revient à évaluer la formule précédente pour $1 \leq i \leq n,\ 1 \leq j \leq n,\ 1 \leq k \leq n/2$, et à faire les $(n/2-1)n^2$ additions restantes. Au totale, sa complexité est donc de

- $2n(n/2)$ multiplications de précalcul,
- $n^3/2$ multiplications faites par la boucle sur (i,j,k),
- $4n^3/2$ additions faites par la boucle sur (i,j,k),
- $(n/2-1)n^2$ additions qui recombinent les $c_{i,j}$,

soit $n^3/2 + n^2$ multiplications et $5n^3/2 - n^2$ additions. L'ajustement des termes qui manquent dans le cas où n est impair n'est pas significatif sur la complexité arithmétique.

Pour $n = 2$, le nombre de multiplications reste donc inchangé. Une amélioration, l'algorithme de Waksman, reprend l'idée de Winograd et permet de descendre à sept multiplications. En revanche, dès $n = 4$, le nombre de multiplications passe de 64 à 48, soit un gain d'un quart. Ce gain relativement petit est plus que justifié lorsque le produit est une opération complexe, par exemple, pour le produit de matrices de petites dimensions mais à entrées gigantesques — ne seraient-elles que des entiers.

2. Gagner sur l'exposant de la complexité arithmétique : l'algorithme de Strassen

L'algorithme de Strassen peut être donné brièvement et sa complexité calculée simplement. La véritable difficulté du problème réside dans l'obtention des formules.

2.1. Algorithme et complexité.

Nous ne donnons le résultat qui suit que pour des matrices carrées ; il se généralise sans peine au cas rectangulaire.

ALGORITHME (Multiplication de matrice à la Strassen).

ENTRÉE : deux matrices m et M de dimension n sur un anneau de base commun

SORTIE : le produit de matrices mM

COMPLEXITÉ ARITHMÉTIQUE : $O(n^{\log 7})$, où $\log 7 < 2{,}81$

(1) Si n vaut 1 (respectivement, si n est inférieur à un seuil prédéfini), calculer le produit naïvement (respectivement, par toute méthode directe) et le renvoyer

(2) Écrire les deux matrices par blocs, sous la forme

$$m = \begin{pmatrix} a & b \\ c & d \end{pmatrix} \qquad \text{et} \qquad M = \begin{pmatrix} A & B \\ C & D \end{pmatrix},$$

où les blocs a et A sont dimension $\lfloor n/2 \rfloor$

(3) Calculer les matrices $a-b$, $c-d$, $a+d$, $a+c$, $b+d$, $A+C$, $B+D$, $D-A$, $A+B$ et $C+D$

(4) Calculer récursivement les sept produits de matrices

$$\begin{aligned} q_1 &= (a-b)D, & q_5 &= (a+d)(D-A), \\ q_2 &= (c-d)A, & q_6 &= (a+c)(A+B), \\ q_3 &= d(A+C), & q_7 &= (b+d)(C+D), \\ q_4 &= a(B+D), \end{aligned}$$

(5) Renvoyer la matrice

$$\begin{pmatrix} q_1 - q_3 - q_5 + q_7 & q_4 - q_1 \\ q_2 + q_3 & -q_2 - q_4 + q_5 + q_6 \end{pmatrix}$$

PREUVE DE L'ALGORITHME. La correction de l'algorithme découle d'un simple développement des formules pour montrer que l'on retrouve la formulation classique d'un produit 2×2 par blocs.

Pour l'estimation de complexité, simplifions la présentation en faisant l'hypothèse que n est une puissance de 2. Observons alors qu'en plus des sept multiplications récursives de l'étape (4), toutes autres manipulations ont un coût au plus quadratique en n. Il existe donc une constante C telle que le coût satisfasse à la relation

$$M(n) = 7M(n/2) + Cn^2.$$

L'utilisation de cette relation de manière itérée aboutit aux formules

$$M(n) = 7^2 M(n/2^2) + C(1 + 7/4)n^2 = \cdots = 7^k M(n/2^k) + C\frac{(7/4)^k - 1}{7/4 - 1}n^2.$$

Fixons maintenant k à une valeur qui ramène $n/2^k$ à $s = 1$ (respectivement sous le seuil s considéré par la variante algorithmique). C'est-à-dire que k vaut $\log n$, à une constante près. La formule devient

$$M(n) = M(s)n^{\log 7} + O(n^{2+\log 7/4}) = O(n^{\log 7}).$$

□

Notons que ce qui fait ici fonctionner la récursivité de l'algorithme est le caractère « bilinéaire » des formules : dans la définition des q_i, les coefficients de m interviennent toujours à gauche, ceux de M toujours à droite. A contrario, la formule de Winograd (5) a utilisé la commutativité $Cb = bC$ et c'est ce qui interdit une exploitation récursive de la formule : elle ne s'applique plus quand b et C sont des blocs de taille quelconque.

2.2. Formules explicites. Les formules récursives de Strassen qui donnent un produit de matrice efficace sont comme un lapin tiré du chapeau. Nous essayons ici de les retrouver par une séries de transformations plus ou moins naturelles — mais il faut bien l'avouer, guidées par le résultat qui est connu.

Posons les notations du produit de matrices 2×2 sous la forme :

$$\begin{pmatrix} a & b \\ c & d \end{pmatrix} \begin{pmatrix} A & B \\ C & D \end{pmatrix} = \begin{pmatrix} aA+bC & aB+bD \\ cA+dC & cB+dD \end{pmatrix}$$

Cette écriture utilise huit produits distincts, aA, bC, aB, bD, cA, dC, cB et dD, et fait quatre additions. L'idée est de remplacer les huit produits par un moins grand nombre de produits de la forme $\phi(a,b,c,d)\Phi(A,B,C,D)$ pour des formes linéaires en quatre variables ϕ et Φ ; le coût d'un produit dominant largement celui d'une addition, l'augmentation du nombre d'additions et soustractions ne sera pas un handicap.

Les formules de transformation qui suivent vont permettre de faire apparaître des formes linéaires autres que les huit formes linéaires triviales apparaissant dans la formule naïve, qui sont toutes réduites à une seule variable :

$$xX + yY = x(X-Y) + (x+y)Y,$$
$$x(X+Y) + y(Y+Z) = x(X-Z) + (x+y)(Y+Z).$$

La seconde est en réalité conséquence de la première. Nous utiliserons aussi toutes formes de variations par changements de signes, telle par exemple la formule

$$xX - yY = x(X+Y) - (x+y)Y$$

qui s'obtient en changeant Y en son opposé dans la première formule, ainsi que toute formule obtenue par symétrie en échangeant les rôles de (x,y) et de (X,Y).

Et maintenant, la dérivation. Deux applications des transformations sur les termes antidiagonaux du produit donnent les égalités

$$aB + bD = a(B+D) - (a-b)D,$$
$$cA + dC = (c-d)A + d(A+C).$$

Le travail sur le terme en haut à gauche pour faire apparaître les mêmes produits est plus artificiel : partant d'une première transformation

$$aA + bC = (a-b)A + b(A+C)$$

qui a réutilisé les formes linéaires $a-b$ et $A+C$, mais non les produits $(a-b)D$, $(c-d)A$ et $d(A+C)$, nous choisissons de forcer l'intervention de deux de ces produits en ajoutant et retranchant $\alpha(a-b)D + \beta d(A+C)$, pour des constantes α et β à déterminer. Après factorisation des formes $a-b$ et $A+C$ que nous visons, le coefficient du produit se récrit

$$aA + bC = (a-b)(A-\alpha D) + (b-\beta d)(A+C) + \alpha(a-b)D + \beta d(A+C).$$

Le point inattendu est qu'il est maintenant nécessaire de briser les formes $a-b$ et $A+C$ des produits de gauche, afin d'éviter de revenir sur ses pas. Heureusement, la deuxième transformation s'applique pour donner

$$aA + bC = (a-\beta d)(A-\alpha D) + (b-\beta d)(C+\alpha D) + \alpha(a-b)D + \beta d(A+C).$$

Les trois termes du produit de matrice considérés jusqu'à présent requièrent encore six multiplications. L'économie va avoir lieu en orientant la récriture du quatrième terme de sorte à utiliser les mêmes produits et en procédant de la même méthode :

$$\begin{aligned} cB + dD &= c(B+D) - (c-d)D \\ &= (c-\gamma a)(B+D) - (c-d)(D-\delta A) + \gamma a(B+D) + \delta(c-d)A \\ &= (c-\gamma a)(B+\delta A) + (d-\gamma a)(D-\delta A) + \gamma a(B+D) + \delta(c-d)A. \end{aligned}$$

Pour déterminer les constantes α, β, γ et δ, il suffit d'identifier

$$(a-\beta d)(A-\alpha D) \qquad \text{et} \qquad (d-\gamma a)(D-\delta A)$$

ainsi que

$$(b-\beta d)(C+\alpha D) \qquad \text{et} \qquad (c-\gamma a)(B+\delta A),$$

peut-être à des facteurs constants de proportionnalité près. Un choix simple est

$$\alpha = -\beta = -\gamma = \delta = 1.$$

Il ne fait intervenir aucun inverse de nombre entier et peut ainsi s'utiliser quelle que soit la caractéristique du corps de base.

Au final, la nouvelle expression du produit de matrices 2×2 est de la forme

$$\begin{pmatrix} a & b \\ c & d \end{pmatrix} \begin{pmatrix} A & B \\ C & D \end{pmatrix} = \begin{pmatrix} q_1 - q_3 - q_5 + q_7 & q_4 - q_1 \\ q_2 + q_3 & -q_2 - q_4 + q_5 + q_6 \end{pmatrix}$$

où

$$\begin{aligned} q_1 &= (a-b)D, \qquad q_2 = (c-d)A, \qquad q_3 = d(A+C), \qquad q_4 = a(B+D), \\ q_5 &= (a+d)(D-A), \qquad q_6 = (a+c)(A+B), \qquad q_7 = (b+d)(C+D). \end{aligned}$$

Cette écriture n'utilise plus que sept produits distincts et fait quatorze additions de plus que le produit naïf.

3. Équivalences entre la multiplication de matrices et l'inversion d'une matrice

Dans cette section, on montre l'équivalence du point de vue de la complexité entre les problèmes de multiplication de matrices et d'inversion de matrices. Plus précisément, en notant $M(n)$ la complexité requise pour multiplier deux matrices de taille n et $I(n)$ celle pour inverser une matrice de taille n, les résultats de cette section montrent qu'il existe un algorithme pour la multiplication en complexité $M(n) = O(n^\omega)$ si et seulement s'il existe un algorithme pour l'inversion en complexité $I(n) = O(n^\omega)$, pour le même ω.

Pour ce résultat, on montre d'abord les relations $I(n) = O\big(M(n)\big)$ et $M(n) \leq I(3n)$. Comme la première relation montre que I est polynomiale, $I(3n)$ est d'ordre $O\big(I(n)\big)$, et ainsi, $M(n)$ est d'ordre $O\big(I(n)\big)$.

Pour formuler les résultats de cette section, rappelons que pour une matrice triangulaire supérieure nilpotente N (avec, donc, des 0 sur la diagonale), l'inverse de $I_n - N$ est donné par

$$(I_n - N)^{-1} = \sum_{k \geq 0} N^k = I_n + N + N^2 + \cdots + N^{n-1}.$$

3.1. Inverser se réduit à multiplier. Cette section présente un algorithme d'inversion de matrice qui se réduit essentiellement à deux inversions en taille moitié et à un nombre fixe de multiplications de matrices de taille moitié. Il s'ensuit que la complexité de l'inversion se ramène facilement à la complexité de la multiplication.

La formule

$$\begin{pmatrix} A & B \\ C & D \end{pmatrix} = \begin{pmatrix} 1 & 0 \\ CA^{-1} & 1 \end{pmatrix} \begin{pmatrix} A & 0 \\ 0 & D - CA^{-1}B \end{pmatrix} \begin{pmatrix} 1 & A^{-1}B \\ 0 & 1 \end{pmatrix}$$

se prouve par simple développement du produit par blocs. Elle induit pour l'inverse la formule

$$\begin{pmatrix} A & B \\ C & D \end{pmatrix}^{-1} = \begin{pmatrix} 1 & -A^{-1}B \\ 0 & 1 \end{pmatrix} \begin{pmatrix} A^{-1} & 0 \\ 0 & \big(D - CA^{-1}B\big)^{-1} \end{pmatrix} \begin{pmatrix} 1 & 0 \\ -CA^{-1} & 1 \end{pmatrix}.$$

Ainsi, l'inversion d'une matrice de taille n se calcule en une complexité qui vérifie l'inégalité

$$I(n) \leq 2I(n/2) + \alpha M(n/2) + \beta n^2$$

pour des constantes entières α et β que nous n'explicitons pas. Par itération, l'inégalité devient

$$I(n) \leq 4I(n/4) + \alpha\big(M(n/2) + 2M(n/4)\big) + \beta(1 + 1/2)n^2.$$

Or, M étant au moins quadratique, $M(n/4)$ est au moins quatre fois plus petit que $M(n/2)$, ainsi, l'inégalité se récrit

$$I(n) \leq 4I(n/4) + \alpha(1+1/2)M(n/2) + \beta(1+1/2)n^2.$$

Par itération du processus un nombre $k = \log n$ de fois et par majoration de la série géométrique, on aboutit à la nouvelle inégalité

$$I(n) \leq 2^k I(n/2^k) + 2\alpha M(n/2) + 2\beta n^2 = O(n) + O\big(M(n/2)\big) + O(n^2) = O\big(M(n)\big),$$

ce qu'il fallait prouver.

3.2. Multiplier se réduit à inverser. La simple observation de la formule

$$\begin{pmatrix} I_n & A & 0 \\ 0 & I_n & B \\ 0 & 0 & I_n \end{pmatrix}^{-1} = \begin{pmatrix} I_n & -A & AB \\ 0 & I_n & -B \\ 0 & 0 & I_n \end{pmatrix}$$

prouve l'inégalité $M(n) \leq I(3n)$.

CHAPITRE 5

Bases de Gröbner et applications

1. Introduction

La division euclidienne de polynômes d'une indéterminée et l'algorithme d'Euclide pour le p. g. c. d. sont des outils algorithmiques centraux en algèbre commutative computationnelle[1]. Ces outils offrent des moyens algorithmiques pour calculer sur des idéaux (somme par le p. g. c. d., intersection par le p. p. c. m.) ou modulo un idéal de polynômes (forme normale par la division euclidienne, inversion modulo un polynôme par la relation de Bézout), mais aussi, dualement, sur les nombres algébriques, les zéros de polynômes d'une indéterminée.

On s'attend à trouver la même universalité d'applications pour les polynômes de plusieurs indéterminées, et c'est le cas, avec même plus de richesse. De même que pour les polynômes d'une seule indéterminée, l'algèbre commutative computationnelle fournit des algorithmes pour la mise sous forme canonique des idéaux de polynômes en plusieurs indéterminées, pour une procédure de division avec unicité du reste ou l'obtention de formes canoniques modulo un idéal de polynômes. Une opération supplémentaire est celle de l'élimination polynomiale. Elle consiste à trouver parmi les combinaisons linéaires d'une famille de polynômes donnés avec des coefficients polynomiaux toutes celles qui ne font plus apparaître telle ou telle indéterminée que l'on s'est fixée.

Du point de vue des applications, on retrouve bien des généralisations au cas de plusieurs indéterminées de celles qui viennent d'être abordées pour le cas d'une indéterminée, mais à chaque fois dans un cadre plus élaboré. La division en plusieurs indéterminées, pour ne prendre que cet exemple, nécessite de savoir diviser par toute une famille de polynômes, et non plus par un seul ; dans le cas de reste nul, elle permet d'exprimer un polynôme comme combinaison d'une famille de polynômes donnés avec des coefficients polynomiaux. Avec l'opération d'élimination, on peut de plus aborder, par exemple, la recherche d'une équation implicite décrivant un lieu géométrique donné par une paramétrisation ou la résolution de systèmes polynomiaux de plusieurs indéterminées.

La théorie algorithmique bien adaptée pour le cas de plusieurs indéterminées est la théorie des bases de Gröbner. Cette dernière a initialement été développée pour les idéaux de polynômes de plusieurs indéterminées dans les années 1960 par B. Buchberger, qui lui donna le nom de son directeur de thèse. L'algorithme pour les calculer est aujourd'hui appelé « algorithme de Buchberger ». Sans entrer dans des querelles d'écoles, il nous faut mentionner les travaux antérieurs d'H. Hironaka qui donna une théorie fort similaire pour les idéaux de séries en plusieurs indéterminées. Aujourd'hui, la théorie a été développée dans divers mondes non commutatifs (algèbres de mots, algèbres de groupes, algèbres d'opérateurs linéaires différentiels, de récurrence, etc). Le champ des applications est vaste et varié : géométrie algébrique algorithmique, théorie des invariants, programmation entière, théorie des codes, étude structurelle des équations aux dérivées partielles linéaires et de leur groupes de symétries, étude de systèmes hypergéométriques, manipulation de fonctions spéciales générales, sommation et intégration symboliques, preuve de théorèmes géométriques assistée par ordinateur, ...

Malgré le demi-siècle d'existence de l'algorithme de Buchberger, sa complexité algorithmique reste encore mal connue. On s'en est longtemps tenu à évoquer sa complexité au pire, doublement exponentielle. Certains l'invoquent encore pour refuser d'utiliser les bases de Gröbner. Pourtant,

[1]On ne trouve pas ce mot, de l'anglais *computational*, dans le dictionnaire. On dit aussi « effective », mais l'auteur a une préférence pour « computationnel », qui rappele plus fortement que le calcul peut vraiment se faire sur ordinateur, *computer* en anglais.

des progrès récents portant sur l'implantation et l'algorithmique permettent de manipuler des systèmes gigantesques. On sait depuis assez longtemps que la complexité au pire n'est que simplement exponentielle dans les cadres d'applications les plus fréquents. La recherche en cours semble être sur le point de pouvoir donner des résultats de complexité en moyenne, indiquant eux aussi une complexité simplement exponentielle en la taille de l'entrée, et par ailleurs une complexité polynomiale en la taille de la sortie sur des entrées génériques.

Les algorithmes et les applications du cas de plusieurs indéterminées vont être présentés en deux temps. Dans ce chapitre, nous présentons la théorie des bases de Gröbner ; nous nous donnons l'algorithme de Buchberger comme une boîte noire et verrons comment l'utiliser dans les applications. La chapitre suivant sera consacrée à détailler l'algorithme de Buchberger.

Dans la section 2, nous rappelons la définition et les premières propriétés algébriques des idéaux. En section 3, nous posons quatre problèmes qui trouveront une solution algorithmique ou au moins constructive par la théorie des bases de Gröbner. La section 4 introduit les ordres sur les monômes qui vont remplacer l'ordre des puissances décroissantes du cas d'une indéterminée. La division euclidienne trouve en section 5 son pendant en plusieurs indéterminées. Nous sommes alors prêts pour définir les bases de Gröbner en section 6. Nous terminons la première partie en section 7 en explicitant des méthodes algorithmiques qui répondent aux problèmes posés en début de texte.

2. Idéaux de polynômes

De même que l'ensemble de tous les multiples d'un polynôme donné dans le cas univarié, l'objet algébrique à la base de la théorie est ici l'ensemble de toutes les combinaisons linéaires à coefficients polynomiaux d'une famille de polynômes donnés, appelé un « idéal ». Rappelons qu'un *idéal* I d'un anneau commutatif unitaire A est un sous-groupe de A stable par l'action (en général notée comme une multiplication) par tout élément de A. Étant donnée une famille $(g_u)_{u\in U}$ d'éléments de A, les combinaisons linéaires finies à coefficients dans A forment un idéal noté $\sum_{u\in U} Ag_u$. On montre que tout idéal est de cette forme. Dans cette présentation, les éléments de la famille sont appelés *générateurs* de I. Insistons bien sur le terme « fini » de la définition, les sommes infinies n'étant pas toutes susceptibles de se sommer dans un anneau général.

Dans le cadre qui nous intéresse, celui d'un anneau de polynômes de la forme $\bar{\mathbb{Q}}[X_1,\dots,X_n]$, pour une clôture algébrique $\bar{\mathbb{Q}}$ de $\mathbb{Q}$, on peut donc tout d'abord se donner un idéal par des générateurs $g_u \in \bar{\mathbb{Q}}[X_1,\dots,X_n]$. Une des questions qu'il faudra se poser est de savoir si un idéal polynomial peut être engendré par un nombre fini de générateurs. Un autre mode de présentation des idéaux de polynômes fait le lien avec la géométrie : l'ensemble des polynômes de $\bar{\mathbb{Q}}[X_1,\dots,X_n]$ qui s'annulent sur un ensemble donné V de points de $\bar{\mathbb{Q}}^n$,

$$I(V) = \big\{\, p \in \bar{\mathbb{Q}}[X_1,\dots,X_n] : \forall x = (x_1,\dots,x_n) \in V,\ p(x) = 0 \,\big\}, \tag{6}$$

est un idéal appelé l'*idéal annulateur* de V.

Rappelons maintenant deux opérations élémentaires sur les idéaux généraux. Étant donnés deux idéaux I et J d'un anneau A, la *somme* $I+J$ est l'idéal engendré par l'union $I\cup J$. Lorsque les idéaux sont donnés par des familles de générateurs, la somme est donnée par l'union de ces familles. L'intersection des idéaux I et J est un idéal. Il n'y a pas de lien explicite immédiat entre une famille de générateurs pour $I\cap J$ et des familles de générateurs pour I et J, mais un algorithme existe dans le cas d'idéaux de polynômes.

3. Quelques problèmes sur les idéaux de polynômes

Dorénavant, sauf mention expresse du contraire, tous les idéaux sont des idéaux d'un anneau de polynômes $\bar{\mathbb{Q}}[X_1,\dots,X_n]$, que l'on notera A_n dans la suite. Nous posons maintenant quatre problèmes qui trouverons une solution algorithmique par la théorie des bases de Gröbner (ou au moins constructive pour le premier). Nous y reviendrons ultérieurement pour expliciter ces solutions.

3.1. Finitude de la présentation d'un idéal. On l'a dit, un idéal peut toujours être vu comme engendré par une famille ; plus précisément, on a toujours la relation triviale $I = \sum_{p\in I} A_n p$. Une question intéressante est de savoir quand on peut se limiter à une somme finie. Notons d'abord que la question n'est absolument pas évidente pour un idéal donné comme annulateur d'un ensemble algébrique, par la définition (6).

D'autre part, le résultat ne peut être vrai pour un anneau général, même commutatif. Par exemple, il est faux pour l'anneau de polynômes $A_\infty = k[X_0, X_1, \dots]$ en une infinité (dénombrable) d'indéterminées. (Pour se convaincre que cet anneau existe bien, on peut se le représenter comme l'union sur n des A_n.) Chaque élément de A_∞ est une somme finie qui tombe dans un des A_n, donc chaque élément ne fait intervenir qu'un nombre fini d'indéterminées. Montrons que l'idéal $I = \sum_{i\in\mathbb{N}} A_\infty X_i$ n'est pas finiment engendré. Pour ce faire, considérons la chaîne infinie d'inclusions $I_0 \subseteq I_1 \subseteq I_2 \subseteq \cdots$ pour les idéaux $I_n = \sum_{i=0}^{n} A_\infty X_i$ de A_∞. Cette chaîne est infinie strictement croissante, car chaque X_n est dans $I_n \setminus I_{n-1}$. Sinon, en exprimant X_n comme combinaison de la forme $p_0X_0 + \cdots + p_{n-1}X_{n-1}$ puis en spécialisant chaque X_i pour $i < n$ à 0, nous obtiendrions une contradiction. Si maintenant l'idéal I était engendré par un nombre fini de générateurs, on trouverait un I_n les contenant tous. Donc nous aurions l'inclusion $I \subseteq I_n$, donc l'égalité $I = I_k$ pour $k \geq n$, ce qui serait une contradiction.

Dans le cas d'une indéterminée et de A_1, les idéaux sont principaux (peuvent être présentés comme engendrés par un unique générateur). Nous verrons que, pour les anneaux A_n, un nombre fini de générateurs suffit.

3.2. Appartenance à un idéal. Soient un idéal I de $A_n = \bar{\mathbb{Q}}[X_1, \dots, X_n]$, donné par une famille finie de générateurs, et un polynôme $p \in A$. Comment déterminer algorithmiquement si p est dans I ? Dans l'affirmative, comment calculer algorithmiquement une représentation de p comme combinaison linéaire à coefficients dans A_n des générateurs de I ?

Par exemple, $X^3 - 1$ est-il combinaison linéaire sur $\bar{\mathbb{Q}}[X, Y, Z]$ de $X + Y + Z$, $XY + YZ + ZX$ et de $XYZ - 1$? La réponse est positive, car

$$X^3 - 1 = (X^2 - XY - XZ - YZ)(X + Y + Z) + (Y + Z)(XY + YZ + ZX) + 1 \times (XYZ - 1).$$

Cependant, on imagine bien qu'une telle décomposition n'est pas unique : il suffit d'ajouter terme à terme l'égalité

$$0 = (XY + YZ + ZX)(X + Y + Z) + (-X - Y - Z)(XY + YZ + ZX) + 0 \times (XYZ - 1)$$

pour obtenir une autre décomposition. Nous verrons comment obtenir de manière générale et algorithmique une décomposition qui sera minimale en un certain sens.

Un lien avec la géométrie se fait par la notion de « variété affine » : pour un ensemble S de polynômes, on définit l'*ensemble algébrique* (ou *variété affine*)

$$V(S) = \big\{\, x = (x_1, \dots, x_n) \in \bar{\mathbb{Q}}^n : \forall f \in S,\ f(x) = 0 \,\big\}.$$

Lorsque S est fini, l'ensemble $V\big(\sum_{i=1}^{s} Ap_i\big)$ est noté plus simplement $V(p_1, \dots, p_s)$. Le problème de l'appartenance d'un polynôme p à l'idéal I de A_n s'interprète alors comme le problème équivalent de l'inclusion $V(I) \subseteq V(p)$. (Remarquons qu'on a pris soin de travailler sur $\bar{\mathbb{Q}}$ et non sur $\mathbb{R}$ pour éviter tout problème crée par une équation polynomiale sans solution réelle, telle $X^2 + 1 = 0$.

3.3. Résolution de systèmes polynomiaux. Un problème qui revient sans cesse dans toutes sortes d'applications est celui de la résolution d'un système polynomial. Il s'agit d'obtenir une description de toutes les solutions dans $\bar{\mathbb{Q}}^n$ d'un système de la forme

$$p_1(x_1, \dots, x_n) = \cdots = p_s(x_1, \dots, x_n) = 0.$$

Dans le cas général, un tel système n'a pas un ensemble fini de solutions ; on s'intéressera alors à donner une description paramétrique des solutions.

L'approche par la théorie des bases de Gröbner ne fournit pas la solution la plus efficace, mais certainement l'approche la plus simple pour traiter le cas général et rechercher des expressions exactes pour les solutions, c'est-à-dire en excluant le calcul numérique. Donnons un exemple : on recherche le lieu et les valeurs des extrema sur la sphère unité d'équation $x^2 + y^2 + z^2 = 1$ de

l'application polynomiale $(x, y, z) \mapsto x^3 + 2xyz - z^2$. La méthode des multiplicateurs de Lagrange indique que les positions (x, y, z) des extrema vérifient le système polynomial

$$3X^2 + 2YZ = 2\Lambda X, \qquad 2XZ = 2\Lambda Y, \qquad 2XY - 2Z = 2\Lambda Z, \qquad X^2 + Y^2 + Z^2 = 1,$$

pour une nouvelle indéterminée Λ, le multiplicateur de Lagrange. Nous verrons comment aborder la résolution de tels systèmes par triangularisation.

3.4. Équations implicites d'un lieu géométrique donné par une paramétrisation. Le problème est le suivant : étant donnée une paramétrisation rationnelle

$$x_i = r_i(t_1, \dots, t_m), \quad i = 1, \dots, n, \quad (t_1, \dots, t_m) \in \bar{\mathbb{Q}}^m$$

d'un ensemble V de points de $\bar{\mathbb{Q}}^n$, trouver algorithmiquement un système d'équations polynomiales qui définisse V sans plus faire référence aux t_i (ou du moins le plus petit ensemble algébrique contenant V).

Le prototype de ce problème est celui de la paramétrisation du cercle unité. Partant de la paramétrisation $t \mapsto \big(x(t), y(t)\big)$ donnée par

$$x = \frac{1 - t^2}{1 + t^2} \quad \text{et} \quad y = \frac{2t}{1 + t^2},$$

il s'agit de calculer la relation implicite $x^2 + y^2 = 1$.

4. Monômes et ordre monomial

Notre premier objectif est de donner une généralisation de la division euclidienne. Dans le cas d'une indéterminée, il existe en fait deux divisions polynomiales : la plus classique, utilisée en arithmétique, est celle par les puissantes décroissantes et termine toujours ; mais il y a aussi une division par les puissantes croissantes, laquelle le plus souvent ne termine pas et est en fait mieux adaptée à décrire une division entre séries formelles. Ces deux modes de division sont rattachés à deux ordres sur les exposants entiers des polynômes : respectivement, l'ordre décroissant et l'ordre croissant sur $\mathbb{N}$. Bien qu'il existe de nombreux autres ordres sur $\mathbb{N}$, ces deux ordres sont les seuls compatibles avec le produit de polynômes, au sens où le « terme de tête », respectivement de plus haut ou de plus petit degré, d'un produit de deux polynômes est le produit des termes de têtes de ces deux polynômes.

En plusieurs indéterminées, on rencontre une première différence : toute une variété de ce que l'on va bientôt appeler « ordres monomiaux » est disponible pour définir une division, même en s'imposant de bonnes propriétés de compatibilité avec le produit. Loin d'être une difficulté, on exploitera cette diversité dans les applications.

4.1. Terminologie sur les polynômes. Fixons d'abord notre terminologie sur les polynômes, celle promue par le courant « algébriste », maintenant la plus usitée dans le contexte des bases de Gröbner, et qui a supplanté la terminologie plus ancienne introduite par le courant « logicien ».

Un *monôme* m sur des indéterminées $X_1, \dots, X_n$ est un produit (fini) des X_i, éventuellement avec répétitions pour permettre tout exposant entier. On note qu'un monôme m est de la forme $X_1^{\alpha_1} \dots X_n^{\alpha_n}$ et ne comporte pas de coefficient. Un polynôme p de A_n est donc une combinaison linéaire de monômes à coefficients dans $\bar{\mathbb{Q}}$. Il s'écrit alternativement sous l'une des formes

$$p = \sum_{\text{finie}} p_{\alpha_1,\dots,\alpha_n} X_1^{\alpha_1} \dots X_n^{\alpha_n} \qquad \text{et} \qquad p = \sum_{j=1}^{s} c_j m_j$$

pour des monômes m_j et des scalaires $p_{\alpha_1,\dots,\alpha_n}$ et c_j. Plus précisément, $p_{\alpha_1,\dots,\alpha_n}$ est le coefficient de $X_1^{\alpha_1} \dots X_n^{\alpha_n}$ dans p, et c_j celui de m_j dans p. Les termes non nuls de l'une ou l'autre des représentations de p comme somme sont appelés *termes* de p. Ce sont des produits d'un scalaire par un monôme.

4.2. Monoïde des monômes. Il sera commode et fructueux de considérer la structure de l'ensemble des monômes d'un anneau A_n donné. C'est celle d'un « monoïde ».

Un *monoïde* M est un ensemble muni d'une loi interne associative pour laquelle il existe un élément neutre. L'exemple le plus simple est celui de l'ensemble $\mathbb{N}$ des nombres entiers, muni de l'addition usuelle, et avec 0 pour neutre. Une généralisation immédiate est celle de $\mathbb{N}^n$, avec l'addition terme à terme. La formule d'addition,

$$(a_1, \ldots, a_n) + (b_1, \ldots, b_n) = (a_1 + b_1, \ldots, a_n + b_n),$$

reflète l'addition des exposants dans un produit de monômes dans A_n. En fait, les monômes de A_n constituent un monoïde commutatif M, isomorphe au précédent, engendré par les indéterminées X_1, …, X_n, ayant pour loi interne le produit usuel de A_n et de neutre $1 = X_1^0 \ldots X_n^0$. La loi produit explicite est donnée par

$$\left(X_1^{a_1} \ldots X_n^{a_n}\right) \times \left(X_1^{b_1} \ldots X_n^{b_n}\right) = X_1^{a_1+b_1} \ldots X_n^{a_n+b_n}.$$

C'est le *monoïde commutatif libre* sur les n générateurs X_1, …, X_n, noté $[X_1, \ldots, X_n]$. Cette notation qui n'est pas classique fait référence à l'absence de coefficients dans les monômes.

4.3. Ordres monomiaux, exemples principaux.

Définition (Ordre monomiaux). Un *ordre monomial* sur M est une relation d'ordre strict $\prec$ qui est :

- totale : deux monômes peuvent toujours être comparés ;
- compatible avec le produit : dès lors que $m_1 \prec m_2$, on a $m'm_1 \prec m'm_2$ pour tout m' ;
- un bon ordre : tout ensemble non vide de monômes a un plus petit élément, ou de façon équivalente, toute suite strictement décroissante de monômes termine.

En particulier, pour tout ordre monomial, on a la relation $1 \prec X_i$ pour chaque i. Sinon nous aurions $X_i \prec 1$ pour un certain i, puis de proche en proche $X_i^{k+1} \prec X_i^k$, d'où une suite infinie strictement décroissante. En conséquence, $1 = X_1^0 \ldots X_n^0$ est le plus petit élément de M pour tout ordre monomial, car tout monôme m peut être obtenu comme dernier élément d'une chaîne

$$1 \prec X_{i_1} \prec X_{i_1} X_{i_2} \prec \cdots \prec X_{i_1} \ldots X_{i_r}.$$

Pour la suite, nous adoptons les notations $|\alpha| = \alpha_1 + \cdots + \alpha_n$, $X^\alpha = X_1^{\alpha_1} \ldots X_n^{\alpha_n}$ et $X^{\alpha+\beta} = X^\alpha X^\beta$ pour tous multi-exposants $\alpha = (\alpha_1, \ldots, \alpha_n)$ et $\beta = (\beta_1, \ldots, \beta_n)$. On notera $\preceq$ l'ordre large associé à l'ordre strict $\prec$.

Nous donnons maintenant les exemples principaux de relations d'ordres employées sur des monômes. Pour les besoins de la définition, nous présentons simultanément trois ordres monomiaux et un ordre qui n'est pas un ordre monomial, l'ordre lexicographique renversé.

- *ordre lexicographique* (ordre du dictionnaire) : $X^\alpha \prec_{\mathsf{lex}} X^\beta$ si $\alpha_k < \beta_k$ pour $k = \min\{\, i : \alpha_i \neq \beta_i \,\}$, ou autrement dit, si la première valeur non nulle de la suite $\alpha_1 - \beta_1$, $\alpha_2 - \beta_2$, … est strictement négative.
- *ordre lexicographique gradué* (ordre du degré total raffiné par $\prec_{\mathsf{lex}}$) : $X^\alpha \prec_{\mathsf{grlex}} X^\beta$ si $|\alpha| < |\beta|$ ou ($|\alpha| = |\beta|$ et $X^\alpha \prec_{\mathsf{lex}} X^\beta$).
- *ordre lexicographique renversé* (n'est pas un ordre monomial) : $X^\alpha \prec_{\mathsf{revlex}} X^\beta$ si $\alpha_k > \beta_k$ pour $k = \max\{\, i : \alpha_i \neq \beta_i \,\}$, ou autrement dit, si la dernière valeur non nulle de la suite $\alpha_1 - \beta_1$, $\alpha_2 - \beta_2$, … est strictement positive.
- *ordre lexicographique renversé gradué* (ordre du degré total raffiné par $\prec_{\mathsf{revlex}}$) : $X^\alpha \prec_{\mathsf{grevlex}} X^\beta$ si $|\alpha| < |\beta|$ ou ($|\alpha| = |\beta|$ et $X^\alpha \prec_{\mathsf{revlex}} X^\beta$).

Pour tous ces ordres, on vérifie la relation $X_1 \succ X_2 \succ \cdots \succ X_n$. Il est aisé de confondre ces différents ordres, aussi donnons-nous explicitement les quelques premiers termes de la suite ordonnée des monômes pour chacun de ces ordres. Il est nécessaire d'avoir au moins trois indéterminées ($n = 3$) et d'aller jusqu'en degré trois pour mettre en évidence les différences entre les ordres.

- *ordre lexicographique,* $\prec_{\mathsf{lex}}$: $1 \prec X_3 \prec X_3^2 \prec X_3^3 \prec \cdots \prec X_2 \prec X_2X_3 \prec X_2X_3^2 \prec X_2X_3^3 \prec \cdots \prec X_2^2 \prec X_2^2X_3 \prec X_2^2X_3^2 \prec X_2^2X_3^3 \prec \cdots \prec X_1 \prec X_1X_3 \prec X_1X_3^2 \prec X_1X_3^3 \prec \cdots \prec$

$X_1X_2 \prec X_1X_2X_3 \prec X_1X_2X_3^2 \prec X_1X_2X_3^3 \prec \cdots \prec X_1X_2^2 \prec X_1X_2^2X_3 \prec X_1X_2^2X_3^2 \prec X_1X_2^2X_3^3 \prec \cdots$.

– *ordre lexicographique gradué,* $\prec_{\mathsf{grlex}} : 1 \prec X_3 \prec X_2 \prec X_1 \prec X_3^2 \prec X_2X_3 \prec X_2^2 \prec X_1X_3 \prec X_1X_2 \prec X_1^2 \prec X_3^3 \prec X_2X_3^2 \prec X_2^2X_3 \prec X_2^3 \prec X_1X_3^2 \prec X_1X_2X_3 \prec X_1X_2^2 \prec X_1^2X_3 \prec X_1^2X_2 \prec X_1^3 \prec \cdots$.

– *ordre lexicographique renversé,* $\prec_{\mathsf{revlex}} : \cdots \prec X_1^3X_2^2X_3 \prec X_1^2X_2^2X_3 \prec X_1X_2^2X_3 \prec X_2^2X_3 \prec \cdots \prec X_1^3X_2X_3 \prec X_1^2X_2X_3 \prec X_1X_2X_3 \prec X_2X_3 \prec \cdots \prec X_1^3X_3 \prec X_1^2X_3 \prec X_1X_3 \prec X_3 \prec \cdots \prec X_1^3X_2^2 \prec X_1^2X_2^2 \prec X_1X_2^2 \prec X_2^2 \prec \cdots \prec X_1^3X_2 \prec X_1^2X_2 \prec X_1X_2 \prec X_2 \prec \cdots \prec X_1^3 \prec X_1^2 \prec X_1 \prec 1$.

– *ordre lexicographique renversé gradué,* $\prec_{\mathsf{grevlex}} : 1 \prec X_3 \prec X_2 \prec X_1 \prec X_3^2 \prec X_2X_3 \prec X_1X_3 \prec X_2^2 \prec X_1X_2 \prec X_1^2 \prec X_3^3 \prec X_2X_3^2 \prec X_1X_3^2 \prec X_2^2X_3 \prec X_1X_2X_3 \prec X_1^2X_3 \prec X_2^3 \prec X_1X_2^2 \prec X_1^2X_2 \prec X_1^3 \prec \cdots$.

Encore une fois, l'ordre $\prec_{\mathsf{revlex}}$ n'est pas un ordre monomial : il fournit une suite infinie décroissante de monômes. En revanche, les ordres $\prec_{\mathsf{lex}}$, $\prec_{\mathsf{grlex}}$, $\prec_{\mathsf{grevlex}}$ sont des ordres monomiaux. Nous laissons la preuve au lecteur. Les ordres monomiaux $\prec_{\mathsf{lex}}$ et $\prec_{\mathsf{grevlex}}$ seront les plus employés, ainsi que d'autres ordres pondérés et d'élimination de bloc. Pour éviter de renommer les indéterminées, on utilisera la notation $\prec_{\mathsf{lex}(X_{\sigma(1)},\dots,X_{\sigma(n)})}$, etc. Ainsi par exemple, les ordres $\prec_{\mathsf{lex}(X_1,\dots,X_n)}$ et $\prec_{\mathsf{lex}}$, sont identiques. Par ailleurs, on a pour tous monômes X^α et X^β l'équivalence « $X^\alpha \prec_{\mathsf{revlex}(X_1,\dots,X_n)} X^\beta$ si et seulement si $X^\alpha \succ_{\mathsf{lex}(X_n,\dots,X_1)} X^\beta$ ». Là encore, nous invitons le lecteur à bien se convaincre de ce point technique.

4.4. Coefficients, monômes et termes de tête. Faire choix d'un ordre monomial $\prec$ sur M permet d'associer des monôme, exposant, coefficient et terme distingués à tout polynôme non nul. En raison de la compatibilité d'un ordre monomial avec le produit, ces éléments distingués se comportent bien face au produit.

Plus précisément, le *monôme de tête* (ou *monôme dominant*) d'un polynôme p non nul est le plus grand monôme de coefficient non nul dans p. Il est noté $\mathrm{mt}(p)$. Le *coefficient de tête* (ou *coefficient dominant*) d'un polynôme p non nul est le coefficient noté $\mathrm{ct}(p)$ de son monôme de tête. Le *terme de tête* (ou *terme dominant*) d'un polynôme p non nul est le produit noté $\mathrm{tt}(p)$ de son coefficient de tête par son monôme de tête.

Évidemment, ces notions sont relatives à l'ordre monomial choisi. Donnons l'exemple dans A_3 du polynôme

$$p = -30X_1X_2^2 - 210X_2^2X_3 + 3X_1^2 + 35X_2^2 + 30X_1X_3 - 105X_3^2 + 140X_2X_4 - 21X_5.$$

Son terme de tête est :

– le terme $-30X_1X_2^2$ pour $\mathsf{grevlex}(X_1,\dots,X_5)$ et $\mathsf{grlex}(X_1,\dots,X_5)$,
– le terme $3X_1^2$ pour $\mathsf{lex}(X_1,\dots,X_5)$,
– le terme $-21X_5$ pour $\mathsf{lex}(X_5,\dots,X_1)$.

Faisons maintenant le lien avec le produit. Par définition d'un ordre monomial, le monôme de tête d'un produit de deux polynômes p et q est le produit des monômes de tête $\mathrm{mt}(p)$ et $\mathrm{mt}(q)$. En effet, notons $p = c_0m_0 + \cdots + c_rm_r$ et $q = c'_0m'_0 + \cdots + c'_sm'_s$ pour deux suites de coefficients non nuls c_i et c'_i et deux suites strictement décroissantes de monômes m_i et m'_i. (En particulier, $c_0 = \mathrm{ct}(p)$, $m_0 = \mathrm{mt}(p)$, et on a les relations analogues pour q.) Ainsi,

$$pq = \sum_{i=0}^{r}\sum_{j=0}^{s} c_im_ic'_jm'_j = \sum_{i=0}^{r}\sum_{j=0}^{s} c_ic'_jm_im'_j.$$

Or, on a $m_i \preceq m_0$ et $m'_j \preceq m'_0$, d'où $m_im'_j \preceq m_0m'_j$ et $m_0m'_j \preceq m_0m'_0$, et donc $m_im'_j \preceq m_0m'_0$. L'égalité ne peut avoir lieu que si $i = j = 0$, ce qui prouve le résultat. Au passage, on a aussi montré que les coefficients de tête se multiplient, de même que les termes de tête.

Cette propriété sur les monômes de tête est à la base de toute la théorie commutative des bases de Gröbner et persiste dans presque tous les cas de généralisations à des cadres non commutatifs. Une variante importante sera celle des algèbres « tordues » dans lesquelles coefficients et indéterminées ne commutent pas librement. Chaque produit $m_ic'_j$ se récrit alors comme un polynôme dont le monôme de tête est m_i, mais qui n'est plus forcément réduit à ce seul monôme

de tête, et dont le coefficient de tête ne sera plus forcément c'_j. Dans ces variantes, la propriété sur les monômes de tête sera préservée, mais pas celles sur les coefficients et termes de tête.

5. Réduction et division en plusieurs indéterminées

La division euclidienne d'un polynôme f en une seule indéterminée X par un polynôme g non nul en la même indéterminée exprime f sous la forme $qg + r$ pour des polynômes q et r tel que r ait degré inférieur à g. Le reste r est l'unique représentant de la classe de f modulo l'idéal $A_1 g$ ayant cette propriété de degré, et l'on peut dire que l'on a divisé f par l'idéal $A_1 g$. Dans le cas d'un anneau de polynômes en plusieurs indéterminées, dans lequel les idéaux ne sont plus nécessairement principaux, il est donc naturel d'autoriser une division par toute une famille de diviseurs $g_1, \ldots, g_s$, l'objectif étant ainsi d'exprimer f sous la forme $q_1 g_1 + \cdots + q_s g_s + r$.

5.1. Réduction. Algorithmiquement, la division euclidienne procède par une succession de « divisions élémentaires », où l'on ne considère que des quotients q qui sont des monômes et donc des restes r qui ne sont pas minimaux au sens du degré. Dans le cas de plusieurs indéterminées, ces étapes sont appelées « réductions ».

On fixe un ordre monomial sur un anneau de polynômes A_n. Un polynôme f non nul de A_n est dit *réductible* par un polynôme g non nul de A_n si $\mathrm{mt}(g)$ divise $\mathrm{mt}(f)$ dans A_n. Un polynôme f non nul de A_n est dit *réductible* par une famille $\{g_i\}_{i\in\{1,\ldots,s\}}$ de polynômes non nuls de A_n si f est réductible par l'un des g_i dans A_n.

En une unique indéterminée, f est réductible par g si et seulement si $\deg f \geq \deg g \geq 0$. En plusieurs indéterminées, remarquons que la notion de divisibilité d'un monôme m par un monôme m' n'est pas équivalente à la relation d'ordre $m' \preceq m$.

Lorsqu'un polynôme est réductible, on va pouvoir le « réduire ». *Réduire* un polynôme f non nul de A_n par un polynôme g non nul de A_n, c'est remplacer f par $f' = f - cmg$ pour $m = \mathrm{mt}(f)/\mathrm{mt}(g)$ et $c \in \bar{\mathbb{Q}}$ de façon que $f' = 0$ ou que $\mathrm{mt}(f') \prec \mathrm{mt}(f)$. Autrement dit, c'est remplacer f par $f' = f - tg$ pour $t = \mathrm{tt}(f)/\mathrm{tt}(g)$. *Réduire* un polynôme f non nul de A_n par une famille $\{g_i\}_{i\in\{1,\ldots,s\}}$ de polynômes non nuls de A_n, c'est réduire f par un des g_i convenable. Notons dès à présent que rien n'impose le choix de i dans les cas d'ambiguïté ou plusieurs g_i permettent la réduction.

5.2. Division en plusieurs indéterminées. *Diviser* un polynôme f non nul de A_n par un polynôme g non nul de A_n, respectivement par une famille $\{g_i\}_{i\in\{1,\ldots,s\}}$ de polynômes non nuls de A_n, c'est le réduire autant de fois que nécessaire jusqu'à aboutir à un polynôme irréductible par g, respectivement par la famille $\{g_i\}_{i\in\{1,\ldots,s\}}$.

On l'a fait observer, divisibilité et ordre ne sont pas des notions équivalentes. Il se peut même qu'un polynôme f ne soit pas réductible par un polynôme g mais qu'un terme de f, autre que le terme de tête, soit lui réductible par g. L'algorithme qui suit propose une division qui renvoie un reste dont tous les termes sont irréductibles.

Algorithme (Division en plusieurs indéterminées).

Entrée : un polynôme f et des polynômes non nuls $g_1, \ldots, g_s$

Sortie : des polynômes $r, q_1, \ldots, q_s$ tels que $f = q_1 g_1 + \cdots + q_s g_s + r$ et tel qu'aucun monôme de r ne soit réductible par $\{g_i\}_{i\in\{1,\ldots,s\}}$

(1) $r \leftarrow 0$; pour i de 1 à s, faire $q_i \leftarrow 0$

(2) tant que $f \neq 0$, faire
- si $\mathrm{mt}(g_i)$ divise $\mathrm{mt}(f)$ pour un certain i, choisir un tel i et faire $q_i \leftarrow q_i + \mathrm{tt}(g_i)^{-1}\,\mathrm{tt}(p)$ et $f \leftarrow f - \mathrm{tt}(g_i)^{-1}\,\mathrm{tt}(p) g_i$
- sinon, faire $r \leftarrow r + \mathrm{tt}(f)$ et $f \leftarrow f - \mathrm{tt}(f)$

(3) renvoyer $r, q_1, \ldots, q_s$

Preuve de l'algorithme. La procédure qui précède est correcte, car elle respecte l'invariant $f = q_1 g_1 + \cdots + q_s g_s + r$ sur les variables du calcul et s'arrête lorsque f est nul et après n'avoir accumulé dans r que des monômes irréductibles. Elle termine pour un ordre monomial car le monôme $\mathrm{mt}(f)$ décroît strictement à chaque passage dans la boucle « tant que » et qu'il ne peut y avoir de suite infinie décroissante de monômes pour un ordre monomial. □

Cet algorithme n'est pas déterministe, en conséquence du choix laissé sur i pour chaque réduction. Ce non déterminisme de la division s'observe bien sur l'exemple suivant, où l'on donne deux divisions pour l'ordre monomial $\mathsf{lex}(X, Y)$ de $f = \underline{X^2Y} + XY^2 + Y^2$ par la famille constituée de $g_1 = \underline{XY} - 1$ et de $g_2 = \underline{Y^2} - 1$. (Pour facilité la lecture, on a souligné les monômes de tête.) En réduisant deux fois successives par g_1, on a la division

$$\begin{aligned}
(\underline{X^2Y} + XY^2 + Y^2) + 0 \times (XY - 1) + 0 \times (Y^2 - 1) + 0 \\
&= (\underline{XY^2} + X + Y^2) + X \times (XY - 1) + 0 \times (Y^2 - 1) + 0 \\
&= (\underline{X} + Y^2 + Y) + (X + Y) \times (XY - 1) + 0 \times (Y^2 - 1) + 0 \\
&= (\underline{Y^2} + Y) + (X + Y) \times (XY - 1) + 0 \times (Y^2 - 1) + X \\
&= (\underline{Y} + 1) + (X + Y) \times (XY - 1) + 1 \times (Y^2 - 1) + X, \\
&= \underline{1} + (X + Y) \times (XY - 1) + 1 \times (Y^2 - 1) + (X + Y), \\
&= 0 + (X + Y) \times (XY - 1) + 1 \times (Y^2 - 1) + (X + Y + 1),
\end{aligned}$$

qui renvoie le résultat

$$q_1 = X + Y, \qquad q_2 = 1, \qquad r = X + Y + 1.$$

En réduisant d'abord une seule fois par g_1, puis une autre fois par g_2, on a la division

$$\begin{aligned}
(\underline{X^2Y} + XY^2 + Y^2) + 0 \times (XY - 1) + 0 \times (Y^2 - 1) + 0 \\
&= (\underline{XY^2} + X + Y^2) + X \times (XY - 1) + 0 \times (Y^2 - 1) + 0 \\
&= (\underline{2X} + Y^2) + X \times (XY - 1) + X \times (Y^2 - 1) + 0 \\
&= \underline{Y^2} + X \times (XY - 1) + X \times (Y^2 - 1) + 2X \\
&= \underline{1} + X \times (XY - 1) + (X + 1) \times (Y^2 - 1) + 2X, \\
&= 0 + X \times (XY - 1) + (X + 1) \times (Y^2 - 1) + (2X + 1),
\end{aligned}$$

qui renvoie le résultat

$$q_1 = X, \qquad q_2 = X + 1, \qquad r = 2X + 1.$$

Ni l'ordre des calculs, ni les quotients, ni les restes finaux ne sont identiques d'un calcul à l'autre.

6. Escaliers, définition et existence des bases de Gröbner

Les bases de Gröbner que nous allons définir sont des systèmes de générateurs d'idéaux de polynômes ayant de bonnes propriétés vis-à-vis de la réduction et de la division. Notamment, un premier point est de comprendre quels monômes peuvent être réduits à l'aide d'un polynôme donné. Ceci va être fait à l'aide de la notion de « partie stable » du monoïde des monômes, aussi appelés « escalier » en référence à sa représentation picturale. On verra qu'à tout système de générateurs d'un idéal est associé un escalier et qu'une base de Gröbner est essentiellement un système qui colle le mieux à l'escalier intrinsèque de l'idéal.

6.1. Parties stables du monoïde des monômes. Pour tout monoïde commutatif M, une *partie stable* S est un sous-ensemble de M clos par produit par tout élément de M. Cette définition est formellement très proche de celle d'un idéal, si ce n'est que l'ensemble de référence est maintenant un monoïde (muni d'une seule loi interne) et non un anneau (muni de deux lois internes) ; c'est pourquoi la notion de partie stable est aussi connue sous le vocable de « monoïdéal ».

De façon analogue aux idéaux définis par générateurs, étant donnée une famille $\{s_i\}_{i\in I}$ d'éléments de M, l'ensemble

$$S = \{\, ms_i \in M : m \in M,\ i \in I \,\} = \bigcup_{i\in I} Ms_i$$

est une partie stable du monoïde M, appelée la partie stable de M *engendrée par la famille de générateurs* s_i. Toute partie stable peut être vue comme engendrée par une famille de générateurs et encore une fois, la question est de comprendre si une partie stable peut être présentée comme engendrée par un nombre fini de générateurs.

Dans le cas du monoïde $[X, Y]$, on obtient une représentation en escaliers des parties stables de la façon suivante. Chaque monôme $m = X^aY^b$ est représenté par le point de coordonnées entières (a, b) de $\mathbb{N}^2$. Pour un monôme fixé $s = X^{a_0}Y^{b_0}$, la partie stable Ms engendrée par s est ainsi représentée par les points entiers (a, b) tels que $a \geq a_0$ et $b \geq b_0$, c'est-à-dire par un quadrant issu de (a_0, b_0). Une partie stable générale étant une union de parties stables de la forme Ms, elle est représentée par une union de quadrant de $\mathbb{N}^2$, dont les coins sont disposés le long d'une forme en escalier.

Rappelons encore une fois l'axiome fondamental à la base de la théorie des bases de Gröbner : le monôme de tête d'un produit, $\mathrm{mt}(fg)$, est le produit des monômes de tête $\mathrm{mt}(f)\,\mathrm{mt}(g)$. Il s'ensuit que la collection des monômes de tête des éléments non nuls de l'idéal est une partie stable : si un monôme s est dans cette partie, c'est qu'il existe un élément f de l'idéal de monôme de tête s ; pour tout monôme m, le polynôme mf est dans l'idéal et a sm pour monôme de tête. Pour un idéal I, nous noterons $\mathrm{mt}(I)$ la partie stable associée : $\mathrm{mt}(I) = \big\{\, \mathrm{mt}(p) : p \in I \setminus \{0\} \,\big\}$.

6.2. Définition des bases de Gröbner. Étant donnée une famille finie $\{g_i\}_{i\in\{1,\dots,s\}}$ de polynômes non nuls de A_n, deux parties stables jouent un rôle particulier. Tout d'abord, la partie stable $\mathrm{mt}(I)$ associée à l'idéal $I = \sum_{i=1}^{s} A_n g_i$, c'est-à-dire l'ensemble des monômes de tête de toutes les combinaisons non nulles $q_1g_1 + \cdots + q_sg_s$ pour des polynômes q_i. D'autre part, la partie stable constituée des monômes de tête de tous les polynômes réductibles par la famille $\{g_i\}_{i\in\{1,\dots,s\}}$, autrement dit, l'ensemble des monômes de tous des produits qg_i pour un polynôme q non nul.

Par construction, la première partie stable, $\mathrm{mt}(I)$, contient toujours la seconde, mais l'égalité n'est pas vérifiée sur tout système de générateurs d'un idéal donné. Considérons l'exemple suivant. On munit $A = \bar{\mathbb{Q}}[X, Y]$ de l'ordre monomial $\mathsf{lex}(Y, X)$. Les deux polynômes $\underline{XY^3} - 1$ et $\underline{X^3Y} + 1$ ne peuvent réduire que la partie stable

$$MXY^3 \cup MX^3Y,$$

alors que la partie stable associée à tout l'idéal I qu'ils engendrent est

$$\mathrm{mt}(I) = MY \cup MX^8.$$

Nous l'affirmons pour le moment sans pouvoir donner de preuve, mais on se convainc au moins de l'inclusion $\mathrm{mt}(I) \supseteq MY \cup MX^8$ quand on observe l'égalité

$$I = A(\underline{Y} + X^5) + A(\underline{X^8} + 1).$$

Ce phénomène motive la définition suivante.

Théorème-Définition. Soit I un idéal de $A_n = \bar{\mathbb{Q}}[X_1, \dots, X_n]$ et $\prec$ un ordre monomial sur A_n. Un sous-ensemble fini G de $I \setminus \{0\}$ est une *base de Gröbner de I pour l'ordre* $\prec$ si l'une quelconque des propriétés équivalentes est vérifiée :

(1) la partie stable de M engendrée par $\mathrm{mt}(G)$ est égale à $\mathrm{mt}(I)$;
(2) $\mathrm{mt}(G)$ et $\mathrm{mt}(I)$ engendrent le même idéal ;
(3) tout polynôme non nul f de I est réductible par G ;
(4) pour tout f dans A_n, il existe un unique polynôme r dans A_n dont aucun monôme ne soit divisible par un monôme de $\mathrm{mt}(G)$ et tel que $f - r$ soit dans l'idéal I ;
(5) pour tout f dans I, le reste de la division de f par G est nul.

Démonstration. Faisons une preuve (presque) circulaire.

$1 \Rightarrow 2$. Supposons que

$$\bigcup_{g\in G} M \operatorname{mt}(g) = \operatorname{mt}(I).$$

Passons alors aux idéaux. En notant (S) pour signifier l'idéal de A_n engendré par la famille $\{s\}_{s\in S}$, on a les égalités :

$$\big(\operatorname{mt}(I)\big) = \sum_{g\in G} \big(M \operatorname{mt}(g)\big) = \sum_{g\in G} \big(\operatorname{mt}(g)\big) = \big(\operatorname{mt}(G)\big).$$

On a prouvé que $\operatorname{mt}(G)$ et $\operatorname{mt}(I)$ engendrent le même idéal.

$2 \Rightarrow 3$. Supposons $\big(\operatorname{mt}(G)\big) = \big(\operatorname{mt}(I)\big)$. Soit $f \in I \setminus \{0\}$. On a d'abord l'égalité

$$\operatorname{mt}(f) = \sum_{g\in G} q_g \operatorname{mt}(g)$$

pour des polynômes q_g, puis, en scindant en monômes, l'égalité

$$\operatorname{mt}(f) = \sum_{j} c_j m_j \operatorname{mt}(g_j)$$

pour des c_j de $\bar{\mathbb{Q}}$, des monômes m_j et des g_j de G. Comme cette somme sur j est en fait une somme de termes, les termes en les monômes autres que $\operatorname{mt}(f)$ doivent s'annuler, et on peut sans perte de généralité supposer que pour chaque j, $m_j \operatorname{mt}(g_j) = \operatorname{mt}(f)$. On a alors, pour j_0 l'un de ces j, l'égalité

$$\operatorname{mt}(f) = m_{j_0} \operatorname{mt}(g_{j_0}),$$

et f est donc réductible par G.

$3 \Rightarrow 4$. Supposons que tout f non nul de I est réductible par G. Soit f un élément de A_n. On a l'existence énoncée au point (4) en prenant pour r le reste de la division de f par g : alors, $f - r$ est élément de I. Supposons que nous ayions deux écritures $f = h_i + r_i$, pour $i \in \{1, 2\}$, avec $h_i \in I$ et des r_i dont aucun des monômes n'est divisible par un monôme de $\operatorname{mt}(G)$. Alors, l'élément

$$r_1 - r_2 = h_2 - h_1 \in I$$

est soit nul, soit réductible. Supposons cette différence non nulle ; alors $\operatorname{mt}(r_1 - r_2)$ est forcément un monôme parmi ceux de r_1 et r_2. Ce monôme de tête est à la fois non divisible par un monôme de $\operatorname{mt}(G)$, par définition des r_i, et divisible par l'un d'entre eux, par l'hypothèse faite du point (3). De cette contradiction découle l'unicité de r.

$4 \Rightarrow 5$. Soit $f \in I$. En application du point (4), on trouve un r, qui par la preuve d'existence et d'unicité précédente ne peut être que le reste de la division de f par G. Comme $r = (r - f) + f$ est élément de I mais n'est pas réductible, c'est que r est nul.

$5 \Rightarrow 2$. Soit f réductible par G. Alors $\operatorname{mt}(f)$ est dans la partie stable engendrée par $\operatorname{mt}(G)$, donc dans l'idéal engendré par $\operatorname{mt}(G)$. Supposons le point (5). Comme tout élément non nul de I est alors réductible par G, on a obtenu dans ce cas l'inclusion

$$\big(\operatorname{mt}(I)\big) \subseteq \big(\operatorname{mt}(G)\big) \ ;$$

l'autre inclusion découle de $G \subseteq I$.

$2 \Rightarrow 1$. L'inclusion de la partie stable S engendrée par $\operatorname{mt}(G)$ dans $\operatorname{mt}(I)$ découle de ce que $G \subseteq I$. Pour l'autre inclusion, supposons le point (2) et soit $m \in \operatorname{mt}(I) \subseteq \big(\operatorname{mt}(I)\big) = \big(\operatorname{mt}(G)\big)$. Par le même raisonnement que pour l'implication $2 \Rightarrow 3$, on écrit m sous la forme $m_{j_0} \operatorname{mt}(g_{j_0})$, qui est un élément de S. La partie stable $\operatorname{mt}(I)$ est donc incluse dans S.

□

Remarquons que le polynôme r du point (4) du théorème précédent n'est autre que le reste de la division de f par G.

Sur un point de terminologie, notons que le terme de « base de Gröbner » est malheureux, puisqu'une base de Gröbner n'est pas une base, mais seulement un système de générateurs d'un idéal : une fois fixée une base de Gröbner $\{g_1, \ldots, g_s\}$ d'un idéal I donné, il n'y a en général pas unicité de l'écriture d'un élément de I comme combinaison des g_i, puisqu'il existe en général des combinaisons nulles des g_i.

Enfin, on peut toujours remplacer un élément d'une base de Gröbner par le reste de sa division par les autres éléments. Nous laissons la vérification de ce point au lecteur.

6.3. Lemme de Dickson et existence des bases de Gröbner. À ce stade, rien n'indique que des bases de Gröbner puissent exister pour tout idéal d'un anneau A_n et tout ordre monomial. La contrainte limitante est la finitude imposée par la définition : sans elle, il suffirait de prendre $I \setminus \{0\}$ comme système de générateurs de I. Cette existence repose sur la structure finie des parties stables, donnée par le lemme suivant. Encore une fois, la partie stable $\mathrm{mt}(I)$ est un bon invariant de I, à ordre monomial fixé.

Lemme (de Dickson). Toute partie stable S de $M = [X_1, \ldots, X_n]$ est finiment engendrée.

Démonstration. On fait une preuve par récurrence sur n.

Le cas $n = 1$ est immédiat : prendre l'élément de S de degré minimal ; celui-ci engendre S.

Supposons démontré le cas de $[X_1, \ldots, X_n]$ et soit S une partie stable de $M = [X_1, \ldots, X_n, Y]$. Considérons S', obtenue en effectuant la substitution $Y = 1$ dans S. C'est une partie stable de $M' = [X_1, \ldots, X_n]$; elle est donc finiment engendrée par des éléments X^{α_1}, …, X^{α_s}. Il en est de même pour chaque partie stable S'_j de M' donnée par $S'_j = \{\, m \in M' : mY^j \in S \,\}$: S'_j est finiment engendrée par des éléments $X^{\alpha_{1,j}}$, …, $X^{\alpha_{s_j,j}}$. Sans perte de généralité, on peut se donner un entier m tel que tous les $X^{\alpha_i}Y^m$ sont dans S. On vérifie aisément que la partie stable (finiment) engendrée par les $X^{\alpha_{i,j}}Y^j$ pour $j < m$ et par les $X^{\alpha_i}Y^m$, tels que définis ci-dessus, n'est autre que S. □

Nous pouvons maintenant énoncer le théorème d'existence des bases de Gröbner.

Corollaire. Pour tout ordre monomial $\prec$ sur $A_n = \bar{\mathbb{Q}}[X_1, \ldots, X_n]$, tout idéal I non nul de A_n admet une base de Gröbner.

Démonstration. Soit I un idéal non nul de A_n. Par le lemme de Dickson, il existe un système fini de générateurs de $\mathrm{mt}(I)$. Considérons un relèvement de ce système en un système d'éléments de I. Par la première définition des bases de Gröbner (par l'égalité des parties stables), celui-ci s'avère être une base de Gröbner de I pour $\prec$. □

7. Applications de la théorie des bases de Gröbner

7.1. Théorème de Hilbert et noethérianité des anneaux de polynômes. L'existence des bases de Gröbner donne la réponse constructive suivante à la question de la finitude de la présentation des idéaux polynomiaux.

Corollaire (Théorème de Hilbert). Tout idéal I de $A_n = \bar{\mathbb{Q}}[X_1, \ldots, X_n]$ admet un système fini de générateurs, ou, de façon équivalente, toute chaîne infinie croissante (pour l'inclusion) d'idéaux de A_n stationne.

Démonstration. Toute base de Gröbner est un système fini de générateurs, ce qui prouve le premier point. Pour l'équivalence annoncée, supposons d'abord que toute chaîne infinie croissante d'idéaux de A_n stationne. Étant donné un idéal I qui ne soit pas finiment engendré, nous pouvons trouver une suite infinie d'éléments dont chaque terme n'est pas dans l'idéal engendré par la sous-suite finie des termes précédents. On produit ainsi une suite infinie strictement croissante d'idéaux, ce qui contredit l'hypothèse. Ainsi, tout idéal est finiment engendré. Réciproquement, supposons que tout idéal soit finiment engendré et donnons-nous une chaîne infinie croissante d'idéaux. L'union de tous ces idéaux est un nouvel idéal, qui est donc finiment engendré. Soit

un système fini de générateur de l'union ; il existe un idéal de la chaîne qui contient tous ces générateurs. Cet idéal, de même que tous les suivants dans la chaîne, est égal à l'union. □

Nous donnons maintenant une autre preuve de ce résultat, laquelle est formellement très analogue à celle du lemme de Dickson.

Autre démonstration directe. On fait une preuve par récurrence sur n.

Le cas $n = 1$ est immédiat : prendre un élément de I de degré minimal ; celui-ci engendre I.

Supposons démontré le cas de A_n et soit I un idéal de $A_n[Y] = \bar{\mathbb{Q}}[X_1, \ldots, X_n, Y]$, qui n'est autre que A_{n+1} après avoir posé $Y = X_{n+1}$. Considérons I', obtenu comme l'ensemble des coefficients de plus haut degré en Y des éléments non nuls de I, auxquels on adjoint 0. C'est un idéal de A_n ; il est donc finiment engendré par des éléments α_1, …, α_s qui correspondent respectivement à des éléments $a_i = \alpha_i Y^{d_i} + \cdots$ de I. Il en est de même pour chaque idéal I'_j de A_n donné comme l'ensemble des coefficients de plus haut degré en Y des éléments de I de degré j en Y, auxquels on adjoint 0 : I'_j est finiment engendré par des éléments $\alpha_{1,j}$, …, $\alpha_{s_j,j}$ qui correspondent respectivement à des éléments $a_{i,j} = \alpha_{i,j} Y^j + \cdots$ de I. La suite des I'_j est une suite croissante d'idéaux de A_n, comme on le vérifie par une multiplication par Y, donc stationnaire par l'hypothèse de récurrence. De plus, chaque I'_j est inclus dans I'. On vérifie aisément que l'idéal (finiment) engendré par les $a_{i,j}$ pour $j < m$ et par les a_i, tels que définis ci-dessus, n'est autre que I, en vérifiant que ces éléments réduisent tout élément de I à zéro. □

La famille des $a_{i,j}$ et des a_i produite par la preuve qui précède est une base de Gröbner de I.

Plus généralement, on dit qu'un anneau A est *noethérien* lorsque tout idéal I de A admet un système fini de générateurs, ou, de façon équivalente, lorsque toute chaîne infinie croissante (pour l'inclusion) d'idéaux de A stationne.

7.2. Problème d'appartenance à un idéal. L'algorithme de division précédemment présenté retourne à la fois un reste et des quotients qui expriment un polynôme initial en terme d'une famille de diviseurs donnés. Dans bien des applications, les quotients explicites sont inutiles et seul le reste compte. Pour ce cas — et en fait pour l'algorithme de Buchberger qui sera vu ultérieurement —, on modifie l'algorithme de division en oubliant de traiter les quotients, ce qui aboutit à l'algorithme de réduction suivant. (La réduction est ici dite « totale » par opposition à la notion de réductibilité jusqu'alors présentée, qui ne concerne que le monôme de tête ; une autre terminologie parle de « réduction en tête » pour ce que nous avons appelé « réduction » et réserve « réduction » pour notre « réduction totale ».)

Algorithme (Réduction totale).

Entrée : un polynôme f et des polynômes non nuls g_1, …, g_s

Sortie : un polynôme r dont aucun monôme ne soit réductible par $\{g_i\}_{i\in\{1,\ldots,s\}}$ et pour lequel il existe des polynômes q_1, …, q_s tels que $f = q_1 g_1 + \cdots + q_s g_s + r$

(1) $r \leftarrow 0$

(2) tant que $f \neq 0$, faire
- si $\mathrm{mt}(g_i)$ divise $\mathrm{mt}(f)$ pour un certain i, choisir un tel i et réduire f par g_i, c'est-à-dire faire $f \leftarrow f - \mathrm{tt}(g_i)^{-1}\,\mathrm{tt}(p) g_i$
- sinon, faire $r \leftarrow r + \mathrm{tt}(f)$ et $f \leftarrow f - \mathrm{tt}(f)$

(3) renvoyer r

Lorsque G est une base de Gröbner, cette procédure est une procédure de mise sous forme canonique des classes de polynômes modulo $I = \sum_{i=1}^{s} A_n g_i$, les éléments du quotient A_n/I. En effet, une classe γ, donnée d'abord par un représentant f, se voit associer un nouveau représentant r unique (pour un choix fixé d'ordre monomial) avec la propriété que r est nul si et seulement si f est dans I, c'est-à-dire si et seulement si la classe γ est nulle.

En particulier, on peut donner un test algorithmique d'appartenance d'un polynôme f à un idéal donné présenté par un système fini de générateurs $g_1, \ldots, g_s$: il suffit de tester la nullité du reste de f par réduction, ce qui est résumé comme suit.

ALGORITHME (Test d'appartenance à un idéal polynomial).

ENTRÉE : un polynôme f et des polynômes non nuls $p_1, \ldots, p_r$ engendrant un idéal I

SORTIE : une valeur booléenne indiquant si f est élément de I

(1) choisir un ordre monomial $\prec$ sur $M = [X_1, \ldots, X_n]$

(2) calculer une base de Gröbner $G = \{g_1, \ldots, g_s\}$ de I pour cet ordre

(3) effectuer la réduction de p par G

(4) si le reste est nul, répondre VRAI, sinon répondre FAUX

7.3. Élimination et résolution de systèmes polynomiaux. La richesse de la théorie des bases de Gröbner provient de son lien avec l'« élimination polynomiale », c'est-à-dire, étant donnés des polynômes $p_1, \ldots, p_r$, avec le problème de la recherche d'une combinaison $f = q_1 p_1 + \cdots + q_r p_r$ des p_i pour des coefficients polynomiaux q_i telle que f ne fasse pas intervenir certaines indéterminées fixées à l'avance.

Ce problème a une reformulation en termes d'idéaux, puisqu'on montre aisément que l'intersection d'un idéal I de $A_n = \bar{\mathbb{Q}}[X_1, \ldots, X_n]$ avec le sous-anneau $A_{n,k} = \bar{\mathbb{Q}}[X_{k+1}, \ldots, X_n]$ est un idéal de $A_{n,k}$. Une question algorithmique naturelle est donc de rechercher une base de Gröbner de l'intersection à partir des p_i. Le résultat est le suivant.

THÉORÈME. Soit I un idéal de $A_n = \bar{\mathbb{Q}}[X_1, \ldots, X_n]$ et G une base de Gröbner de I pour l'ordre lexicographique sur les monômes de A_n. Soit encore k un entier entre 1 et $n-1$ inclus. Notons $A_{n,k}$ le sous-anneau $\bar{\mathbb{Q}}[X_{k+1}, \ldots, X_n]$ et $I_{n,k}$ l'idéal $I \cap A_{n,k}$. Alors, l'ensemble des éléments de G qui ne font intervenir aucun des X_i pour $i \leq k$ est une base de Gröbner de $I_{n,k}$ pour l'ordre monomial lexicographique induit sur $A_{n,k}$ par celui de A_n.

DÉMONSTRATION. Observons qu'appartenir à $A_{n,k}$ sans être nul est équivalent à avoir un monôme de tête en $X_{k+1}, \ldots, X_n$. La preuve découle ensuite de la dernière définition des bases de Gröbner (par la nullité des restes de la division par base de Gröbner). □

Géométriquement, $V(I \cap A_{n,k})$ est donné par l'image de $V(I)$ par la projection de $\bar{\mathbb{Q}}^n$ sur ses $n-k$ dernières composantes. Précisément, c'est la clôture pour la topologique de Zariski de cette image, c'est-à-dire le plus petit ensemble algébrique contenant cette image.

En vue des applications, intéressons-nous maintenant à la forme d'une base de Gröbner pour l'ordre lexicographique. En triant ses éléments par ordre monomial décroissant des monômes de

tête, on obtient un système de forme triangulaire, ou tout au moins de la forme échelonnée

$$
\begin{aligned}
g_1(x_1, x_2, \dots, x_n) &= 0,\\
&\vdots\\
g_{s_1}(x_1, x_2, \dots, x_n) &= 0,\\
g_{s_1+1}(\quad x_2, \dots, x_n) &= 0,\\
&\vdots\\
g_{s_2}(\quad x_2, \dots, x_n) &= 0,\\
&\vdots\\
g_{s_{n-1}+1}(\qquad x_n) &= 0,\\
&\vdots\\
g_{s_n}(\qquad x_n) &= 0,
\end{aligned}
\tag{7}
$$

où $s_k = r_1 + \cdots + r_k$ pour des entiers positifs r_ℓ (éventuellement nuls). Encore une fois, ce résultat provient de l'équivalence entre appartenir à $A_{n,k}$ sans être nul et avoir un monôme de tête en X_{k+1}, …, X_n.

Étant donné un algorithme pour la résolution d'une équation polynomiale d'une indéterminée en ses solutions complexes, on obtient ainsi l'algorithme suivant pour donner toutes les solutions dans $\bar{\mathbb{Q}}^n$ d'un système d'équations polynomiales lorsque celui-ci n'a que des solutions isolées.

Algorithme (Résolution d'un système polynomial à solutions toutes isolées).

Entrée : un système d'équations polynomiales $p_1(x_1, \dots, x_n) = \cdots = p_s(x_1, \dots, x_n) = 0$

Sortie : la famille des solutions dans $\bar{\mathbb{Q}}^n$ ou l'exception « existence de solutions non-isolées »

(1) Calculer une base de Gröbner de l'idéal engendré par les p_i pour l'ordre lexicographique et la mettre sous la forme échelonnée (7)

(2) Si r_n est nul, renvoyer « existence de solutions non-isolées »

(3) Considérer le p. g. c. d. des polynômes $g_{s_{n-1}+1}$, …, g_{s_n}

(4) Le résoudre en des racines α_1, …, α_t

(5) Pour i de 1 à n :

  (a) Évaluer la forme échelonnée en $x_n = \alpha_i$

  (b) Résoudre récursivement en x_1, …, x_{n-1}

  (c) En cas de solutions non-isolées, renvoyer « existence de solutions non-isolées »

(6) Renvoyer la collection des $(\gamma_1, \dots, \gamma_{n-1}, \alpha_i)$ pour chaque solution $(\gamma_1, \dots, \gamma_{n-1})$ obtenue à l'étape (b) lors de la i-ème itération de la boucle 5.

Remarquons que dans le cas où une exception est renvoyée, il serait possible de donner une preuve d'existence de solutions non-isolées.

À chaque spécialisation d'une indéterminée par un zéro d'un polynôme, on est amené à recalculer une base de Gröbner. Cet algorithme n'est donc pas un moyen économique pour résoudre.

7.4. Élimination et équations implicites. Redonnons explicitement le problème de la recherche d'équations implicites définissant un lieu géométrique donné par une paramétrisation. Étant donnée une paramétrisation rationnelle

$$x_i = r_i(t_1, \ldots, t_m), \quad i = 1, \ldots, n,$$

d'un ensemble de points de $\bar{\mathbb{Q}}^n$, il s'agit de trouver un système d'équations polynomiales qui définisse le plus petit ensemble algébrique qui le contienne.

La méthode consiste à éliminer les indéterminées T_i de l'écriture polynomiale (sans fractions) des équations tout en évitant les pôles des fractions rationnelles r_i. On notera particulièrement dans l'algorithme qui suit l'introduction d'une nouvelle indéterminée U dont le rôle est d'interdire l'annulation des dénominateurs des r_i.

ALGORITHME (Mise sous forme implicite d'une paramétrisation).

ENTRÉE : les fractions $r_i = p_i/q_i$, en les $T_1, \ldots, T_m$, pour $1 \leq i \leq n$

SORTIE : système d'équations algébriques implicites décrivant le plus petit ensemble algébrique contenant l'image de la paramétrisation

(1) Choisir un ordre monomial $\prec$ sur $M = [X_1, \ldots, X_n, T_1, \ldots, T_m, U]$ qui élimine U et les T_i (par exemple, $\mathsf{lex}(U, T_1, \ldots, T_m, X_1, \ldots, X_n)$)

(2) Introduire le p. p. c. m. $\tilde{q}$ des q_i

(3) Calculer pour cet ordre une base de Gröbner de l'idéal engendré par les polynômes $q_i(T_1, \ldots, T_m)X_i - p_i(T_1, \ldots, T_m)$ et par $U\tilde{q}(T_1, \ldots, T_m) - 1$

(4) En retirer les polynômes qui font intervenir U ou l'un des T_i et renvoyer la famille ainsi obtenue

---

Donnons un exemple montrant la nécessité de la variable ajoutée U et explicitant ainsi mieux son rôle. Considérons la nappe paramétrique donnée par la paramétrisation

$$x = \frac{s^2}{t}, \qquad y = \frac{t^2}{s}, \qquad z = s.$$

Un calcul sans introduire U et le polynôme $STU - 1$ renvoie le polynôme $Z(X^2Y - Z^3)$, dont le lieu géométrique des zéros est l'union de l'hypersurface $\mathscr{S}$ d'équation $x^2y = z^3$ et de l'hyperplan $\mathscr{H}$ d'équation $z = 0$. Cependant, pour qu'une solution soit sur l'hyperplan $\mathscr{H}$, il est nécessaire que le paramètre s soit nul, donc ne permette pas de définir une valeur de y. La nappe paramétrique est donc tracée toute entière sur $\mathscr{S}$ et le polynôme obtenu, $Z(X^2Y - Z^3)$, n'est pas minimal. En reprenant le calcul en ajoutant le polynôme $STU - 1$, on obtient le seul facteur $X^2Y - Z^3$ qui ne décrit que l'hypersurface $\mathscr{S}$. Bien que les deux droites données respectivement par $x = 0$, $z = 0$ et par $y = 0$, $z = 0$ soient dans cette hypersurface sans être sur la nappe, il n'est pas possible de donner une équation algébrique valide pour toute la nappe mais qui exclue ces deux droites : le polynôme $X^2Y - Z^3$ est irréductible dans $\bar{\mathbb{Q}}[X, Y, Z]$ puisqu'il est de degré 1 en Y.

CHAPITRE 6

Algorithme de Buchberger

1. Saturation des escaliers et algorithme naïf

On l'a vu, la question des bases de Gröbner revient à faire coïncider deux escaliers : l'un associé à l'idéal, unique une fois un ordre monomial choisi, un autre qui dépend du système de générateurs considéré, qui correspond en général à une partie stable plus petite que celle associée à l'idéal. Le calcul d'une base de Gröbner va reposer sur une technique de saturation visant à accroître la partie stable donnée par les générateurs, accroissement qui correspondant à faire un changement de système de générateurs. L'outil qui va permettre cette saturation est appelé « S-polynôme ». Le calcul d'une base de Gröbner par l'algorithme de Buchberger se résumera ensuite essentiellement à une itération tant qu'il sera possible de produire de nouveaux S-polynômes.

1.1. S-polynômes et relation avec les bases de Gröbner. Un exemple simple motive la définition qui va suivre : soit I l'idéal $A_1(X-1)+A_1X$ dans l'anneau A_1 muni de l'ordre monomial défini par les puissances croissantes L'escalier associés aux générateurs $X-1$ et X est réduit à la partie stable des puissances de X à exposant strictement positifs puisque les deux générateurs ont X comme monôme de tête. Pourtant, le polynôme $1\times(X-1)+(-1)\times X$ vaut 1 et la partie stable associée à l'idéal est ainsi l'ensemble de tous les monômes en X. De manière générale, le phénomène est qu'un polynôme p peut très bien être dans un idéal $I=\sum_{i=1}^{r}A_np_i$ sans que son monôme de tête $\mathrm{mt}(p)$ ne soit dans la partie stable $\bigcup_{i=1}^{r}M\,\mathrm{mt}(p_i)$, ce qui a lieu lorsqu'une combinaison $\sum_{i=1}^{r}l_ip_i$ produit une annulation des termes de tête des l_ip_i.

Définition (S-polynômes). Soient deux polynômes non nuls p_1 et p_2 et posons $m_1=\mathrm{mt}(p_1)$, $m_2=\mathrm{mt}(p_2)$ et $m=\mathrm{ppcm}(m_1,m_2)=n_1m_1=n_2m_2$. On appelle *$S$-polynôme* des deux polynômes p_1 et p_2 toute combinaison linéaire non nulle de la forme $l_1p_1+l_2p_2$ pour des polynômes non nuls l_1 et l_2 tels que $\mathrm{mt}(l_i)=n_i$ et $\mathrm{tt}(l_1)\,\mathrm{tt}(p_1)+\mathrm{tt}(l_2)\,\mathrm{tt}(p_2)=0$. En pratique, on se restreint à des termes et on pose :

$$\mathrm{Spoly}(p_1,p_2)=l_1p_1+l_2p_2\quad\text{pour}\quad l_1=\mathrm{ct}(p_2)n_1,\quad l_2=-\,\mathrm{ct}(p_1)n_2.$$

Il convient maintenant de faire le lien entre bases de Gröbner et S-polynômes. Comme les S-polynômes sont éléments de l'idéal considéré, ils se réduisent nécessairement à zéro par toute base de Gröbner de l'idéal. À l'inverse, étant donné un système de générateurs $P=\{p_k\}_{1\leq k\leq r}$ d'un idéal dont les S-polynômes des générateurs pris deux à deux ne se réduisent pas tous à zéro par P, alors, après adjonction à P des restes non nul des divisions correspondantes, on aboutit à un nouveau système de générateurs du même idéal qui par construction réduit à zéro les S-polynômes initiaux. La section qui suit montre qu'on aboutit à une base de Gröbner en répétant cette opération un nombre fini de fois. Au préalable, nous donnons une nouvelle caractérisation des bases de Gröbner, en termes de S-polynômes.

Théorème (Propriété caractéristique des bases de Gröbner). Soit $P=\{p_k\}_{1\leq k\leq r}$ un système de générateurs non nuls d'un idéal de polynômes. Tous les S-polynômes $\mathrm{Spoly}(p_i,p_j)$, associés aux paires de générateurs pour $1\leq i<j\leq r$, se réduisent à 0 par P si et seulement si P est une base de Gröbner de l'idéal.

Démonstration. On montre l'implication directe en montrant que tout élément de l'idéal se réduit à zéro par P ; l'implication inverse a été montrée précédemment. Soit p irréductible par P et exprimé sous la forme $\sum_{i=1}^{r}l_ip_i$. Sans perte de généralité, on peut supposer que le monôme

$\delta = \max_{1\le i\le r}\{\mathrm{mt}(l_ip_i)\}$ est minimal parmi les écritures de p en termes des p_i et que pour un entier k bien choisi, on a la relation $\delta = \mathrm{mt}(l_ip_i) \succ \mathrm{mt}(l_jp_j)$ dès que $1 \le i \le k < j \le r$. Alors,

$$p = \sum_{i=1}^{k} \mathrm{tt}(l_i)p_i + \sum_{i=1}^{k} \big(l_i - \mathrm{tt}(l_i)\big)p_i + \sum_{i=k+1}^{r} l_ip_i = \sum_{i=1}^{k} \mathrm{tt}(l_i)p_i + \sum_{i=1}^{r} l'_ip_i,$$

où dans la dernière somme, on a $\mathrm{mt}(l'_ip_i) \prec \delta$ dès lors que le polynôme l'_i est non nul. Sans plus de perte de généralité, on peut encore supposer que k est minimal parmi les écritures de p sous cette forme qui minimisent δ. Notons que k vaut au moins 2, sinon δ serait monôme de tête de p et p serait réductible par p_1. Observons que δ est divisible par le p. p. c. m. des monômes de tête de p_1 et p_2. On introduit donc les monômes n_1 et n_2 qui interviennent dans la définition de $\mathrm{Spoly}(p_1, p_2)$, ainsi que des constantes λ_1 et λ_2 de $\bar{\mathbb{Q}}$ et le monôme $m = \delta/\operatorname{ppcm}\big(\mathrm{mt}(p_1), \mathrm{mt}(p_2)\big)$, pour obtenir :

$$\mathrm{tt}(l_1)p_1 + \mathrm{tt}(l_2)p_2 = \lambda_1 m\,\mathrm{ct}(p_2)n_1p_1 + \lambda_2 m\,\mathrm{ct}(p_1)n_2p_2 = \lambda_1 m\,\mathrm{Spoly}(p_1,p_2) + (\lambda_1+\lambda_2)\,\mathrm{ct}(p_1)mn_2p_2.$$

Par construction, le premier polynôme de cette dernière somme a son monôme de tête strictement plus petit que δ alors que le monôme de tête du second polynôme est exactement δ, à moins qu'il ne soit nul. Par hypothèse, le S-polynôme de p_1 et de p_2 se réduit à zéro par P ; le premier terme $\lambda_1 m\,\mathrm{Spoly}(p_1, p_2)$ se récrit donc comme une somme $\lambda_1 m \sum_i l''_ip_i = \sum_i \lambda_1 m l''_ip_i$ dans laquelle chaque terme a son monôme de tête strictement plus petit que δ. On obtient ainsi une contradiction à la minimalité de k, d'où la conclusion que tout élément $\sum_{i=1}^{r} l_ip_i$ de l'idéal est réductible par P. Cette famille est donc une base de Gröbner de l'idéal. □

1.2. Version rudimentaire de l'algorithme de Buchberger. Nous donnons dans cette section un premier algorithme pour le calcul d'une base de Gröbner d'un idéal donné par des générateurs, algorithme qui découle immédiatement du théorème caractéristique précédent.

Algorithme (Algorithme rudimentaire de calcul d'une base de Gröbner).

Entrée : un ensemble fini P de polynômes p_i non nuls et un ordre monomial $\preceq$

Output : une base de Gröbner G pour le même idéal

(1) Initialiser G à P et S à l'ensemble des paires d'éléments de G
(2) Tant que S n'est pas vide,
  (a) Choisir une paire $p = \{g, g'\}$ et la retirer de S
  (b) Calculer $\mathrm{Spoly}(g, g')$ et le réduire par G
  (c) Si le reste r est non nul, alors
    (i) Adjoindre à S toutes les paires $\{g, r\}$ pour $g \in G$
    (ii) Adjoindre r à G
(3) Renvoyer G

Preuve de l'algorithme. Un invariant de cet algorithme est que l'ensemble G ne contient que les générateurs p_i initiaux et des recombinaisons finies de ceux-ci à coefficients polynomiaux : l'idéal engendré par G est donc constant. De plus, si l'algorithme termine, la sortie G réduit à zéro chacun des S-polynômes de ses éléments pris deux à deux. Le théorème précédent fournit donc la correction de l'algorithme.

Pour la terminaison, on remarque que la partie stable engendrée par les monômes de tête des éléments de G croît strictement à chaque adjonction dans G. En considérant les idéaux engendrés successivement par cette partie stable, on obtient ainsi une suite strictement croissante d'idéaux, puisque l'idéal engendré par une partie stable admet en tant qu'espace vectoriel sur $\bar{\mathbb{Q}}$ la base constituée exactement des monômes de la partie stable. Par noethérianité de A_n, cette suite d'idéaux ne peut être infinie et il ne peut donc y avoir qu'un nombre fini d'adjonctions dans G. □

2. Réductions à zéro et algorithme classique

On l'observe sur des implantations, une grande partie du temps passé par l'algorithme de la section qui précède (et même sur ses optimisations dont on va parler) est passé dans la réduction des S-polynômes par la base de Gröbner en construction. Par ailleurs, la preuve de terminaison qui vient d'être faite indique que, nécessairement, à partir d'un certain stade de l'exécution, tous les S-polynômes se réduisent à zéro. Il apparaît donc comme important de savoir prédire quelles réductions doivent aller à zéro afin d'éviter autant que possible les calculs correspondants.

2.1. Paires triviales et paires inutiles. Dans l'approche de Buchberger, on identifie deux causes différentes de réduction à zéro, en liaison avec des propriétés différentes des paires de polynômes dont on réduit le S-polynôme.

Paires triviales. D'abord, les *paires triviales* sont des paires de polynômes qui réduisent leur propre S-polynôme à zéro indépendamment des autres polynômes de la base de Gröbner en construction. Dit autrement, à eux deux ils forment une base de Gröbner de l'idéal qu'ils engendrent. Il est intéressant de pouvoir identifier une telle paire sans calcul. Une condition suffisante pour qu'une paire soit triviale est que les monômes de tête des deux polynômes soient premiers entre eux, autrement dit, que les indéterminées qui apparaissent dans l'un n'apparaissent pas dans l'autre.

Lemme. Si $\operatorname{ppcm}\big(\operatorname{mt}(p),\operatorname{mt}(p')\big) = \operatorname{mt}(p)\operatorname{mt}(p')$, alors $\operatorname{Spoly}(p,p')$ se réduit à zéro par $\{p,p'\}$.

Démonstration. Écrivons $p = t_1 + \cdots + t_r$ et $p' = t'_1 + \cdots + t'_s$ pour deux suites de termes t_i et t'_i, chacunes en des monômes qui décroissent strictement avec i. Le S-polynôme $\operatorname{Spoly}(p,p')$ vaut alors $t'_1 p - t_1 p' = p t'_1 - t_1 p'$, soit

$$\begin{pmatrix} t_1 & t_2 & \dots & t_r \end{pmatrix} \begin{pmatrix} 0 & -1 & \dots & -1 \\ 1 & 0 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 1 & 0 & \dots & 0 \end{pmatrix} \begin{pmatrix} t'_1 \\ t'_2 \\ \vdots \\ t'_s \end{pmatrix}.$$

On va montrer par récurrence que la réduction du S-polynôme va passer par des polynômes q donnés sous la forme

$$\begin{pmatrix} t_1 & \dots & t_r \end{pmatrix} \begin{pmatrix} 0 & -U \\ V & 0 \end{pmatrix} \begin{pmatrix} t'_1 \\ \vdots \\ t'_s \end{pmatrix}$$

pour des matrices rectangulaires U et V dont toutes les entrées valent 1. Le S-polynôme est évidemment de cette forme. Considérons une étape de la réduction. Soit à réduire le polynôme q donné par la forme ci-dessus par p et p'. Tout d'abord, la contribution à q provenant de la matrice U, respectivement V, a pour monôme de tête celui correspondant au coin en haut à gauche dans U, respectivement V. Cette contribution s'écrit $-t_1 t'_i$ avec $i > 1$, respectivement $t_j t'_1$ avec $j > 1$. Les deux termes correspondant ne peuvent se compenser et s'annuler, car $t_1 t'_i$ ne peut valoir $t_j t'_1$: s'il y avait égalité, puisque $\operatorname{mt}(t_1) = \operatorname{mt}(p)$ et $\operatorname{mt}(t'_1) = \operatorname{mt}(p')$ sont étrangers, il faudrait $i = j = 1$, ce qui est impossible. Si le monôme de tête de q est donné par $-t_1 t'_i$, la réduction consiste à ajouter $p_1 t'_i$, ce qui rétrécit U d'une colonne et élargit V d'une colonne ; s'il est donné par $t_j t'_1$, la réduction consiste à retrancher $t_1 p'$, ce qui élargit U d'une ligne et rétrécit V d'une ligne. Après $r + s$ étapes de réduction, q est réduit à zéro. □

On notera l'importance dans cette preuve de la relation $t'_1 p - t_1 p' = p t'_1 - t_1 p'$, qui fait que le résultat ne se transpose pas à des théories de bases de Gröbner dans des contextes non commutatifs.

Paires inutiles. Une autre forme de paires qui se réduisent à zéro, les *paires inutiles*, correspond à des calculs redondants au sein de l'algorithme de Buchberger : la réduction à zéro envisagée n'a lieu que grâce au contexte des autres paires déjà réduites et incorporées à la base de Gröbner en construction.

Lemme. Si $\operatorname{mt}(g_k)$ divise $\operatorname{ppcm}\big(\operatorname{mt}(g_i),\operatorname{mt}(g_j)\big)$ et si les paires $\{g_i,g_k\}$ et $\{g_j,g_k\}$ ont déjà été réduites et les restes correspondants introduits, alors le S-polynôme de $\{g_i,g_j\}$ se réduit à zéro.

2.2. Forme canonique pour les idéaux de polynômes. Quelques propriétes générales des bases de Gröbner vont nous permettre de rendre unique la base de Gröbner associée à un idéal donné pour un ordre monomial choisi.

Tout d'abord, on a déjà indiqué qu'on peut toujours remplacer un élément d'une base de Gröbner par le reste de sa division par les autres éléments.

Par ailleurs, on peut toujours supprimer un élément p d'une base de Gröbner dont le monôme de tête est divisible par celui d'un autre élément q. En effet, la première étape de la réduction du polynôme p par le polynôme q est en fait le calcul du S-polynôme $\mathrm{Spoly}(p, q)$, lequel se réduit à zéro dans une base de Gröbner, mais sans plus pouvoir se réduire par p puisque tous les polynômes considérés à partir du S-polynôme ont un monôme de tête strictement plus petit que p. En d'autres termes, p se réduit à zéro par division par les autres éléments de la base de Gröbner.

Les propriétés précédentes motivent la définition qui suit.

Définition (Base de Gröbner réduite). Une *base de Gröbner réduite* est une base de Gröbner telle qu'aucun monôme appaissant dans l'un quelconque de ses éléments ne soit réductible par le reste de la base de Gröbner, et dont les coefficients de tête sont normalisés à 1.

En termes de la partie stable de l'idéal, exhiber une base de Gröbner réduite revient à se donner un polynôme par coin de l'escalier saillant vers les monômes de plus petits degrés, à calculer le reste de chaque polynôme par les autres, puis à normaliser les coefficients de tête à 1. Ainsi, à ordre monomial donné, tout idéal admet une unique base de Gröbner réduite. En effet, le polynôme d'une base de Gröbner réduite correspondant à un coin donné de l'escalier de l'idéal ne peut alors qu'être unique, puisque son terme de tête est un monôme et que ses autres termes correspondent à des monômes sous l'escalier : si deux tels polynômes proviennent du même coin, par différence on obtient un polynôme de l'idéal qui n'a que des monômes sous l'escalier et est donc nul.

2.3. Algorithme de Buchberger et ses stratégies classiques. Les critères des deux sections précédentes pour identifier à l'avance une réduction à zéro et pour l'unicité d'une base de Gröbner fournissent l'optimisation qui suit de l'algorithme naïf déjà donné pour le calcul de bases de Gröbner.

Algorithme (Algorithme de Buchberger).

Entrée : un ensemble fini P de polynômes p_i et un ordre monomial $\preceq$

Sortie : une base de Gröbner réduite G pour le même idéal

(1) Initialiser G à P et S à l'ensemble des paires d'éléments de G
(2) Tant que S n'est pas vide,
  (a) Choisir une paire $p = \{g, g'\}$ et la retirer de S
  (b) Si p est inutile ou triviale, passer à la paire suivante
  (c) Calculer $\mathrm{Spoly}(g, g')$ et le réduire par G
  (d) Si le reste r est non nul, alors
    (i) Adjoindre à S tous les paires $\{g, r\}$ pour $g \in G$
    (ii) Retirer de G les polynômes dont le monôme de tête est divisible par celui de r et y adjoindre r
(3) Inter-réduire G et renvoyer le résultat

Preuve de l'algorithme. La terminaison de l'algorithme se fait comme pour la version naïve de l'algorithme ; la correction s'appuie en plus sur les critères de réduction à zéro et de normalisation des deux sections précédentes. □

Donnons un exemple de calcul en recherchant une base de Gröbner pour l'ordre lexicographique de l'idéal engendré par $p_1 = \underline{X^2} - Y$ et $p_2 = \underline{X^3} - Z$. Le premier S-polynôme à considérer

est $\mathrm{Spoly}(p_1, p_2) = Xp_1 - p_2 = -\underline{XY} + Z$, qui est irréductible par $\{p_1, p_2\}$. Nous posons $p_3 = \underline{XY} - Z$. Puisque X^2 divise le p. p. c. m. X^3Y de X^3 et de XY et que la paire (p_1, p_2) a déjà été traitée, traiter la paire (p_1, p_3) rend la paire (p_2, p_3) inutile. Le S-polynôme $\mathrm{Spoly}(p_1, p_3)$ est $Yp_1 - Xp_3$ et vaut $\underline{XZ} - Y^2$, qui est irréductible et que nous baptisons p_4. De la même façon, la paire (p_1, p_4) rend la paire (p_2, p_4) inutile. Restent donc à traiter les deux paires (p_1, p_4) et (p_3, p_4). Le S-polynôme $\mathrm{Spoly}(p_1, p_4)$ vaut $Zp_1 - Xp_4 = \underline{XY^2} - YZ = Yp_3$ et se réduit donc à zéro par p_3. Le S-polynôme $\mathrm{Spoly}(p_3, p_4)$ vaut $Zp_3 - Yp_4 = \underline{Y^3} - Z^2$, lequel est irréductible par $\{p_1, \ldots, p_4\}$ et que nous baptisons p_5. Maintenant, par le critère sur les paires triviales, nous ne retenons que le S-polynôme $\mathrm{Spoly}(p_3, p_5) = Y^2p_3 - Xp_5 = XZ^2 - Y^2Z = Zp_4$ et se réduit donc à zéro par p_4. La base de Gröbner calculée est donc $\{p_1, \ldots, p_5\}$, et, comme p_2 se réduit à zéro par p_1 et p_3, la base de Gröbner réduite est $\{p_1, p_3, p_4, p_5\}$, donnant la partie stable engendrée par X^2, XY, XZ, Y^3.

Deux sources d'indéterminismes restent présentes dans l'algorithme de Buchberger tel qu'il a été décrit : d'une part, il n'est pas dit comment une paire doit être choisie parmi les paires restant dans S ; d'autre part, la procédure de réduction que nous avons donnée, de même que celle de division, ne précise pas comment choisir le polynôme servant à chaque étape de division dans les cas où il y a ambiguïté. Si des choix quelconques assurent la correction et la terminaison de l'algorithme, il s'avère que ces choix ont un impact très fort sur le temps d'exécution de l'algorithme. On observe que les temps de calcul sont meilleurs pour certaines stratégies de choix, mais en général il y a peu de résultats formellement établis. Par ailleurs, une approche qui semble être le plus souvent meilleure que l'algorithme de Buchberger est celle de l'algorithme F5 de Faugère, bien que celui-ci ne fasse une hypothèse sur son entrée. Il ne nous est pas possible d'exposer dans ce cours introductif cette variation de l'algorithme de Buchberger, et nous retiendrons l'existence des quelques stratégies suivantes.

Stratégie normale. La stratégie dite « normale » est due à Buchberger. Elle réduit en priorité par les polynômes les plus anciennement introduits dans la base de Gröbner en construction, au motif que ceux-ci sont souvent de plus petite taille que les polynômes récemment introduits. De plus, les S-polynômes sont traités dans l'ordre croissant des degrés des p. p. c. m. des monômes de tête des polynômes de la paire correspondante.

Stratégie du sucre. La stratégie dite « du sucre » est due à Giovini, Mora, Niesi, Robbiano et Traverso. Elle vise dans le cas lexicographique, d'ordinaire le plus coûteux, à simuler un calcul sur des polynômes homogènes, pour lequel une optimisation est possible. À cette fin, on décore les polynômes d'un degré fantôme qui est le degré homogène qu'aurait le polynôme si le calcul avait été fait sur les polynômes homogénéisés et on procède par degrés fantômes croissants. Le degré fantôme n'est autre que le degré total sur les polynômes initiaux du calcul, mais il peut devenir plus grand que lui en cours de calcul.

Stratégie de Gebauer et Möller. Une stratégie due à Gebauer et Möller tente d'exploiter au mieux les critères de rejet des paires triviales et inutiles. En particulier, le critère sur les paires inutiles permet souvent de rejeter l'une parmi deux paires. Plutôt que de choisir au hasard, on s'efforce de ne pas rejeter une paire qui pourrait aussi être rejetée pour une autre raison (par exemple parce qu'elle est reconnue comme triviale), ce qui permet de rejeter deux paires sans calcul au lieu d'une seule. Cette stratégie cherche aussi à maintenir la base en construction aussi petite que possible pendant tout le calcul, c'est-à-dire que la phase d'inter-réduction n'a pas lieu seulement à la fin du calcul, mais tout au long de celui-ci.

Changement d'ordres. Dans les logiciels modernes, on dispose en général d'une implantation de l'algorithme de Buchberger optimisée pour l'ordre grevlex en utilisant notamment la stratégie de Gebauer et Möller raffinée par la stratégie normale. Pour les ordres monomiaux « distants » de l'ordre grevlex, en particulier pour l'ordre lex, on préfère utiliser un algorithme de changement d'ordre, c'est-à-dire un algorithme qui calcule une base de Gröbner pour l'ordre cible par de l'algèbre linéaire à partir de la base de Gröbner pour l'ordre initial. Les algorithmes de changement d'ordre les plus employés sont l'algorithme FGLM et l'algorithme de marche de Gröbner.

3. Cas particulier et extensions de l'algorithme de Buchberger

Pour des calculs en une seule indéterminée, l'algorithme de Buchberger se spécialise en une version sans optimisation de l'algorithme d'Euclide. Pour des entrées linéaires en toutes les indéterminées, il se spécialise en une version de l'algorithme de Gauss à pivot partiel (pivot de colonne), et pour un ordre de traitement des colonnes induit par l'ordre monomial. Dans ce dernier calcul et pour l'ordre lexicographique, certains S-polynômes se réduisent trivialement à zéro, à savoir ceux qui correspondent à des paires de lignes de la matrice dont les termes non nuls les plus à gauche ne sont pas sur la même colonne. Cette remarque est à la base d'une généralisation de l'algorithme de Buchberger aux bases de Gröbner de modules. Quoiqu'un peu plus technique, cette généralisation ouvre la voie à bien de nouvelles applications : l'expression des éléments de la base de Gröbner en terme des polynômes initiaux, thème d'une prochaine section ; le calcul d'inverses modulo un idéal (pas nécessairement principal) par une généralisation du calcul de relations de Bézout ; la saturation d'un idéal par un polynôme (pour un polynôme p donné, on adjoint à un idéal tous les polynômes f tel que pf soit dans l'idéal de départ) ; d'autres calculs en algèbre homologique. Nous détaillons ci-dessous quelques-uns de ces points.

3.1. Cas particulier de l'algorithme d'Euclide. Pour deux polynômes $p_1 = c_1 X^{d_1} + \cdots$ et $p_2 = c_2 X^{d_2} + \cdots$ de $\bar{\mathbb{Q}}[X]$, avec $\deg p_1 = d_1 > d_2 = \deg p_2$,

$$\operatorname{Spoly}(p_1, p_2) = c_2 p_1 - c_1 X^{d_1 - d_2} p_2$$

est la première étape élémentaire d'une division euclidienne. Les étapes suivantes de la division reproduisent ensuite les mêmes calculs que la réduction de $\operatorname{Spoly}(p_1, p_2)$ par $\{p_2\}$. Ainsi, la réduction de $\operatorname{Spoly}(p_1, p_2)$ par $\{p_2\}$ fournit le reste p_3 de la division euclidienne de p_1 par p_2 (à multiplication par une constante près).

Suivons de près le calcul de l'algorithme de Buchberger sur une entrée constituée des deux polynômes p_1 et p_2. Ce calcul détermine d'abord le reste p_3, puis il crée les paires $\{p_1, p_3\}$ et $\{p_2, p_3\}$. La première devient redondante si on traite la seconde. L'algorithme de Buchberger calcule donc une suite de restes successifs, jusqu'à obtenir un reste nul : il simule ainsi l'algorithme d'Euclide et renvoie finalement le p. g. c. d. des polynômes initiaux p_1 et p_2.

Ce calcul se généralise à une d'entrée p_1, …, p_n constituée de plus de deux polynômes : l'algorithme de Buchberger calcule encore le p. g. c. d. de ses entrées, mais en effectuant les réductions dans un ordre différent de celui qui serait suivi en calculant le p. g. c. d. g_1 de p_1 et p_2, puis le p. g. c. d. g_2 de g_1 et p_3, et ainsi de suite jusqu'au p. g. c. d. g_{n-1} de g_{n-2} et p_n.

3.2. Cas particulier de l'algorithme de Gauss. Considérons le système linéaire

$$a_{i,1} x_1 + \cdots + a_{i,n} x_n = 0, \qquad 1 \le i \le m.$$

La réduction de Gauss de ce système renvoie un système triangulaire équivalent, de la forme

$$\begin{aligned} b_{1,1} x_1 + \cdots + b_{1,r} x_r + \cdots + b_{1,n} x_n &= 0, \\ &\cdots \\ b_{r,r} x_r + \cdots + b_{r,n} x_n &= 0 \end{aligned}$$

pour des $b_{i,i}$ non nuls.

Ici, « équivalent » signifie qu'il existe des matrices U et V donnant les relations $A = UB$ et $B = VA$ entre les matrices $A = (a_{i,j})_{1 \le i \le m,\, 1 \le j \le n}$ et $B = (b_{i,j})_{1 \le i \le r,\, 1 \le j \le n}$. Ainsi, les deux systèmes de polynômes $\{a_{i,1} X_1 + \cdots + a_{i,n} X_n\}_{1 \le i \le m}$ et $\{b_{i,i} X_i + \cdots + b_{i,n} X_n\}_{1 \le i \le r}$ engendrent le même espace vectoriel sur $\bar{\mathbb{Q}}$, donc le même idéal de A_n.

Vu leur structure en monômes, les polynômes $b_{i,i} X_i + \cdots + b_{i,n} X_n$ constituent pour l'ordre lexicographique une base de Gröbner de l'idéal engendré par les $a_{i,1} X_1 + \cdots + a_{i,n} X_n$. Une base de Gröbner réduite fournit un système linéaire triangulaire réduit : la matrice $(b_{i,j})_{1 \le i,j \le r}$ est diagonale.

Comparons les réductions par l'algorithme de Gauss et dans le cas linéaire de l'algorithme de Buchberger. L'algorithme de Gauss n'est amené à réduire une ligne $U = u_i x_i + \cdots + u_n x_n$ par une autre ligne $V = v_j x_j + \cdots + v_n x_n$ que dans le cas où le scalaire v_j est non nul et sert de

pivot et si $i \geq j$. Sans perte de généralité, nous pouvons aussi supposer que u_i est non nul, quitte à incrémenter i. Si i est alors strictement plus grand que j, la réduction ne fait en réalité aucun calcul. Dans le cas contraire, la ligne U est remplacée par la ligne $W = U - (u_j/v_j)V$, qui est de la forme $W = w_k x_k + \cdots + w_n x_n$ avec $k > j$. Par le calcul analogue par algorithme de Buchberger, le S-polynôme des polynômes U et V est trivial lorsque i et j sont différents, alors que s'ils sont égaux, le calcul de S-polynôme correspond en fait à la réduction de U par V dans l'algorithme de Gauss. Pour un calcul par l'algorithme de Buchberger sur des entrées linéaires, on peut donc adapter les critères de rejet de paires en définissant à zéro les S-polynômes de polynômes possédant des indéterminées de tête différentes.

3.3. Bases de Gröbner de modules. La théorie des bases de Gröbner se généralise pour donner de bons systèmes de générateurs de modules sur A_n, c'est-à-dire pour faire de l'algèbre linéaire à coefficients dans un anneau de polynômes au lieu d'un corps. Rappelons qu'un *module* M sur un anneau A est un groupe additif stable par multiplication par tout élément de A. Par exemple, tout idéal de A est un A-module, de même que les quotients de la forme A/I pour un idéal I de A sont des modules sur A. Plus généralement, étant donnée une famille $(g_u)_{u\in U}$ d'éléments de A^r, les combinaisons linéaires finies à coefficients dans A forment un module noté $M = \sum_{u\in U} Ag_u$. Les quotients de la forme A^r/M sont des modules sur A.

La généralisation de la théorie repose sur la bonne notion d'ordre pour les modules. Considérons la base canonique $B = (b_i)_{1\leq i\leq p}$ de A^r, avec b_i le vecteur-ligne de A^r constitué de 0 sauf en i-ème colonne où 0 est remplacé par 1. On a $A^r = Ab_1 \oplus \cdots \oplus Ab_r$. Deux façons d'étendre les ordres monomiaux de M à à $M \times B$ sont toute naturelles :

– l'ordre top, pour *term over position*, pour lequel les mb_i sont triés sur m, puis sur i,

– l'ordre pot, pour *position over term*, pour lequel les mb_i sont triés sur i puis sur m.

On obtient ainsi des ordres $\prec_{\mathsf{top,lex}}$, $\prec_{\mathsf{pot,lex}}$, $\prec_{\mathsf{top,grevlex}}$, $\prec_{\mathsf{pot,grevlex}}$, etc.

De même, on introduit des S-polynômes pour les modules — ou devrions-nous dire « S-vecteurs » ? — en généralisant la remarque faite dans l'interprétation de l'algorithme de Gauss en termes de l'algorithme de Buchberger. Ici encore, on déclare nul un S-polynôme de deux vecteurs ayant des monômes de tête sur des b_i et b_j différents. Soient deux vecteurs $p_1 = \sum_{i=1}^r p_{1,i}b_i$ et $p_2 = \sum_{i=1}^r p_{2,i}b_i$ deux éléments de M et posons $m_1 b_{i_1} = \mathrm{mt}(p_1)$, $m_2 b_{i_2} = \mathrm{mt}(p_2)$ et $m = \mathrm{ppcm}(m_1, m_2) = n_1 m_1 = n_2 m_2$. Lorsque $i_1 = i_2$, on appelle *S-polynôme* de p_1 et p_2 toute combinaison linéaire de la forme $l_1 p_1 + l_2 p_2$ pour tous polynômes l_i tels que $\mathrm{mt}(l_i) = n_i$ et $\mathrm{tt}(l_1)\,\mathrm{tt}(p_{1,i}) + \mathrm{tt}(l_2)\,\mathrm{tt}(p_{2,i}) = 0$. Lorsque $i_1 \neq i_2$, on déclare que le S-polynôme de p_1 et p_2 est nul, ou bien on dit qu'ils n'ont pas de S-polynôme. En se restreignant à des l_i monômiaux, on a ainsi la formule synthétique

$$\mathrm{Spoly}(p_1, p_2) = \begin{cases} \mathrm{ct}(p_2)n_1 p_1 - \mathrm{ct}(p_1)n_2 p_2 & \text{si } i_1 = i_2, \\ 0 & \text{sinon.} \end{cases}$$

3.4. Application : base de Gröbner en terme des polynômes initiaux. Traitons succintement une application de la théorie des bases de Gröbner aux modules sur un anneau de polynômes : étant donnés des polynômes $p_1, \ldots, p_r$, on cherche à exprimer les éléments d'une base de Gröbner de l'idéal engendré I en terme des p_i.

Dans $A^{r+1} = \bigoplus_{i=0}^r Ab_i$, considérons une base de Gröbner pour un ordre pot du sous-module S engendré par les $b_0 - p_i b_i$. Cette modélisation réalise l'invariant que tout vecteur $c_0 b_0 + \cdots + c_r b_r$ considéré au court du calcul vérifie la relation $c_0 = \sum_{i=1}^r c_i p_i$. On vérifie ainsi que la base de Gröbner pour S contient :

– des éléments de la forme $gb_0 - \sum_{i=1}^r l_i b_i$, pour lesquels $g = \sum_{i=1}^r l_i p_i$. Les g constituent alors une base de Gröbner de I.

– des éléments de la forme $\sum_{i=1}^r u_i b_i$, pour lesquels $\sum_{i=1}^r u_i p_i = 0$. Ceux-ci engendrent ainsi le module des relations entre les p_i.

4. Complexité intrinsèque et bases de Gröbner

Dans cette section, nous annonçons très brièvement quelques résultats qui montrent la nature exponentielle ou doublement exponentielle de problèmes rattachés à la notion de base de Gröbner ou à leur calcul.

4.1. Problèmes complets. Le problème général de la recherche d'une base de Gröbner réduite et le problème d'appartenance à un idéal générique de polynômes sont des problèmes EXPSPACE-complets (pour des coefficients dans $\mathbb{Q}$).

Restreints à des idéaux binomiaux (engendrés par des binômes $X^\alpha - X^\beta$), il en est de même pour les deux problèmes.

Restreint à des idéaux homogènes (engendrés par des polynômes dont tous les monômes ont même degré), le problème d'appartenance à un idéal n'est que PSPACE-complet.

4.2. Taille de la sortie. Les degrés des polynômes d'une base de Gröbner réduite d'un idéal $Ap_1 + \cdots + Ap_s \subseteq \overline{\mathbb{Q}}[X_1, \ldots, X_n]$, pour des p_i de degré au plus d, sont au plus

$$2\left(\frac{d^2}{2} + d\right)^{2^n - 1}.$$

Il existe des idéaux dont toutes les bases de Gröbner contiennent au moins $2^{2^{cn}}$ éléments et des éléments de degré au moins $2^{2^{c'n}}$, pour des constantes réelles c et c' strictement positives.

CHAPITRE 7

Calculs en Magma

En *Magma*, la création d'un anneau de polynômes nécessite de déclarer d'abord le corps de coefficients sur lequel on travaille, ici le corps $\mathbb{Q}$ des nombres rationnels, puis de déclarer le nombre d'indéterminées transcendantes qui servent à étendre ce corps en un anneau de polynômes. Ce nombre est appelé le « rang » de l'anneau par *Magma*.

```
> Q:=RationalField();
> A:=PolynomialRing(Q,3);
> Rank(A);
3
> [A.i:i in [1..Rank(A)]];
[
    $.1,
    $.2,
    $.3
]
```

Dans l'exemple précédent, les indéterminées de `A` ne sont pas nommées. On peut y faire référence par `A.1`, `A.2` et `A.3`. Pour éviter cette notation, il est possible de voir ces objets sous d'autres noms par la notation suivante.

```
> Q:=RationalField();
> A<X,Y,Z>:=PolynomialRing(Q,3);
> [A.i:i in [1..Rank(A)]];
[
    X,
    Y,
    Z
]
```

Cet appel a effectivement affecté la valeur `A.1` à la variable `X`. Attention à la difficulté suivante de *Magma* : après déclaration d'un autre anneau utilisant aussi `X`, il n'est plus possible d'accéder à `A.1` à l'aide de `X`, bien qu'à l'affichage `A.1` soit toujours présenté comme `X`.

Une fois un anneau créé, on peut introduire un polynôme ou un idéal de celui-ci.

```
> f:=X^3-1;
> I:=ideal<A|X+Y+Z,X*Y+Y*Z+Z*X,X*Y*Z-1>;
> I;
Ideal of Polynomial ring of rank 3 over Rational Field
Lexicographical Order
Variables: X, Y, Z
Basis:
[
    X + Y + Z,
    X*Y + X*Z + Y*Z,
    X*Y*Z - 1
]
```

Un idéal est donné par un système de générateurs, notion pour laquelle *Magma* utilise le mot anglais *basis*. Un idéal admettant plusieurs systèmes de générateurs, *Magma* change librement

le système en cours d'utilisation, selon les calculs envisagés. En particulier, lorsqu'une base de Gröbner est calculée, elle devient le nouveau système de générateurs en cours.

La déclaration de l'anneau `A` ci-dessus est en fait la déclaration d'un anneau muni d'un ordre monomial. Implicitement, c'est l'ordre $\prec_{\mathsf{lex}(X,Y,Z)}$ qui a été choisi. L'exemple qui suit montre l'effet du calcul d'une base pour cet ordre.

```
> Basis(I);
[
    X + Y + Z,
    X*Y + X*Z + Y*Z,
    X*Y*Z - 1
]
> Groebner(I);
> Basis(I);
[
    X + Y + Z,
    Y^2 + Y*Z + Z^2,
    Z^3 - 1
]
```

Une fois une base calculée, la mise sous forme normale modulo l'idéal et le test d'appartenance à l'idéal se font comme ci-dessous. Puisque notre polynôme `f` est dans l'idéal, on peut donner ses coordonnées sur la base de Gröbner qui sert de système de générateurs en cours.

```
> NormalForm(f,I);
0
> f in I;
true
> Coordinates(I,f);
[
    X^2 - X*Y - X*Z + Y^2 + 2*Y*Z + Z^2,
    -Y - 2*Z,
    1
]
```

Mais rechercher les coordonnées d'un polynôme hors de l'idéal provoque une erreur.

```
> NormalForm(f+1,I);
1
> f+1 in I;
false
> Coordinates(I,f+1);

>> Coordinates(I,f+1);
              ^
Runtime error in 'Coordinates': Argument 2 is not
in argument 1
```

Intéressons-nous maintenant à d'autres ordres monomiaux. Ceux-ci doivent être déclarés dès la création de l'anneau de polynômes. En effet, c'est dès cet instant que se décide la représentation de données qui va servir pour le stockage des polynômes. Ici, nous donnons un ordre d'élimination, ou ordre par blocs, qui place `A.1` et `A.2`, à savoir `T` et `U` lexicographiquement avant les deux autres générateurs de l'anneau, et trie les monômes en T et U par l'ordre par $\mathsf{grevlex}(T,U)$.

```
> Q:=RationalField();
> A<T,U,X,Y>:=PolynomialRing(Q,4,"elim",[1,2]);
> I:=ideal<A|(1+T^2)*X-(1-T^2),(1+T^2)*Y-2*T,(1+T^2)*U-1>;
> Groebner(I);
> Basis(I);
[
```

```
        T*X + T - Y,
        T*Y + X - 1,
        U - 1/2*X - 1/2,
        X^2 + Y^2 - 1
    ]
    > [b:b in Basis(I)|Degree(b,T) eq 0 and Degree(b,U) eq 0];
    [
        X^2 + Y^2 - 1
    ]
```

Magma dispose aussi de fonctions pour calculer des bases de Gröbner pour les modules. Avant tout, il convient de déclarer le module libre `M` dans lequel on va décrire un sous-module `S` donné par générateurs. Ci-dessous, le module `M` est le module libre sur `P` de rang 2, le rang (mathématique) étant appelé *degree* par *Magma*. Les éléments de ce module sont présentés comme des vecteurs lignes de longueur 2.

```
> Q:=RationalField();
> P<X,Y,Z>:=PolynomialRing(Q,3,"grevlex");
> M:=Module(P,2);
> M;
Full Module of degree 2
TOP Order
Coefficient ring:
    Polynomial ring of rank 3 over Rational Field
    Graded Reverse Lexicographical Order
    Variables: X, Y, Z
> S:=sub<M|(X^2-1)*M.1+(X*Y-Z)*M.2,(Y^2+X)*M.1+(3*Z-X)*M.2>;
> S;
Module of degree 2
TOP Order
Coefficient ring:
    Polynomial ring of rank 3 over Rational Field
    Graded Reverse Lexicographical Order
    Variables: X, Y, Z
Basis:
( X^2 - 1  X*Y - Z)
( Y^2 + X -X + 3*Z)
> Groebner(S);
> Basis(S);
[
    (0   X*Y^3 + X^3 + X^2*Y - 3*X^2*Z - Y^2*Z - X*Z - X + 3*Z),
    (X^2 - 1 X*Y - Z),
    ( Y^2 + X -X + 3*Z)
]
>
```

CHAPITRE 8

Réduction de réseaux et algorithme LLL

La recherche de vecteurs courts dans des réseaux de vecteurs à coefficients entiers permet, en calcul formel, la factorisation en temps polynomial et la recherche de dépendances linéaires entre constantes réelles données par des approximations, et, dans des domaines plus proches de la cryptanalyse, la mise en évidence de faiblesses de cryptosystèmes et de générateurs pseudo-aléatoires. Nous présentons l'algorithme LLL maintenant célèbre pour la réduction de réseaux et analysons sa complexité.

Ce texte suit d'assez près mais dans un autre ordre les chapitres 16 et 17 du livre *Modern computer algebra* de von zur Gathen et Gerhard.

1. Réseaux, vecteurs courts et résultats principaux

On appelle *réseau* de $\mathbb{Z}^n$ un $\mathbb{Z}$-module $\sum_{i=1}^{n} \mathbb{Z}v_i$ engendré par des vecteurs v_i linéairement indépendants sur $\mathbb{Z}$. Le thème de ces notes de cours est la recherche de vecteurs « courts » dans un réseau donné : par exemple, le réseau engendré par les vecteurs $(12, 2)$ et $(13, 4)$ contient le vecteur plus court $(2, 1)$. Ici, court s'entend pour la norme euclidienne : un vecteur $v = (v_1, \ldots, v_n)$ a pour norme $\|v\| = (v_1^2 + \cdots + v_n^2)^{1/2}$.

Cette question est motivée par un certain nombre d'applications :

- factorisation en temps polynomial ;
- cassage d'un cryptosystème basé sur le problème du sac-à-dos ;
- mise en évidence de la faiblesse de générateurs pseudo-aléatoires ;
- recherche de dépendances linéaires entre nombres donnés par des approximations numériques, dont la recherche du polynôme minimal d'un nombre algébrique.

La première de ces applications fera l'objet d'un autre cours ; la deuxième et la dernière sont détaillées dans la section qui suit.

La recherche d'un vecteur de norme minimale dans un réseau donné est un problème difficile, en fait, même NP-dur. Il est donc naturel de relâcher la contrainte de minimalité et de considérer le problème de la recherche approchée à un facteur constant près, mais le problème de la recherche à un facteur $\sqrt{2}$ près reste NP-dur. En revanche, le problème redevient polynomial si on autorise le facteur d'approximation à croître avec la dimension n du réseau.

De façon précise, on décrit dans la suite l'algorithme LLL, qui pour une base $(v_1, \ldots, v_n)$ d'un réseau L renvoie en particulier un vecteur u approximant les plus courts vecteur du réseau à pas plus d'un facteur $2^{(n-1)/2}$ près :

$$\|u\| \leq 2^{(n-1)/2} \min\{\, \|f\| : f \in L,\ f \neq 0 \,\}.$$

Cet algorithme termine après $O(n^4 \log A)$ opérations arithmétiques correspondant à $O_{\log}(n^5 \log^2 A)$ opérations binaires, où la constante A borne chacun des v_i. (Seule la complexité arithmétique est démontrée dans ce qui suit.)

2. Applications

2.1. Cassage du cryptosystème de Merkle et Hellman. Merkle et Hellman ont proposé en 1978 un cryptosystème basé sur le problème du sac-à-dos. Ce cryptosystème devait nécessiter des calculs moins lourds que le célèbre système RSA. Néanmoins, une faiblesse du système reposait sur l'existence d'un vecteur court dans un réseau.

L'idée de Merkle et Hellman est la suivante. Bob se dote d'une clé secrète, une famille d'entiers positifs b_1, …, b_n, telle que chaque b_i soit supérieur à la somme des précédents. Bob se donne

aussi un multiplicateur c et un module m, eux deux aussi secrets. Il publie la clé privée constituée des restes positifs a_i de cb_i modulo m. Quand Alice veut lui envoyer le message constitué de la suite de bits $(x_1, \dots, x_n)$, elle construit la somme $s = x_1a_1 + \cdots + x_na_n$ qui constitue le message crypté. Bob n'a plus qu'à multiplier par l'inverse de c modulo m et à tester si les b_i sont dans la somme restante (en commençant par les plus grands) pour déterminer les x_i.

Le problème sur lequel s'appuie ce cryptosystème, celui du sac-à-dos, consiste précisément à décider l'existence des $x_i \in \{0, 1\}$ tels que $s = x_1a_1 + \cdots + x_na_n$. Ce problème est NP-complet en l'absence d'hypothèse sur la famille des a_i. Mais dans le cas présent, l'origine des a_i crée une faiblesse cryptographique. Considérons les vecteurs $(0, \dots, 0, 1, 0, \dots, 0, -a_i) \in \mathbb{Z}^{n+1}$, où 1 est en i-ième position, ainsi que le vecteur $(0, \dots, 0, s)$. Tous ces vecteurs sont « longs », puisque de l'ordre de m, mais le réseau qu'ils engendrent contient le vecteur $(x_1, \dots, x_n, 0)$, lequel est manifestement très court. L'algorithme LLL qui va suivre permet de le retrouver en complexité polynomiale.

2.2. Relations de dépendance entre constantes numériques. Étant donnés n nombres réels non nuls, r_1, …, r_n, une relation de dépendance linéaire à coefficients entiers entre ces réels est une relation de la forme

$$c_1r_1 + \cdots + c_nr_n = 0$$

pour des coefficients c_i entiers non tous nuls. Lorsqu'on se donne une approximation rationnelle de chaque réel r_i, ou mieux, la possibilité d'obtenir autant de chiffres décimaux que souhaité, la recherche des vecteurs courts dans un réseau permet la détermination de relations de dépendance linéaire entre les r_i. Pour cela, on considère les combinaisons linéaires à coefficients entiers des vecteurs lignes de la matrice

$$F = \begin{bmatrix} 1 & 0 & \dots & 0 & a_1 \\ 0 & 1 & \dots & 0 & a_2 \\ & & \ddots & & \vdots \\ 0 & \dots & 0 & 1 & a_{n-1} \\ 0 & \dots & 0 & 0 & a_n \end{bmatrix},$$

où chaque a_i est une troncature entière de Nr_i pour un grand entier N. (Par exemple, N est une grande puissance de 10.)

Un vecteur court est alors de la forme

$$v = (c_1, \dots, c_{n-1}, c_1a_1 + \cdots + c_na_n) \simeq \big(c_1, \dots, c_{n-1}, N(c_1r_1 + \cdots + c_nr_n)\big).$$

En particulier,

$$|c_1r_1 + \cdots + c_nr_n| \simeq \big|N^{-1}(c_1a_1 + \cdots + c_na_n)\big| \leq N^{-1}|v|$$

se doit d'être petit, ce qui fait de

$$c_1r_1 + \cdots + c_nr_n = 0$$

un bon candidat pour une relation de dépendance entre les r_i. Bien qu'un tel argument ne constitue pas une preuve, la preuve formelle de relation entre constantes a dans un bon nombre de cas été grandement facilitée une fois que la bonne relation a pu être découverte expérimentalement par la méthode proposée ci-dessus. Dans un certain nombre de cas, même, des identités n'ont pu être prouvées que parce qu'elles avaient été découvertes heuristiquement en utilisant l'algorithme LLL.

Donnons un exemple. Des arguments théoriques donnent la certitude que le nombre

$$V = \int_0^\infty \frac{\sqrt{x}\,\ln^5 x}{(1-x)^5}\,\mathrm{d}x$$

peut se représenter comme l'évaluation en π d'une fonction polynomiale à coefficients rationnels. Il s'agit donc de trouver une dépendance linéaire entre

$$V, 1, \pi, \dots, \pi^d$$

pour un degré de polynôme d à déterminer. Tout système de calcul formel généraliste connait des approximations pour π et permet de calculer aisément une approximation numérique de V. On prend donc par exemple $a_1 = N = 10^{25}$, $a_2 = 31415926535897932384626434 \simeq N\pi$, …, $a_9 = 94885310160705740071285755038 \simeq N\pi^8$, $a_{10} = -16699473719229070496187243 3 \simeq NV$ pour

construire la matrice F. Les normes des vecteurs lignes de cette matrice sont toutes supérieures à $N = 10^{25}$. Par l'algorithme LLL, on trouve une base du même réseau de vecteurs, dont le premier vecteur ligne,

$$v = (c_1, \dots, c_{n-1}, c_1 a_1 + \cdots + c_n a_n) = (0, 0, 120, 0, 140, 0, -15, 0, 0, 33),$$

est de norme inférieure à 200, tous les autres vecteurs de base étant de norme supérieure à 400. Les a_i étant connus, on trouve

$$(c_1, \dots, c_n) = (0, 0, 120, 0, 140, 0, -15, 0, 0, 24),$$

d'où la relation linéaire

$$V = \int_0^\infty \frac{\sqrt{x}\,\ln^5 x}{(1-x)^5}\,\mathrm{d}x = \frac{5\pi^2}{24}(3\pi^4 - 28\pi^2 - 24)$$

qui résout le problème posé.

2.3. Polynôme minimal de nombres algébriques. Un nombre complexe α est dit algébrique (sur les nombres rationnels) lorsqu'il est solution d'un polynôme P à coefficients rationnels

$$P(\alpha) = p_0 + \cdots + p_d \alpha^d = 0.$$

Pour pouvoir utiliser l'approche de la section précédente, on se limite au cas de nombres réels.

Étant donnée une approximation numérique (réelle) d'un nombre α qu'on a de bonnes raisons de penser être algébrique, la détermination heuristique de P peut se voir comme la recherche d'une relation de dépendance linéaire sur des approximations numériques rationnelles de

$$1, \alpha, \dots, \alpha^d.$$

Dès lors qu'on a une borne supérieure sur le degré d de P, on peut donc employer la méthode de la section précédente.

Par exemple, soit à déterminer si un nombre

$$r \simeq 0{,}266252646290196114530247765575844548176501286103 95\dots$$

est algébrique. Par la méthode esquissée, en testant jusqu'à $d = 6$ et pour $N = 10^{28}$, on trouve (à partir de vecteurs lignes de norme supérieure à 10^{20}), le vecteur court

$$v = (-1, 0, 0, 54, 0, 0, -10),$$

de norme inférieure à 55, tous les autres vecteurs calculés étant de norme supérieure à 5000. En poursuivant avec la méthode, on trouve

$$(c_1, \dots, c_7) = (-1, 0, 0, 54, 0, 0, -54),$$

soit

$$P = 54X^6 - 54X^3 + 1.$$

En effet, le nombre r s'avère être une approximation du nombre algébrique

$$\alpha = \sqrt[3]{\frac{1}{2} - \frac{5\sqrt{3}}{18}}.$$

3. Le procédé d'orthogonalisation de Gram–Schmidt

L'algorithme LLL aura fort à voir avec le procédé d'orthogonalisation de Gram–Schmidt qui, partant d'une base $(f_1, \dots, f_n)$ d'un espace vectoriel sur $\mathbb{Q}$, calcule une base orthogonale du même espace vectoriel.

Le contexte de cette méthode est celui de l'espace vectoriel $\mathbb{Q}^n$ muni du produit scalaire euclidien $(a, b) \mapsto (a|b) = a_1 b_1 + \cdots + a_n b_n$ qui a servi à définir la norme euclidienne par $\|a\|^2 = (a|a)$. Rappelons que les vecteurs a et b sont dits orthogonaux lorsque leur produit scalaire est nul et que l'orthogonal d'un ensemble $S \subset \mathbb{Q}^n$ est l'ensemble noté $S^\perp$ des vecteurs orthogonaux à tous les éléments de S. On introduit les espaces vectoriels emboîtés $U_i = \mathbb{Q}f_1 \oplus \cdots \oplus \mathbb{Q}f_i$. La projection orthogonale sur U_i est le projecteur linéaire sur U_i parallèlement à $U_i^\perp$. On définit alors f_i^* comme la projection orthogonale de f_i sur $U_i^\perp$ (donc parallèlement à $U_i^{\perp\perp}$, qui n'est autre que U_i).

Les considérations qui précèdent définissent uniquement les f_i^* à partir des f_i et des U_i ; pour le calcul, on détermine les f_i^* pour des i successifs par les formules

$$f_i^* = f_i - \sum_{j=1}^{i-1} \mu_{i,j} f_j^* \qquad \text{où} \qquad \mu_{i,j} = \frac{(f_i|f_j^*)}{(f_j^*|f_j^*)},$$

où $\mu_{i,j}$ est ajusté à l'unique valeur convenable pour avoir orthogonalité entre f_i^* et f_j^* quand $i > j$.

Après avoir posé $\mu_{i,i} = 1$ et $\mu_{i,j} = 0$ quand $i < j$, on obtient une matrice

$$M = (\mu_{i,j}) = \begin{bmatrix} 1 & & 0 \\ & \ddots & \\ \mu_{i,j} & & 1 \end{bmatrix}.$$

vérifiant la relation

$$M \begin{bmatrix} f_1^* \\ \vdots \\ f_n^* \end{bmatrix} = \begin{bmatrix} f_1 \\ \vdots \\ f_n \end{bmatrix}.$$

En particulier, les espaces vectoriels sur $\mathbb{Q}$ engendrés par les f_i d'une part et par les f_i^* d'autre part sont les mêmes.

Soit maintenant un vecteur non nul f du réseau engendré par les f_i, qui s'écrit donc $f = \lambda_1 f_1 + \cdots + \lambda_n f_n$ pour des entiers λ_i non tous nuls. En passant au carré de la norme et en utilisant le théorème de Pythagore, on a

$$\|f\|^2 = \sum_{i=1}^{n} \lambda_i^2 \|f_i^*\|^2 \geq \|f_k^*\|^2 \geq \left(\min_{j=1}^{n} \|f_j^*\| \right)^2$$

dès lors que $\lambda_k \neq 0$, car alors $\lambda_k^2 \geq 1$.

À ce stade, nous aurions trouvé des vecteurs parmi les plus courts, si ce n'est que les f_i^* sont généralement éléments de l'espace vectoriel engendré par les f_i (avec des coefficients rationnels), mais hors du réseau engendré par les même f_i (avec des coefficients entiers). Il n'en reste pas moins que l'algorithme LLL qui va suivre s'appuie sur l'orthogonalisation de Gram–Schmidt de façon à contrôler des transformations de la base du réseau, car, à un niveau intuitif, plus une base de réseau est « orthogonale », plus elle est à même de contenir un vecteur court. Cette heuristique se justifie par la notion de déterminant d'un réseau : étant donné une base $(f_1, \ldots, f_n)$ d'un réseau, le déterminant de cette famille, c'est-à-dire le déterminant de la matrice $F = (f_{i,j})$ obtenue en plaçant les vecteurs lignes $f_i = (f_{i,1}, \ldots, f_{i,n})$ les uns au dessus des autres, est, au signe près, invariant par changement de base. En effet, soit $(g_1, \ldots, g_n)$ une autre base du même réseau et $G = (g_{i,j})$ la matrice associée, avec des notations évidentes ; il existe alors une matrice U à coefficients entiers, admettant un inverse à coefficients entiers, telle que $F = UG$. En passant aux déterminants, on a que $\det U$ vaut 1 au signe près, donc que F et G ont même déterminant au signe près. Toujours au signe près, ce déterminant n'est autre que le volume du parallèloïde construit en s'appuyant sur les f_i. Les côtés de ce parallèloïde sont d'autant plus courts que ce parallèloïde est orthogonal.

Une base $(f_1, \ldots, f_n)$ d'un réseau est dite *réduite* lorsque les images f_i^* des f_i par le procédé d'orthogonalisation de Gram–Schmidt ont la propriété que $\|f_i^*\|^2 \leq 2\|f_{i+1}^*\|^2$ pour tout i entre 1 et $n-1$. En particulier, f_i^* n'est dans ce cas pas plus de $2^{(i-1)/2}$ fois plus petit que f_1^*, c'est-à-dire que f_1. Il s'ensuit que pour tout vecteur non-nul f du réseau, on a la relation

$$\|f\| \geq \min_{j=1}^{n} \|f_j^*\| \geq \min_{j=1}^{n} 2^{-(j-1)/2} \|f_1\| \geq 2^{-(n-1)/2} \|f_1\|.$$

Autrement dit, le premier élément d'une base réduite est un vecteur court, au sens où pour tout f non nul du réseau

$$\|f_1\| \leq 2^{(n-1)/2} \|f\|.$$

4. L'algorithme LLL

L'algorithme qui suit a été introduit par Lenstra, Lenstra et Lovász en 1982 dans l'objectif de réaliser la factorisation de polynômes d'une variable à coefficients entiers en complexité arithmétique polynomiale. Pour un réel x, on note $\lceil x \rfloor$ l'entier de plus proche de x (par défaut, l'entier immédiatement inférieur si x est un demi-entier).

Algorithme (Réduction de réseau « LLL »).

Entrée : $f_1, \ldots, f_n$, des vecteurs de $\mathbb{Z}^n$ engendrant un réseau

Sortie : $g_1, \ldots, g_n$, des vecteurs de $\mathbb{Z}^n$ constituant une base réduite du même réseau

Complexité : $O(n^4 \log A)$ opérations arithmétiques et $O(n^5 \log^2 A)$ opérations binaires, où $A = \max_{i=1}^n \|f_i\|$

(1) Initialiser g_i à f_i pour chaque i de 1 à n

(2) Calculer l'orthogonalisation de Gram–Schmidt (g_i^*) de (g_i), ainsi que la matrice M des $\mu_{i,j}$ telle que $M\,{}^t(g_1^*, \ldots, g_n^*) = {}^t(g_1, \ldots, g_n)$

(3) Pour i à partir de 2, tant que $i \leq n$, faire

  (a) pour j de $i-1$ à 1, remplacer g_i par $g_i - \lceil \mu_{i,j} \rfloor g_j$ et réaliser la même transformation sur les lignes de M

  (b) si $i \geq 2$ et si $\|g_{i-1}^*\|^2 > 2\|g_i^*\|^2$,

    (i) échanger g_{i-1} et g_i

    (ii) recalculer g_{i-1}^* et g_i^* et les lignes correspondantes de M par le procédé de Gram–Schmidt

    (iii) décrémenter i

  sinon incrémenter i

(4) Renvoyer $(g_1, \ldots, g_n)$

Traitons l'exemple donné par les deux vecteurs $f_1 = (12, 2)$ et $f_2 = (13, 4)$ de $\mathbb{Z}^2$. Après les étapes (1) et (2), on a la relation

$$\begin{bmatrix} 1 & 0 \\ \frac{41}{37} & 1 \end{bmatrix} \begin{bmatrix} 12 & 2 \\ -\frac{11}{37} & \frac{66}{37} \end{bmatrix} = \begin{bmatrix} 12 & 2 \\ 13 & 4 \end{bmatrix}.$$

Comme $\lceil \frac{41}{37} \rfloor = 1$, la transformation (de réduction) de l'étape (3)(a) revient à des multiplications à gauche par la matrice $\left[\begin{smallmatrix} 1 & 0 \\ -1 & 1 \end{smallmatrix}\right]$ et transforme la relation matricielle ci-dessus en

$$\begin{bmatrix} 1 & 0 \\ \frac{4}{37} & 1 \end{bmatrix} \begin{bmatrix} 12 & 2 \\ -\frac{11}{37} & \frac{66}{37} \end{bmatrix} = \begin{bmatrix} 12 & 2 \\ 1 & 2 \end{bmatrix}.$$

Les normes à considérer vérifient $\|g_1^*\|^2 = 4 \times 37 > 2\|g_2^*\|^2 = 2 \times 11^2/37$, ce qui provoque un échange en (3)(b)(i) avec recalcul en (3)(b)(ii) par le procédé de Gram–Schmidt, et fournit la nouvelle relation matricielle

$$\begin{bmatrix} 1 & 0 \\ \frac{16}{5} & 1 \end{bmatrix} \begin{bmatrix} 1 & 2 \\ \frac{44}{5} & -\frac{22}{5} \end{bmatrix} = \begin{bmatrix} 1 & 2 \\ 12 & 2 \end{bmatrix}.$$

Comme $\lceil \frac{16}{5} \rfloor = 3$, une nouvelle transformation (de réduction) à l'étape (3)(a) revient à des multiplications à gauche par la matrice $\left[\begin{smallmatrix} 1 & 0 \\ -3 & 1 \end{smallmatrix}\right]$ et transforme la relation matricielle ci-dessus en

$$\begin{bmatrix} 1 & 0 \\ \frac{1}{5} & 1 \end{bmatrix} \begin{bmatrix} 1 & 2 \\ \frac{44}{5} & -\frac{22}{5} \end{bmatrix} = \begin{bmatrix} 1 & 2 \\ 9 & -4 \end{bmatrix}.$$

Comme $\|g_1^*\|^2 = 5 \leq 2\|g_2^*\| = 2 \times 11^2 \times 4$, l'algorithme termine en renvoyant $g_1 = (1, 2)$ et $g_2 = (9, -4)$.

5. Preuve de l'algorithme LLL

Nous développons maintenant des invariants et des variants de l'algorithme qui permettent de montrer la correction et la terminaison de l'algorithme, puis une borne polynomiale sur sa complexité arithmétique.

5.1. Correction. À l'entrée de la boucle (3), la base de réseau (g_i) et son orthogonalisée (g_i^*) par la méthode de Gram–Schmidt vérifient la relation $M\,{}^t(g_1^*, \ldots, g_n^*) = {}^t(g_1, \ldots, g_n)$. Observons d'abord que cet invariant reste respecté à chaque sortie du corps de boucle. Il est immédiat que la relation matricielle reste vérifiée après la transformation de l'étape (3)(a) ; montrons que, pour λ quelconque et tout $j < i$, le remplacement de g_i par $g_i - \lambda g_j$ et de chaque $\mu_{i,k}$ par $\mu_{i,k} - \lambda\mu_{j,k}$ quand $1 \leq k \leq j$ fait de la nouvelle famille des g_i^* l'orthogonalisée de la nouvelle famille des g_i. (Dans l'algorithme, on choisit $\lambda = \lceil \mu_{i,j} \rfloor$.) Puisque la famille d'espaces emboîtés $\mathbb{Q}g_1 \oplus \cdots \oplus \mathbb{Q}g_k$, d'une part, et celle des $\mathbb{Q}g_1^* \oplus \cdots \oplus \mathbb{Q}g_k^*$, d'autre part, n'ont pas changé, il suffit pour cela de vérifier que le nouveau g_i^* vérifie bien sa définition. En termes des anciennes valeurs, ceci se résume, pour $j < i$, à vérifier la relation

$$g_i^* = g_i - \sum_{k=1}^{i-1} \mu_{i,k} g_k^* = \big(g_i - \lambda g_j\big) - \sum_{k=1}^{i-1} \big(\mu_{i,k} - \lambda\mu_{j,k}\big) g_k^* + \lambda\left(g_j - \sum_{k=1}^{j} \mu_{j,k} g_k^*\right).$$

La dernière parenthèse étant nulle, l'invariant recherché reste préservé par la transformation (3)(a). Quand à l'étape (3)(b), on observe de nouveau que la famille d'espaces emboîtés $\mathbb{Q}g_1 \oplus \cdots \oplus \mathbb{Q}g_k$, d'une part, et celle des $\mathbb{Q}g_1^* \oplus \cdots \oplus \mathbb{Q}g_k^*$, d'autre part, ne sont pas perturbées par l'échange (3)(b)(i), sauf peut-être pour $k = i$ ou $k = i - 1$. C'est pourquoi l'algorithme recalcule explicitement g_{i-1}^* puis g_i^* par les formule de Gram–Schmidt pour restaurer l'invariant.

Un second invariant respecté par la même boucle (3) est la suite d'inégalités $\|g_1^*\|^2 \leq 2\|g_2^*\|^2, \ldots$, $\|g_{i-2}^*\|^2 \leq 2\|g_{i-1}^*\|^2$. Cet invariant se réduit en effet à une absence de contraintes à l'entrée de la boucle (3), pour $i = 2$. Puis, la stratégie de l'étape (3)(b) est de simplement incrémenter i si la suite d'inégalité peut se prolonger par $\|g_{i-1}^*\|^2 \leq 2\|g_i^*\|^2$, ou au contraire de décrémenter i lorsque g_{i-1}^* est modifié.

Ainsi, si une exécution de l'algorithme sort de la boucle (3), la base (g_i) est réduite, ce qui prouve la correction de l'algorithme.

5.2. Terminaison. Pour montrer que l'algorithme termine, on va montrer que l'échange et la mise à jour (3)(b)(i–ii), seule étape qui modifie les g_i^*, les réduit en norme, et réduit dans le même temps une grandeur entière positive associée aux carrés des normes $\|g_i\|^2$. Le nombre d'échange ne pourra donc être que fini. Un invariant entre nombre d'incrémentations et de décrémentations dans l'étape (3)(b) montre alors la terminaison de la boucle (3), et donc de l'algorithme.

Pour un passage dans l'étape (3)(b) avec $i \geq 2$ et $\|g_{i-1}^*\|^2 > 2\|g_i^*\|^2$, notons (h_k) la base du réseau obtenue après l'échange (3)(b)(i) et (h_k^*) l'orthogonalisée de Gram–Schmidt obtenue après le recalcul (3)(b)(ii). On a $h_i = g_{i-1}$, $h_{i-1} = g_i$, et $h_k = g_k$ sinon. De même, on a les égalités entre espaces vectoriels $\mathbb{Q}g_1 \oplus \cdots \oplus \mathbb{Q}g_k = \mathbb{Q}h_1 \oplus \cdots \oplus \mathbb{Q}h_k$, sauf pour $k = i - 1$. Ainsi, les vecteurs orthogonalisés h_k^* et g_k^* sont égaux, sauf peut-être pour $k = i$ et $k = i - 1$. Posons $U = \mathbb{Q}g_1 \oplus \cdots \oplus \mathbb{Q}g_{i-2}$. Par la définition du procédé d'orthogonalisation de Gram–Schmidt, on a d'abord $g_i = h_{i-1} = h_{i-1}^* + u_1$ pour $u_1 \in U$, et aussi $g_i = g_i^* + \mu_{i,i-1} g_{i-1}^* + u_2$ pour $u_2 \in U$. Comme u_1 et u_2 sont les projetés orthogonaux du même vecteur orthogonalement à U, ils sont égaux, de même que h_{i-1}^* et $g_i^* + \mu_{i,i-1} g_{i-1}^*$. En passant aux carrés des normes, on obtient $\|h_{i-1}^*\|^2 = \|g_i^*\|^2 + \mu_{i,i-1}^2 \|g_{i-1}^*\|^2 < \|g_{i-1}^*\|^2/2 + (1/2)^2 \|g_{i-1}^*\|^2$, où on a utilisé l'hypothèse que le test en (3)(b) a été positif et le fait que l'étape (3)(a) a forcé la relation $\mu_{i,i-1} \leq 1/2$. En résumé, $\|h_{i-1}^*\|^2 < 3\|g_{i-1}^*\|^2/4$. Par la même méthode, en considérant des projections orthogonales de g_{i-1} sur $U' = U \oplus \mathbb{Q}g_{i-1}$, on montre la relation $\|h_i^*\| \leq \|g_i^*\|$.

Notons G_i la matrice obtenue en superposant les i premiers vecteurs $g_1, \ldots, g_i$, et G_i^* celle obtenue à partir des g_k^* correspondants. La matrice de Gram des vecteurs $g_1, \ldots, g_i$ est la matrice

de taille $i \times i$ dont l'entrée (u, v) est le produit scalaire $(g_u|g_v)$. C'est donc une matrice de $\mathbb{Z}^{i\times i}$, qui s'exprime aussi $G_i\,{}^tG_i$. Puisque le bloc M_i de taille $i \times i$ en haut à gauche dans la matrice M est encore trigonal avec des 1 sur la diagonale, donc de déterminant 1, on a

$$d_i = \det(M_i G_i^{*}\,{}^tG_i^{*t} M_i) = \det(M_i)\det(G_i^{*}\,{}^tG_i^{*})\det({}^tM_i) = \|g_1^*\|^2 \dots \|g_i^*\|^2.$$

Ce nombre est donc un entier strictement positif.

Le variant qui va s'avérer adéquat est le produit $D = d_1 \dots d_{n-1}$, de nouveau un entier strictement positif. Ce nombre est divisé par au moins 4/3 à chaque échange aux étapes (3)(b)(i–ii), tout en restant entier strictement positif. Il ne peut donc y avoir qu'un nombre fini d'échanges lors d'une exécution de l'algorithme LLL. Considérons les nombres e d'échanges en (3)(b)(i) et v de passages en (3)(b) avec un test négatif effectués depuis le début d'une exécution de l'algorithme. En observant ces valeurs à chaque entrée de la boucle (3), on a l'invariant $i = 2 - e + v$. Chaque passage dans la boucle incrémente l'un ou l'autre de ces deux nombres. Mais comme le nombre e ne peut croître indéfiniment, le nombre v doit ultimement ne plus cesser de croître, et i avec, ce jusqu'à la valeur $i = n + 1$ qui provoque la fin de la boucle (3). L'algorithme termine donc.

5.3. Complexité arithmétique. Au début de l'algorithme, le nombre D qui vient d'être introduit pour montrer la terminaison vaut

$$D_0 = \|g_1^*\|^{2(n-1)}\|g_2^*\|^{2(n-2)} \dots \|g_{n-1}^*\|^2.$$

Mais chaque g_i^* étant alors une certaine projection de l'entrée f_i, on a

$$D_0 \le \|f_1\|^{2(n-1)}\|f_2\|^{2(n-2)} \dots \|f_{n-1}\|^2 \le A^{n(n-1)}.$$

Après e échanges en (3)(b)(i), on a l'encadrement $1 \le D \le (3/4)^e D_0$, d'où la borne supérieure $n(n-1)\log_{4/3} A$ sur e.

Notons e_{final} et v_{final} les valeurs finales de e et v. On vient de prouver la relation $e_{\text{final}} = O(n^2 \log A)$; par ailleurs l'invariant sur i donne $n+1 = 2 - e_{\text{final}} + v_{\text{final}}$, d'où $v_{\text{final}} = O(n^2 \log A)$. Comme l'orthogonalisation initiale se fait en $O(n^3)$ opérations arithmétiques et que chacune des étapes (3)(a) et (3)(b)(i–ii) se fait en $O(n^2)$ opérations arithmétiques, la complexité arithmétique totale de l'algorithme LLL est

$$O(n^3) + (e_{\text{final}} + v_{\text{final}})O(n^2) = O(n^4 \log A).$$

La borne sur la complexité en bits est réellement plus technique et n'est pas présentée ici. L'idée de la preuve est que les entiers utilisés dans l'algorithme ne dépassent pas la taille $O(n \log A)$. La borne annoncée n'est ensuite que le produit de la borne sur la complexité arithmétique par cette borne sur les entiers utilisés.

CHAPITRE 9

Intégration symbolique des fractions rationnelles

1. Le problème de l'intégration symbolique et le cas des fractions rationnelles

L'intégration symbolique en « forme close » est un succès des débuts du calcul formel. Avec l'algorithme d'intégration de Risch [3, 4], il est possible de déterminer si une expression rationnelle complexe en des fonctions obtenues par compositions de logarithmes et exponentielles admet une primitive de même nature, et dans ce cas la calculer. Par exemple, on calcule qu'une primitive de

$$f = \frac{x\left(\left(x^2e^{2x^2} - \ln^2(x+1)\right)^2 + 2xe^{3x^2}\left(x - (2x^2+1)(x+1)\ln(x+1)\right)\right)}{(x+1)\left(\ln^2(x+1) - x^2e^{2x^2}\right)^2}$$

est

$$F = x - \ln(x+1) + \frac{xe^{x^2}\ln(x+1)}{x^2e^{2x^2} - \ln(x+1)^2} + 1/2\ln\left(\ln(x+1) + xe^{x^2}\right) - 1/2\ln\left(\ln(x+1) - xe^{x^2}\right).$$

On démontre à l'inverse que e^{x^2} n'admet aucune écriture rationnelle en des fonctions obtenues par compositions de logarithmes et exponentielles. Notons que vérifier le résultat $\int f\,dx = F$ est d'une grande simplicité puisqu'il suffit de dériver et de normaliser les expressions. La difficulté est de déterminer F à partir de f, sans autre indication a priori sur les exponentielles et les logarithmes qui interviennent dans le résultat.

Un cadre simplifié auquel se ramène ultimement l'algorithme de Risch est celui de l'intégration des fractions rationnelles. De plus, il présente déjà un certain nombre des idées de l'algorithme de Risch général et une étude de sa complexité reste abordable à ce stade du cours. C'est pourquoi nous choisissons de le présenter bien que nous n'abordions pas l'algorithme général dans ce cours.

Du point de vue algébrique, une fraction rationnelle, disons, à coefficients dans un corps k, est un élément f/g de $k(X)$ où f et g sont des polynômes de $k[X]$ que l'on prendra premiers entre eux. D'un point de vue analytique, en faisant $k = \mathbb{C}$, une fraction rationnelle représente une fonction de $\mathbb{C}\cup\{\infty\}$ dans lui-même qui a la propriété de n'avoir qu'un nombre fini de points singuliers (où elle n'est pas développable en série entière) et de n'avoir que des singularités de type polaire. Les deux points de vue se rejoignent avec la conséquence que toute fraction rationnelle de $\mathbb{C}(X)$ peut se récrire de façon unique sous la forme

$$\frac{f}{g} = c_{r_\infty}^{(\infty)}X^{r_\infty} + \cdots + c_1^{(\infty)}X + \sum_{i=1}^{s}\left(\frac{c_{r_i}^{(i)}}{(X-\alpha_i)^{r_i}} + \cdots + \frac{c_1^{(i)}}{X-\alpha_i}\right) + c_0$$

pour des complexes uniquement déterminés α_i pour $1 \le i \le s$, et c_0 et $c_j^{(i)}$ pour $1 \le i \le s$ et $1 \le j \le 2i$.

De manière générale, le problème de l'intégration formelle, de même que sa généralisation à la résolution d'équation différentielles linéaires en solutions rationnelles, repose sur la compréhension de l'action de la dérivation sur les singularités des fonctions. Dans le cas qui nous occupe, la « difficulté » mathématique de l'intégration formelle repose sur le fait que l'intégration des pôles finis d'ordre 1 engendre des logarithmes.

En calcul formel, une difficulté supplémentaire est de pouvoir travailler avec des expressions rationnelles sur une extension algébrique finie k de $\mathbb{Q}$ telle que les polynômes de $k[X]$ ne se factorisent pas nécessairement en facteurs linéaires. En particulier, on s'intéresse aussi à fournir

des primitives exprimées lorsque cela est possible avec des coefficients de $\mathbb{Q}$ et d'éviter l'introduction de nombres complexes.

Dans ce qui suit, $M(n)$ représente la complexité arithmétique de la multiplication de deux polynômes de degrés au plus n et ω est l'exposant de la complexité arithmétique de l'algèbre linéaire pour des matrices de taille $n \times n$. Par exemple, $M(n) = O(n \ln n \ln\ln n)$ pour la multiplication par FFT ; par ailleurs, $2 \leq \omega \leq 3$, la borne supérieure correspondant au produit matriciel. L'objectif est ici de décrire un algorithme pour l'intégration d'une fraction rationnelle f/g pour des polynômes f et g de degrés bornés par n en complexité arithmétique $O\big(M(n)^2 \ln n\big)$, modulo le coût de factorisation d'un résultant de degré au plus n.

Dans tout ce chapitre, la notation $f \wedge g$ représente le p. g. c. d. unitaire des polynômes f et g. De plus, il est sous-entendu que les algorithmes à utiliser pour les évaluations de complexité sont les algorithmes optimaux en termes de nombres de multiplications polynomiales. Ainsi, pour la division euclidienne, on choisira l'algorithme du chapitre **??** de complexité $O\big(M(n)\big)$, pour l'algorithme d'Euclide (de base ou étendu), ce sera l'algorithme du chapitre **??** de complexité $O\big(M(n) \ln n\big)$, et ainsi de suite.

2. Intégration de la partie polynomiale

En considérant la décomposition en éléments simples présentée plus haut, une partie de la fraction rationnelle f/g à intégrer à un comportement simple, à savoir la « partie polynomiale » provenant du « pôle » à l'infini.

Algorithme (Intégration de la partie polynomiale).

Entrée : une primitive $\int f/g$ à calculer pour des polynômes f et g de degrés au plus n donnés par leur développements

Sortie : des polynômes Q et h, le premier de degré moindre que n, le second de degré strictement moindre que n, tels que $\int f/g = Q + \int h/g$

Complexité arithmétique : $O\big(M(n)\big)$

(1) calculer le quotient q et le reste h de la division euclidienne de f par g

(2) calculer $Q = q_d X^{d+1}/(d+1) + \cdots + q_0 X$ à partir de $q = q_d X^d + \cdots + q_0$

(3) renvoyer l'expression $Q + \int h/g$

Preuve de l'algorithme. La complexité de l'intégration de q étant linéaire en d, la complexité dominante est celui de la division euclidienne, en $O\big(M(n)\big)$. □

Pour la suite, l'intégration de h/g va reposer sur la détermination de deux fractions rationnelles c/d et a/b telles que $\int h/g = c/d + \int a/b$. La première fraction c/d est la « partie rationnelle » de l'intégrale $\int f/g$ initiale et l'intégrale $\int a/b$ qu'il restera à déterminer est sa « partie logarithmique », que nous allons réexprimer comme combinaison linéaire de logarithmes.

3. Partie sans carré et factorisation sans carré

Il est bien connu que tout polynôme unitaire f de $k[X]$ se factorise de façon unique (à permutation près des facteurs) sous la forme

$$f = f_1^{e_1} \dots f_r^{e_r}$$

pour des polynômes irréductibles unitaires f_i deux à deux premiers entre eux et des entiers strictement positifs e_i. En regroupant les facteurs de même exposant, on obtient une autre factorisation en termes de polynômes unitaires g_i deux à deux premiers entre eux, de la forme

$$f = g_1 g_2^2 \dots g_m^m.$$

Dans cette écriture, chacun des g_i est sans carré, au sens où aucun polynôme non constant n'a son carré qui divise g_i. Il en est de même du produit $g_1 \dots g_m$, ce qui motive la définition suivante.

Définition. En termes de la notation ci-dessus, le polynôme $f_1 \dots f_r = g_1 \dots g_m$ est appelé *partie sans carré* du polynôme unitaire f et l'expression $g_1 g_2^2 \dots g_m^m$ est appelée *factorisation sans carré* de f.

Notons que cet ingrédient intervient aussi dans les algorithmes classiques de factorisation de polynômes à coefficients entiers. Nous donnons maintenant un algorithme pour son calcul.

Si pour un polynôme g, une puissance g^i divise f, on trouve un polynôme h tel que $f = g^i h$. Par dérivation, on a l'égalité $f' = g^{i-1}(ig'h + gh')$. On voit donc que g^{i-1} divise f'. Ainsi, g^{i-1} divise le p. g. c. d. $f \wedge f'$ et il en est de même pour le produit $g_2 g_3^2 \dots g_m^{m-1}$.

Inversement, comme $f = g_i^i h$ pour un certain polynôme h premier avec g_i, $f' = g_i^{i-1}(ig_i'h + g_i h')$ est divisible par g_i^i si et seulement si g_i divise ig_i', d'où, par comparaison des degrés, si et seulement si ig_i' est nul. Pour exclure ce cas, on fera dorénavant l'hypothèse que le corps k est de caractéristique nulle. Donc, chaque g_i^{i-1}, mais aucun des g_i^i, divise $f \wedge f'$ et finalement $g_2 g_3^2 \dots g_m^{m-1}$ divise $f \wedge f'$.

Ainsi, puisque $g_2 g_3^2 \dots g_m^{m-1}$ et $f \wedge f'$ sont unitaires, ils sont égaux et égaux à $f_1^{e_1 - 1} \dots f_r^{e_r - 1}$. Cette égalité donne l'idée de l'algorithme qui suit.

Algorithme (Partie sans carré d'un polynôme).

Entrée : un polynôme unitaire f de degré n

Sortie : la partie sans carré de f

Complexité arithmétique : $O\big(M(n) \ln n\big)$

(1) calculer le p. g. c. d. $u = f \wedge f'$ par l'algorithme d'Euclide

(2) calculer le quotient de la division f/u et le renvoyer

Preuve de l'algorithme. La complexité de l'algorithme d'Euclide est de $O\big(M(n) \ln n\big)$ et domine celle de la division, $O\big(M(n)\big)$. □

Avec les mêmes notations que ci-dessus, la première étape de l'algorithme calcule $g_2 g_3^2 \dots g_m^{m-1}$. En réitérant le calcul sur ce résultat intermédiaire, on obtient $g_3 g_4^2 \dots g_m^{m-2}$, et ainsi de suite jusqu'à obtenir g_m. Ceci donne un algorithme de complexité presque optimale pour le calcul d'une factorisation sans carré.

Algorithme (Factorisation sans carré d'un polynôme).

Entrée : un polynôme unitaire f de degré n

Sortie : la factorisation sans carré de f, $f = g_1 g_2^2 \dots g_m^m$

Complexité arithmétique : $O\big(mM(n) \ln n\big)$

(1) poser $u_0 = f$

(2) en partant de $i = 1$, calculer $u_i = u_{i-1} \wedge u_{i-1}'$ par l'algorithme d'Euclide et les quotients $v_i = u_{i-1}/u_i$, ce jusqu'à trouver $u_i = 1$

(3) appeler m la dernière valeur de i pour laquelle u_i est autre que 1

(4) pour i de 1 à $m - 1$, calculer $h_i = v_i/v_{i+1}$ et poser $h_m = v_m$

(5) renvoyer la factorisation $h_1 h_2^2 \dots h_m^m$

Preuve de l'algorithme. On a immédiatement les relations $u_i = g_{i+1} g_{i+2}^2 \dots g_m^{m-i}$, $v_i = g_i \dots g_m$ et $h_i = g_i$. La complexité reste encore dominée par celle de l'algorithme d'Euclide. Le nombre d'étapes de la boucle est m, d'où la complexité annoncée. □

Un autre algorithme (non présenté en cours), l'algorithme de Yun [6], effectue cette factorisation en complexité arithmétique $O\big(M(n)\ln n\big)$, donc sans le facteur arithmétique m.

4. Décomposition en fractions partielles relativement à une factorisation sans carré

La détermination des parties rationnelle et logarithmique d'une intégrale $\int h/f$ où le polynôme h est de degré strictement plus petit que celui du polynôme unitaire f s'obtient après avoir décomposé la fraction h/f en « fractions partielles » relativement à la factorisation sans carré de f. L'idée de la décomposition en fractions partielles, *partial fractions* en anglais, est une généralisation de la décomposition en éléments simples qui ne sépare pas les facteurs irréductibles du dénominateur lorsqu'ils peuvent être traités en parallèle.

En repartant de la notation $f = f_1^{e_1}\dots f_r^{e_r} = g_1 g_2^2 \dots g_m^m$, on cherche d'abord à écrire la fraction h/f sous la forme

$$\frac{h}{f} = \frac{p_1}{g_1} + \frac{p_2}{g_2^2} + \dots + \frac{p_m}{g_m^m} \tag{8}$$

où chaque polynôme p_i a degré strictement inférieur à celui de g_i^i, avant de scinder chaque p_i/g_i^i sous la forme

$$\frac{p_i}{g_i^i} = \frac{p_{i,i}}{g_i^i} + \dots + \frac{p_{i,1}}{g_i}$$

pour des polynômes $p_{i,j}$ de degré strictement inférieur à celui de g_i. Notons encore que les g_i ne sont pas nécessairement irréductibles.

Après multiplication par f, la relation cherchée (8) devient

$$h = p_1 \widehat{g_1} g_2^2 \dots g_m^m + \dots + p_i g_1 \dots \widehat{g_i^i} \dots g_m^m + \dots p_m g_1 \dots g_{m-1}^{m-1} \widehat{g_m^m},$$

où le chapeau représente un facteur omis. Pour chaque i, le facteur g_i^i intervient en facteur de tous les termes de la somme sauf un, d'où les relations de congruence

$$h \equiv p_i g_1 \dots \widehat{g_i^i} \dots g_m^m \mod g_i^i.$$

Les polynômes unitaires g_j^j étant deux à deux premiers entre eux, chaque g_j^j pour $j \neq i$ est inversible modulo g_i^i et on a la nouvelle congruence

$$p_i \equiv h g_1^{-1} \dots \widehat{g_i^{-i}} \dots g_m^{-m} \mod g_i^i.$$

Les polynômes p_i sont donc uniquement déterminés par la contrainte de degré annoncée plus haut.

Quant au passage de p_i aux $p_{i,j}$, une approche possible consiste à poser $r_i = p_i$, puis à calculer les divisions euclidiennes $r_j = r_{j-1} g_i + p_{i,j}$ pour j de i à 2. On a alors $p_i = p_{i,1} g_i^{i-1} + \dots + p_{i,i}$, d'où l'expression désirée pour p_i/g_i^i.

On obtient ainsi l'algorithme suivant.

ALGORITHME (Décomposition en fractions partielles relativement à une factorisation sans carré).

ENTRÉE : une fraction rationnelle h/f pour un polynôme f de degré n et un polynôme h de degré strictement inférieur à n, ainsi que la factorisation sans carré $f = g_1 g_2^2 \dots g_m^m$

SORTIE : des polynômes $p_{i,j}$ pour $1 \le j \le i \le m$ avec chaque $p_{i,j}$ de degré strictement moindre que g_i, et tels que $h/f = \sum_{i=1}^{m} \sum_{j=1}^{i} p_{i,j}/g_i^j$

COMPLEXITÉ ARITHMÉTIQUE : $O\big(mM(n)\ln n\big)$

(1) pour i de 1 à m
   (a) calculer $u_i = f/g_i^i$
   (b) calculer une relation de Bézout $a_i u_i + b_i g_i^i = 1$ par l'algorithme d'Euclide étendu
   (c) calculer le reste p_i du produit ha_i modulo g_i^i
   (d) poser $r_i = p_i$, puis effectuer les divisions $r_j = r_{j-1} g_i + p_{i,j}$ pour j de i à 2

(2) renvoyer les $p_{i,j}$ pour $1 \le j \le i \le m$

PREUVE DE L'ALGORITHME. Pour un i fixé à l'étape (1)(a), la fraction h/f se récrit sous la forme suivante, où le degré de p_i est strictement inférieur à celui de g_i^i :

$$\frac{h}{f} = \frac{p_i}{g_i^i} + \frac{q_i}{u_i}.$$

La multiplication par $f = u_i g_i^i$ fournit une relation entre polynômes,

$$h = p_i u_i + q_i g_i^i.$$

Avec les cofacteurs obtenus à l'étape (1)(b), on déduit la relation

$$ha_i = p_i + (a_i q_i - b_i p_i) g_i^i$$

qui, au vu des degrés de p_i et g_i^i, justifie la définition de p_i à l'étape (1)(c).

Connaissant h, f et les g_i^i, la détermination de chaque p_i par le calcul qui vient d'être esquissé fait une division exacte de polynômes de degrés au plus n, celle de f par g_i^i, une inversion de polynômes de degré au plus n (par l'algorithme d'Euclide étendu), une multiplication en degrés au plus n et calcul de reste de division de polynômes de degré au plus $2n$ par des polynômes de degré au plus n. D'où une complexité totale pour le calcul des p_i en $O\big(mM(2n)\ln(2n)\big)$, qui est aussi $O\big(mM(n)\ln n\big)$. Reste à extraire les $p_{i,j}$ à partir des p_i. Pour chaque i, la complexité arithmétique est en $O\big(iM(\deg g_i)\big) + O\big((i-1)M(\deg g_i)\big) + \dots + O\big(M(\deg g_i)\big)$, soit en $O\big(i^2 M(\deg g_i)\big)$. En sommant sur i, on obtient, par sous-additivité de M, la complexité

$$\sum_{i=1}^{m} O\big(i^2 M(\deg g_i)\big) \le O\left(mM\left(\sum_{i=1}^{m} i \deg g_i\right)\right) \le O\big(mM(n)\big),$$

complexité qui reste dominée par celle du reste de l'algorithme. □

Indiquons une autre approche, récursive, pour le calcul des $p_{i,j}$ à partir des p_i, qui donnera une complexité en $O\big(M(n)\ln n\big)$. Par simplicité, donnons l'idée pour f de la forme g_m^m avec $m = 2^k$ une puissance de 2, et notons $d = \deg g$, si bien que $n = 2^k d$. On effectue d'abord une division euclidienne en taille $n/2$, celle de p_m par $g^{m/2}$, puis deux en taille $n/4$, celles du quotient et du reeste par $g^{m/4}$, puis quatre en taille $n/8$, celles des deux quotients et des deux restes obtenus, et ainsi de suite. La complexité obtenue est alors $O\big(M(n/2)\big) + 2O\big(M(n/4)\big) + \dots + 2^{k-1} O\big(M(n/2^k)\big)$, qui par sous-additivité de M est majorée par $O\big(M(n/2)\ln n\big)$. Par ailleurs, le calcul de g, g^2, g^4, …,

g^m coûte $O\big(M(n)\big)$ opérations arithmétiques, encore une fois par sous-additivité de M, d'où la complexité globale annoncée.

5. Intégration de la partie rationnelle

À ce stade, on a ramené, en complexité essentiellement linéaire, le problème de l'intégration d'une fraction rationnelle sous forme normale à celui de l'intégration d'une expression de la forme

$$\sum_{i=1}^{m}\sum_{j=1}^{i}\frac{p_{i,j}}{g_i^j}$$

pour des polynômes $p_{i,j}$ de degré strictement plus petit que celui de g_i.

Considérons un instant un terme de la somme, $p_{i,j}/g_i^j$. Chaque g_i étant sans carré, on a une relation de Bézout,

$$s_i g_i' + t_i g_i = 1$$

pour des polynômes s_i et t_i. En multipliant cette relation par $p_{i,j}$, on déduit la relation

$$\int\frac{p_{i,j}}{g_i^j}=\int\frac{-s_ip_{i,j}}{j-1}\cdot\frac{-(j-1)g_i'}{g_i^j}+\int\frac{t_ip_{i,j}}{g_i^{j-1}}=\frac{-s_ip_{i,j}}{(j-1)g_i^{j-1}}+\int\frac{t_ip_{i,j}+(s_ip_{i,j})'/(j-1)}{g_i^{j-1}}$$

par intégration par parties. On a ainsi ramené l'intégration d'un pôle d'ordre j à celle d'un pôle d'ordre $j-1$. Ceci suggère une méthode itérative pour l'intégration de la partie rationnelle, en commençant par les ordres maximaux ; c'est le procédé de réduction de Hermite. Un ingrédient masqué par le calcul prédédent est que le produit $s_ip_{i,j}$ peut ne pas être réduit modulo g_i.

Algorithme (Intégration de la partie rationnelle par réduction de Hermite).

Entrée : une fraction rationnelle h/f avec f de degré n et h de degré strictement inférieur à n, ainsi que la décomposition $\sum_{i=1}^{m}\sum_{j=1}^{i}\frac{p_{i,j}}{g_i^j}$ en fractions partielles relativement à la factorisation sans carré $f=g_1g_2^2\dots g_m^m$

Sortie : des familles de polynômes (a_i) et (b_i) pour $1\le i\le m$, et (c_i) et (d_i) pour $1\le i\le\ell$ vérifiant la relation $\int h/f=\sum_i c_i/d_i+\int\sum_i a_i/b_i$ et tels que les b_i divisent la partie sans carré $g_1\dots g_m$ et les d_i divisent $g_2\dots g_m^{m-1}$

Complexité arithmétique : $O\big(M(n)\ln n\big)$

(1) faire $u=0$ et $\ell=0$

(2) pour i de 1 à m

  (a) par l'algorithme d'Euclide étendu, calculer une relation de Bézout $s_ig_i'+t_ig_i=1$ pour des polynômes s_i et t_i de degré strictement inférieur à celui de g_i

  (b) pour j de i à 2

    (i) effectuer la division euclidienne de $s_ip_{i,j}$ par g_i pour trouver $s_ip_{i,j}=qg_i+\bar{s}$ avec un reste $\bar{s}$ de degré strictement inférieur à g_i

    (ii) poser $\bar{t}=t_ip_{i,j}+qg_i'$

    (iii) ajouter $\bar{t}+\bar{s}'/(j-1)$ à $p_{i,j-1}$ et normaliser

    (iv) incrémenter ℓ avant de faire $c_\ell=-\bar{s}/(j-1)$ et $d_\ell=g_i^{j-1}$

(3) poser $a_i=p_{i,1}$ et $b_i=g_i$ pour $1\le i\le m$

(4) renvoyer les familles (a_i), (b_i), (c_i) et (d_i)

Preuve de l'algorithme. La correction de l'algorithme découle immédiatement de la remarque précédent l'énoncé de l'algorithme et des observations que les d_t sont par construction de la forme g_i^{j-1}, divisant ainsi g_i^{i-1}, et que chaque b_i vaut en fait g_i.

Appelons δ_i le degré de g_i. Pour chaque i, le calcul de s_i et t_i a lieu en $O\big(M(\delta_i)\ln\delta_i\big)$ opérations arithmétiques. Pour chaque étape de la réduction de Hermite, le calcul de $\bar{s}$ et $\bar{t}$ se fait en complexité $O\big(M(\delta_i)\big)$, la mise à jour de $p_{i,j-1}$ en $O(\delta_i)$. En effet, $\bar{s}$ est par construction de degré inférieur à δ_i ; puisqu'on a la relation $\bar{s}g_i' + \bar{t}g = p_{i,j}$, par comparaison des degrés on déduit que $\bar{t}$ est aussi de degré inférieur à δ_i. La complexité totale est donc, par sous-additivité de M,

$$\sum_{i=1}^{m} O\big(M(\delta_i)\ln\delta_i\big) + \sum_{i=1}^{m} O\big(iM(\delta_i)\big) \le O\left(M\Big(\sum_{i=1}^{m}\delta_i\Big)\right)\ln n + O\left(M\Big(\sum_{i=1}^{m} i\delta_i\Big)\right) \le O\big(M(n)\ln n\big).$$

□

Notons que l'algorithme tel qu'il est donné ci-dessus ne renvoie pas un résultat sous la forme $c/d + \int a/b$, pour deux fractions rationnelles a/b et c/d sous forme normalisée, mais donne plutôt ces deux fractions rationnelles sous forme de sommes d'expressions rationnelles. Dans les systèmes de calcul formel, cette mise sous forme normale est laissée à l'initiative de l'utilisateur.

Prenons l'exemple du calcul de

$$\int \frac{1}{(2X^2+1)^2}.$$

Ici, nous avons simplement $h = 1/4$ et $f = g_2^2$ pour $g_2 = X^2 + 1/2$. Comme $g_2' = 2X$, nous trouvons la relation de Bézout $-Xg_2' + 2g_2 = 1$, d'où la réduction

$$\int \frac{1}{(2X^2+1)^2} = \int \frac{X}{4}\frac{-2X}{(X^2+1/2)^2} + \int \frac{1/2}{X^2+1/2} = \frac{X}{2(2X^2+1)} + \int \frac{1}{2(2X^2+1)}.$$

La structure des b_ℓ et celle des d_ℓ suggèrent de rechercher une primitive de la partie rationnelle directement sous la forme $c/d + \int a/b$. En suivant le calcul des a_ℓ et des c_ℓ au cours de l'algorithme précédent, il apparaît que les polynômes a et c ont des degrés strictement inférieurs à $d_1 + \cdots + d_m$ et $d_2 + 2d_3 + \cdots + (m-1)d_m$, respectivement, d'où l'algorithme suivant, par coefficients indéterminés, dû à Horowitz en 1971. Cette méthode est de moins bonne complexité que la méthode de Hermite, mais est bien pratique pour des fractions de petite taille si l'on ne dispose pas du temps pour bien programmer la méthode de Hermite.

Algorithme (Intégration de la partie rationnelle par l'algorithme de Horowitz).

Entrée : une fraction rationnelle h/f avec f de degré n et h de degré strictement inférieur à n, ainsi que la factorisation sans carré $f = g_1 g_2^2 \dots g_m^m$

Sortie : des polynômes a et c vérifiant la relation $\int h/f = c/(g_2 \dots g_m^{m-1}) + \int a/(g_1 \dots g_m)$

Complexité arithmétique : $O(n^\omega)$

(1) poser $\delta = d_1 + \cdots + d_m$ et écrire $a = a_0 + \cdots + a_{\delta-1}X^{\delta-1}$ et $c = c_0 + \cdots + c_{n-\delta-1}X^{n-\delta-1}$ pour des coefficients indéterminés

(2) écrire $h = c'g_1 \dots g_m - c\sum_{i=2}^{m}(i-1)g_1 \dots g_i' \dots g_m + ag_2 \dots g_m^{m-1}$ et égaler les coefficients en X à zéro

(3) résoudre le système linéaire obtenu

(4) substituer dans a et c pour renvoyer le résultat

Preuve de l'algorithme. La relation polynomiale servant à déterminer le système linéaire s'obtient en dérivant $\int h/f = c/(g_2 \dots g_m^{m-1}) + \int a/(g_1 \dots g_m)$ et en chassant les dénominateurs. Le coût des multiplications de polynômes est dominé par celui de l'algèbre linéaire sur le système qui est de dimension $n \times (n-2)$. □

Reprenons l'exemple de l'intégration de $1/(2X^2+1)^2$, avec toujours $h = 1/4$, $f = g_2^2$ pour $g_2 = X^2 + 1/2$ et $g_2' = 2X$. La mise en équation par la méthode d'Horowitz donne

$$1/4 = c'(X^2 + 1/2) - c(2X) + a(X^2 + 1/2)$$

avec $a = a_0 + a_1 X$ et $c = c_0 + c_1 X$ vu que $\delta = 2$. Le système linéaire se résoud par $a_0 = 1/2$, $a_1 = 0$, $c_0 = 0$, $c_1 = 1/4$, ce qui redonne la primitive déjà calculée par la méthode de Hermite.

6. Algèbre différentielle pour l'intégration rationnelle

Que ce soit par la méthode de Hermite ou par celle d'Horowitz, on a ramené l'intégration d'une fraction rationnelle générale à celle d'une expression rationnelle de la forme a/b pour b sans carré et a de degré strictement inférieur à b. L'exemple le plus simple est maintenant celui de l'intégration de la fraction $1/X$ et chacun s'attend dès lors à ce que l'intégration rationnelle introduise des logarithmes. Comme on l'a déjà annoncé, on se dote d'un cadre algébrique pour évacuer les questions de déterminations multiples des logarithmes. Cette section introduit la terminologie d'algèbre différentielle utile pour notre problème.

Les définitions qui suivent soulignent le fait qu'il est possible de considérer l'opération de dérivation sans référence à aucune variable.

Définition. Une *dérivation* d'un anneau A est une application $a \mapsto a'$ de A dans lui-même qui est additive, au sens où $(a+b)' = a' + b'$ pour tous éléments a et b de A, et qui vérifie la règle de Leibniz $(ab)' = a'b + ab'$. L'anneau A est alors appelé un *anneau différentiel*. Si l'anneau A est un corps, il est dit *corps différentiel*. S'il est une algèbre sur un corps commutatif k et si la dérivation est l'application nulle sur k, l'anneau A est dit une *k-algèbre différentielle*.

Étant donné un corps commutatif k, nous allons donner plusieurs exemples de structures de k-algèbres différentielles sur l'anneau des polynômes $A = k[X]$. Notons d'abord la relation $f(X)' = D(f)(X)X'$ pour tout polynôme f de dérivée formelle usuelle $D(f)$, que l'on montre aisément d'abord sur les monômes X^i par récurrence sur le degré i, puis que l'on étend par linéarité. Pour $X' = 1$, on obtient la k-algèbre différentielle avec pour dérivation la dérivation formelle usuelle des polynômes. Pour $X' = 0$, on obtient la dérivation triviale, nulle. Pour $X' = X$, on obtient une dérivation sur $A = k[X]$ qui donne une structure différentielle isomorphe à celle de la k-algèbre $k[e^\theta]$ avec dérivation par rapport à la variable θ, par l'isomorphisme d'algèbres valant e^θ sur X.

Notons encore qu'il existe une unique extension d'une structure d'anneau différentiel sur $k[X]$ en une structure de corps différentiel sur $k(X)$. En effet, on a nécessairement, pour tout a de A, $0 = 1' = (aa^{-1})' = a'a^{-1} + a(a^{-1})'$, d'où $(a^{-1})' = -a^{-2}a'$.

Les définitions qui suivent donnent un cadre pour étendre le corps des fractions rationnelles par des objets algébriques qui se comportent algébriquement comme des logarithmes.

Définition. Un élément a d'un anneau différentiel A est *logarithmique* s'il existe b de A tel que $a' = b'/b$. On note alors $a = \ln b$. Un anneau différentiel B est une *extension différentielle* d'un anneau différentielle A s'il contient l'anneau A et si la dérivation de B agit sur les éléments de A comme la dérivation de A. Dans le cas où A est un corps et où B peut être vu comme le corps $A(\theta)$ pour un élément a de A tel que $\theta' = a'/a$, l'extension est dite *logarithmique*.

7. Intégration de la partie logarithmique

L'intégration de la partie logarithmique sans passer à une clôture algébrique $\bar{k}$ du corps k des coefficients de la fraction rationnelle s'appuie sur le théorème suivant, qui donne un critère pour décider si une primitive peut s'écrire avec des coefficients dans une extension algébrique fixée K de k. Dans la pratique, on a le plus souvent $k = \mathbb{Q}$ et on recherche des primitives à coefficients dans une extension algébrique réelle finie K de $\mathbb{Q}$, en d'autres termes qui vérifie $\mathbb{Q} \subset K \subset \bar{\mathbb{Q}} \cap \mathbb{R}$.

Le théorème ci-dessous décrit précisément l'extension algébrique minimale qui permet d'écrire une primitive de la partie logarithmique. Son idée est la suivante. Supposons que b se factorise sous la forme $b = v_1 \dots v_\ell$ pour des polynômes deux à deux irréductibles et qui fournissent une

primitive de a/b de la forme 9. Alors, après dérivation et multiplication par b, nous obtenons la relation entre polynômes

$$a = \sum_{i=1}^{\ell} c_i v_1 \dots \widehat{v_i} \dots v_\ell v_i'.$$

Pour un i fixé, le facteur v_i apparaît dans tous les termes de la somme sauf un. Comme les v_j sont deux à deux premiers entre eux et sans carrés, v_i divise $a - c_i v_1 \dots \widehat{v_i} \dots v_\ell v_i'$. Après l'ajout des termes manquant pour faire apparaître b', on voit qu'il divise aussi $a - c_i b'$. Une autre interprétation de cette divisibilité est que les polynômes b et $a - b'Y$, pour une nouvelle indéterminée Y, ont un zéro commun pour chaque spécialisation de Y à l'un des c_i : tout zéro de v_i (pour le même i). Ce point de vue explique l'introduction de résultants dans le résultat qui suit.

Théorème. Soit une fraction rationnelle a/b de $k(X)$, avec $a \wedge b = 1$ pour b unitaire, de degré n et sans carré et a de degré strictement inférieur à n. Soit encore une extension algébrique finie K de k. Considérons des scalaires c_i de K et des polynômes v_i de $K[X]$ deux à deux premiers entre eux. Alors, une primitive de a/b est donnée par

$$\int \frac{a}{b} = \sum_{i=1}^{\ell} c_i \ln v_i \tag{9}$$

si et seulement si le résultant $R(Y) = \operatorname{res}_X(b, a - b'Y)$ se factorise en termes linéaires en Y dont les c_i sont les zéros et tels que $v_i = b \wedge (a - b'c_i)$.

Avant de donner la preuve, faisons quelques rappels sur le résultant. Pour un polynôme $f = f_0 + \dots + f_n X^n$ de degré n et un polynôme $g = g_0 + \dots + g_m X^m$ de degré m, le résultant $\operatorname{res}_X(f, g)$ est donné comme le déterminant de la matrice suivante de taille $(n+m) \times (n+m)$, où toutes les entrées en dehors des deux parallèlogrammes sont nulles, et dite « matrice de Sylvester »,

$$\begin{pmatrix}
f_n & & & & g_m & & & \\
f_{n-1} & f_n & & & g_{m-1} & g_m & & \\
\vdots & \vdots & \ddots & & \vdots & \vdots & \ddots & \\
\vdots & \vdots & & & \vdots & \vdots & & \\
 & \vdots & & f_n & \vdots & \vdots & & g_m \\
 & \vdots & & \vdots & \vdots & \vdots & & \vdots \\
f_0 & \vdots & & \vdots & g_0 & \vdots & & \vdots \\
 & f_0 & & \vdots & & g_0 & & \vdots \\
 & & \ddots & f_1 & & & \ddots & g_1 \\
 & & & f_0 & & & & g_0
\end{pmatrix},$$

où les m premières colonnes contiennent des coefficients de f et les n dernières des coefficients de g. Cette matrice est, sur des bases monomiales appropriées, la matrice de l'application k-linéaire du produit cartésien $k[X]_{<m} \times k[X]_{<n}$ dans $k[X]$ qui au couple (u, v) associe la combinaison $uf + vg$. La définition par un déterminant donne une notion de résultant même dans le cas de polynômes à coefficients dans un anneau au lieu d'un corps k. En outre, pour deux polynômes scindés $f = u(X - a_1) \dots (X - a_d)$ et $g = v(X - b_1) \dots (X - b_e)$, on a les réexpressions suivantes du résultant, sur lesquelles on voit que le résultant est nul si et seulement si les deux polynômes ont un zéro commun :

$$\operatorname{res}_X(f, g) = (-1)^{de} v^d \prod_{g(\alpha)=0} f(\alpha) = u^e \prod_{f(\alpha)=0} g(\alpha) = u^e v^d \prod_{i,j} (a_i - b_j).$$

En conséquence de la définition, pour des polynômes a et b de deux indéterminées X et Y vérifiant la condition technique que l'un au moins soit unitaire en tant que polynôme en X, le résultant $\operatorname{res}_X\big(a(X,Y), b(X,Y)\big)$ s'annule en $Y = c$ si et seulement si les spécialisations $a(X, c)$ et $b(X, c)$ ont un p. g. c. d. non trivial dans $k[X]$.

Démonstration. Pour montrer l'implication directe, supposons $\int a/b = \sum_{i=1}^{\ell} c_i \ln v_i$, qui par dérivation conduit à $a/b = \sum_{i=1}^{\ell} c_i v_i'/v_i$, puis, après avoir multiplié par le produit convenable,

à $av_1 \dots v_\ell = b \sum_{i=1}^{\ell} c_i v_1 \dots v_i' \dots v_\ell$. Comme les v_i sont sans carré et deux à deux premiers entre eux, par réduction modulo v_i on trouve que chaque v_i divise b, donc que le produit $v_1 \dots v_\ell$ divise b. Comme par ailleurs a et b sont premiers entre eux, b divise le produit $v_1 \dots v_\ell$. Ces deux polynômes étant unitaires, ils sont égaux. On a donc les égalités $a = \sum_{i=1}^{\ell} c_i v_1 \dots v_i' \dots v_\ell = \big(\sum_{i=1}^{\ell} c_i v_i'/v_i\big) b$, d'où il vient $a - b'Y = \sum_{i=1}^{\ell} (c_i - Y) v_1 \dots v_i' \dots v_\ell$. Ainsi, le p. g. c. d. $b \wedge (a - b'Y)$ est non trivial si et seulement si $Y = c_i$ pour l'un des i et il vaut v_i pour le même i dans ce cas. Le résultant R est non nul sous ces mêmes conditions équivalentes. On a donc une factorisation de R en facteurs linéaires de la forme $Y - c_i$.

Pour l'implication inverse, considérons une clôture algébrique $\bar{K}$ de K et notons d'abord que pour tout c de $\bar{K}$, le résultant R s'annule en c si et seulement s'il existe un zéro μ de b dans $\bar{K}$ réalisant l'égalité $a(\mu) = b'(\mu)c$. Dans ce cas, puisque b est sans carré, $b'(\mu)$ ne peut être nul et on a $c = a(\mu)/b'(\mu)$. Introduisons l'application ϕ de $b^{-1}(0)$ dans $R^{-1}(0)$ qui à un zéro μ de b associe $a(\mu)/b'(\mu)$, un zéro de R. Par l'équivalence qui vient d'être donnée, cette application ϕ est bien définie et surjective. Notons c_1, …, c_ℓ les zéros distincts de R. On déduit alors l'égalité

$$\Pi_{\mu \in \phi^{-1}(c_i)} (X - \mu) = b \wedge (a - b'c_i) = v_i,$$

puis par produit sur i,

$$b = \Pi_{\mu \in b^{-1}(0)} (X - \mu) = v_1 \dots v_\ell.$$

On a donc pour tout i et tout μ dans $\phi^{-1}(c_i) = v_i^{-1}(0)$ les égalités

$$\left(\left(\sum_{j=1}^{\ell} c_j \frac{v_j'}{v_j} \right) b \right)(\mu) = c_i v_1(\mu) \dots v_i'(\mu) \dots v_\ell(\mu) = c_i b'(\mu) = a(\mu).$$

Les polynômes $\big(\sum_{i=1}^{\ell} c_i v_i'/v_i\big) b$ et a sont de degrés strictement inférieurs à n et coïncident sur les n racines distinctes de b ; ils sont donc égaux. Ainsi, une primitive de a/b est la combinaison linéaire de logarithmes $\sum_{i=1}^{\ell} c_i \ln v_i$. □

Algorithme (Algorithme de Rothstein–Trager).

Entrée : une fraction rationnelle a/b de $k(X)$ sous forme réduite, avec b unitaire, sans carré, de degré n et a de degré strictement inférieur à n

Sortie : une extension algébrique finie K de k, des constantes c_i de K et des polynômes v_i de $K[X]$ tels que $\int a/b = c_1 \ln v_1 + \dots + c_\ell \ln v_\ell$

Complexité arithmétique : $O\big(M(n)^2 \ln n\big)$ plus le coût de la factorisation du résultant R ci-dessous, de degré au plus n

(1) calculer $R(Y) = \operatorname{res}_X(b, a - b'Y)$

(2) factoriser R en facteurs linéaires en introduisant une extension algébrique adéquate K de k

(3) appeler c_1, …, c_ℓ les zéros deux à deux distincts de R dans K

(4) pour chaque i, calculer $v_i = b \wedge (a - c_i b')$

(5) renvoyer une description de K, les c_i et les v_i

Preuve de l'algorithme. La correction de l'algorithme provient immédiatement du théorème de structure précédent. Pour une complexité optimale, le résultant calculé à l'étape 1 doit être obtenu par l'algorithme d'Euclide rapide. Les polynômes b et $a - b'Y$ sont de degré $O(n)$ en X, d'où une complexité arithmétique en $O\big(M(n) \ln n\big)$ opérations dans $k[Y]$. Les coefficients polynomiaux de $k[Y]$ ayant des degrés pouvant atteindre n, la complexité du calcul de ce résultant est donc de $O\big(M(n)^2 \ln n\big)$ opérations dans k. De la même façon, la complexité arithmétique sur k du calcul des v_i doit tenir compte du degré de l'extension algébrique K, potentiellement aussi grand que n, donnant encore une fois $O\big(M(n)^2 \ln n\big)$ opérations dans k. □

Considérons l'exemple de l'intégration de $1/(X^3+X)$. On a d'abord le calcul naïf suivant, qui passe par les complexes et la décomposition en éléments simples classique :

$$\int \frac{1}{X^3+X} = \int \left(\frac{1}{X} - \frac{1/2}{X-i} - \frac{1/2}{X+i} \right) = \ln X - \frac{1}{2}\ln(X-i) - \frac{1}{2}\ln(X+i) = \ln X - \frac{1}{2}\ln(X^2+1).$$

En suivant la méthode de Rothstein–Trager, on calcule d'abord le résultant

$$R(Y) = \operatorname{res}_X\big(X^3+X, 1-(3X^2+1)Y\big) = \begin{vmatrix} 1 & 0 & -3Y & 0 & 0 \\ 0 & 1 & 0 & -3Y & 0 \\ 1 & 0 & 1-Y & 0 & -3Y \\ 0 & 1 & 0 & 1-Y & 0 \\ 0 & 0 & 0 & 0 & 1-Y \end{vmatrix} = (2Y+1)^2(1-Y),$$

qui admet une factorisation en facteurs linéaires sur $\mathbb{Q}$ sans recours à aucune extension algébrique. On pose donc $c_1 = 1$ et $c_2 = -1/2$ et on calcule les p. g. c. d.

$$v_1 = (X^3+X) \wedge (-3X^2) = X, \qquad v_2 = (X^3+X) \wedge \left(\frac{3}{2} + \frac{3X^2}{2} \right) = X^2+1,$$

qui donnent la représentation finale suivante pour une primitive :

$$\int \frac{1}{X^3+X} = \ln X - \frac{1}{2}\ln(X^2+1).$$

Remarquons que la propriété pour une primitive de pouvoir être représentée sans extension algébrique est liée à toute la structure de la fraction rationnelle et non pas seulement aux coefficients avec lesquels des racines complexes conjuguées apparaissent. À titre de comparaison, l'intégration de la fraction $1/(X^2+1)$ par la méthode de Rothstein–Trager donne le résultant

$$R(Y) = \operatorname{res}_X\big(X^2+1, 1-2XY\big) = \begin{vmatrix} 1 & -2Y & 0 \\ 0 & 1 & -2Y \\ 1 & 0 & 1 \end{vmatrix} = 4Y^2+1,$$

qui cette fois-ci n'a de factorisation en facteurs linéaires qu'après passage à $K = \mathbb{Q}[i]$. On trouve $c_1 = i/2,\ c_2 = -i/2,\ v_1 = (X^2+1) \wedge (1-iX) = X+i,\ v_2 = (X^2+1) \wedge (1+iX) = X-i,$ d'où

$$\int \frac{1}{X^2+1} = \frac{i}{2}\ln(X-i) - \frac{i}{2}\ln(X+i).$$

Par la même méthode, on obtient qu'aucune primitive de $1/(2X^2+1)$ ne s'exprime sans passage aux complexes et la formule

$$\int \frac{1}{2X^2+1} = \frac{i}{2\sqrt{2}} \left(\ln\big(X + i/\sqrt{2}\,\big) - \ln\big(X - i/\sqrt{2}\,\big) \right).$$

On peut donc terminer l'exemple pris pour la méthode de Hermite en donnant l'intégrale

$$\int \frac{1}{(2X^2+1)^2} = \frac{X}{2(2X^2+1)} + \frac{i}{4\sqrt{2}} \left(\ln\big(X + i/\sqrt{2}\,\big) - \ln\big(X - i/\sqrt{2}\,\big) \right).$$

8. Complexité globale de l'intégration rationnelle

Pour finir, disons quelques mots sur la complexité globale de l'intégration rationnelle. La méthode de Hermite pour l'intégration de la partie rationnelle a complexité $O\big(M(n)\ln n\big)$, où n borne les degrés des numérateurs et dénominateurs en entrée, alors que la méthode de Rothstein–Trager pour l'intégration de la partie logarithmique a complexité $O\big(M(n')^2 \ln n'\big)$, où n' borne les degrés des numérateurs et dénominateurs de ses entrées. Il semble donc que la complexité globale soit dominée par celle de l'intégration de la partie logarithmique, mais il faut en réalité préciser le lien entre n et n'. Ce dernier nombre étant le degré du dénominateur de la fraction a/b en sortie de la méthode de Hermite, deux cas extrêmes sont possibles : soit l'intégration d'une fraction h/f aboutit à une partie polynomiale a/b avec n' de l'ordre de n, et c'est alors la factorisation du résultant dans l'intégration de la partie logarithmique qui prédomine, avec une complexité arithmétique polynomiale en n ; soit l'intégration aboutit à une partie polynomiale donnant un n'

significativement plus petit que n, ce qui est le cas quand la factorisation sans carré de f fait intervenir un g_m^m pour un m grand, et la complexité globale reste dominée par le coût de la méthode de Hermite.

EXERCICE. Donner les complexités que les algorithme de ce chapitre auraient les algorithmes sur lesquels ils se basent étaient choisis naïfs.

Bibliographie

[A] BRONSTEIN, M. *Symbolic integration. I*, vol. 1 of *Algorithms and Computation in Mathematics*. Springer-Verlag, Berlin, 1997. Transcendental functions, With a foreword by B. F. Caviness.

[B] GEDDES, K. O., CZAPOR, S. R., AND LABAHN, G. *Algorithms for computer algebra*. Kluwer Academic Publishers, Boston, MA, 1992.

[C] RISCH, R. H. The problem of integration in finite terms. *Trans. Amer. Math. Soc. 139* (1969), 167–189.

[D] RISCH, R. H. The solution of the problem of integration in finite terms. *Bull. Amer. Math. Soc. 76* (1970), 605–608.

[E] VON ZUR GATHEN, J., AND GERHARD, J. *Modern computer algebra*. Cambridge University Press, New York, 1999.

[F] YUN, D. Y. Y. Fast algorithm for rational function integration. In *Information processing 77 (Proc. IFIP Congr., Toronto, Ont., 1977)*. North-Holland, Amsterdam, 1977, pp. 493–498. IFIP Congr. Ser., Vol. 7.

Ce chapitre s'appuie en particulier sur les sections 5.11 et 14.6 et le chapitre 22 de [5]. Le même sujet est traité au chapitre 11 de [2], en n'abordant cependant absolument pas les questions de complexité ; le prolongement de ces méthodes au cadre des fonctions liouviliennes par l'algorithme de Risch y est fait au chapitre 12 de [2]. Le livre [1] est à l'heure actuelle la référence la plus aboutie sur le sujet.

CHAPITRE 10

Sommation et intégration symboliques des fonctions spéciales

1. Expression de la création télescopique en termes d'algèbres de Ore rationnelles

On a déjà exposé dans ce cours la méthode de la création télescopique, en l'appliquant à la sommation hypergéométrique définie par une combinaison de l'algorithme de Gosper et d'une idée due à Zeilberger. Cette approche se généralise en des algorithmes de sommation et intégration pour les suites et fonctions ∂-finies.

Rappelons le principe de la méthode. Soit à évaluer une somme paramétrée

$$F_n = \sum_{k=a}^{b} f_{n,k}.$$

En toute généralité, le principe de la création télescopique est de déterminer une suite auxiliaire $g = (g_{n,k})$ ainsi que des coefficients $\eta_0, \ldots, \eta_r$, fonctions de la variable n, tels que se trouve vérifiée la relation

$$\eta_r(n) f_{n+r,k} + \cdots + \eta_0(n) f_{n,k} = g_{n,k+1} - g_{n,k}.$$

Ici, nous ne faisons pas plus d'hypothèses sur les η_i et g que celle de pouvoir évaluer la relation ci-dessus pour tout n quand k décrit les entiers de a à b. Dans ce cas, une sommation sur k fournit l'égalité

$$\eta_r(n) F_{n+r} + \cdots + \eta_0(n) F_n = g_{n,b+1} - g_{n,a}.$$

Ensuite, soit le membre de droite est naturellement nul, soit on recherche un opérateur annulateur de ce second membre ; par composition, on obtient une récurrence homogène sur F.

Des algorithmes d'efficacité différentes ont été donnés selon le domaine de recherche des η_i et de g, et selon le compromis choisi entre efficacité et richesse de la classe de suites f en entrée. En particulier, l'algorithme de Zeilberger, optimisé pour une suite f hypergéométrique, revient à rechercher des η_i polynomiaux et une suite g similaire à f, c'est-à-dire un multiple ϕf pour une fraction rationnelle ϕ en n et k. La suite $g = \phi f$ devant être une somme indéfinie, la recherche de ϕ et des η_i se fait par une variante paramétrée de l'algorithme de Gosper. Notons que le domaine de recherche de g est l'espace vectoriel $\mathbb{C}(n,k)f$, qui n'est autre, dans le cas hypergéométrique, que le module engendré par f sur l'algèbre de Ore $A = \mathbb{C}(n,k)\langle S_n, S_k; \sigma_n, \sigma_k, 0, 0\rangle$, pour des opérateurs de décalages σ_n et σ_k relatifs à n et k. Nous considérons ici la généralisation au cas où f est une fonction ∂-finie et où le module $A \cdot f$ est un espace vectoriel de dimension finie sur $\mathbb{C}(n,k)$. Soit $v_1, \ldots, v_d$ les éléments d'une base vectorielle de $A \cdot f$, l'algorithme de Zeilberger étendu recherche g sous la forme indéterminée $\phi_1 v_1 + \cdots + \phi_d v_d$, pour des fractions rationnelles ϕ_i en n et k. Cette recherche se fait par une extension ∂-finie de la variante paramétrée de l'algorithme de Gosper.

Tout ce qui a été dit s'étend au monde différentiel pour l'évaluation d'une intégrale paramétrée

$$F(x) = \int_a^b f(x,y)\, dy.$$

On cherche alors une relation

$$\eta_r(x) \frac{\partial^r f}{dx^r}(x,y) + \cdots + \eta_0(n) f(x,y) = \frac{\partial g}{dy}(x,y),$$

qui après intégration fournit l'égalité

$$\eta_r(n)F^{(r)}(x) + \cdots + \eta_0(n)F(x) = \int_a^b g(x,y)\,dy.$$

La même méthode permet aussi de traiter des sommations paramétrées continûment,

$$F(x) = \sum_{k=a}^{b} f_k(x),$$

et des suites d'intégrales de la forme

$$F_n = \int_a^b f_n(y)\,dy.$$

EXERCICE. Formuler la relation entre f et g à rechercher dans ces deux derniers cas.

2. L'algorithme sur l'exemple $\frac{1}{2}J_0(x)^2 + J_1(x)^2 + J_2(x)^2 + \cdots = \frac{1}{2}$

Nous allons montrer que la famille paramétrée des fonctions de Bessel de première espèce, J_ν, où chaque J_ν est une solution que nous allons préciser de l'équation de Bessel

$$x^2y''(x) + xy'(x) + (x^2 - \nu^2)y(x) = 0,$$

a une somme $\frac{1}{2}J_0(x)^2 + J_1(x)^2 + J_2(x)^2 + \cdots$ qui s'évalue à $\frac{1}{2}$.

L'équation de Bessel et les fonctions de Bessel peuvent être considérées pour des valeurs complexes du paramètre ν, mais vu la nature de la somme à étudier, nous nous limiterons dorénavant à des valeurs entières $\nu \in \mathbb{N}$. En étudiant l'équation indicielle de l'équation de Bessel, on s'aperçoit qu'il existe pour chaque ν des solutions dans les séries formelles $\mathbb{C}[[x]]$ et que ces solutions constituent un espace vectoriel de dimension 1 sur $\mathbb{C}$ de séries. Une base de ces solutions formelles est donnée par la série de Bessel

$$J_\nu(x) = (z/2)^\nu \sum_{n=0}^{\infty} \frac{(-1)^n (z/2)^{2n}}{n!\,(n+\nu)!},$$

de valuation ν, qui vu la décroissance de ses coefficients est pour chaque entier ν une série entière.

EXERCICE. Vérifier ces résultats.

On vérifie par simple substitution et évaluation que ces fonctions J_ν satisfont aussi aux relations

$$xJ'_\nu(x) + xJ_{\nu+1}(x) - \nu J_\nu(x) = 0 \qquad \text{et} \qquad xJ_{\nu+2}(x) - 2(\nu+1)J_{\nu+1}(x) + xJ_\nu(x) = 0.$$

En introduisant l'algèbre de Ore $A = \mathbb{C}(\nu, x)\langle S, D; \sigma, \mathrm{id}, 0, \delta\rangle$ où σ est le décalage avant sur ν et δ est la dérivation par rapport à x, on a donc un système d'annulateurs pour J,

$$\begin{aligned} p_1 &= x^2D^2 + xD + (x^2 - \nu^2),\\ p_2 &= xD + xS - \nu,\\ p_3 &= xS^2 - 2(\nu+1)S + x. \end{aligned}$$

Les deux premiers forment une base de Gröbner de l'idéal engendré pour l'ordre $\mathsf{lex}(S, D)$; les deux derniers pour l'ordre $\mathsf{lex}(D, S)$.

Bien évidemment, J est une fonction ∂-finie. Le module $A \cdot J$ est donné, par exemple, comme l'espace vectoriel sur $\mathbb{C}(\nu, x)$ de base $(J, S \cdot J)$. Pour représenter le carré de J en vue d'une sommation, on peut observer que, en tant qu'espace vectoriel, le module $A \cdot J^2$ admet la base $\big(J^2, J \times (S \cdot J), (S \cdot J)^2\big)$ et utiliser l'algorithme de clôture par produit pour obtenir une base de Gröbner. En fait, le calcul qui suit n'a même pas besoin d'une représentation aussi explicite de J^2 : pour calculer la somme $h(x) = \frac{1}{2}J_0(x)^2 + J_1(x)^2 + J_2(x)^2 + \cdots$ comme fonction de x, on recherche une fonction η de x, indépendante de ν, telle que $h' + \eta h$ soit la différence finie en ν d'un élément g de $A \cdot J^2$. Pour la suite du calcul, nous fixons cet élément sous la forme indéterminée donnée par

$$g(\nu) = \phi_0(\nu)J_\nu^2 + \phi_1(\nu)J_{\nu+1}^2 + \phi_2(\nu)J_\nu J_{\nu+1},$$

où nous avons omis de faire référence à la variable x dans les évaluations de g, des ϕ_i et de J, car cette variable ne va intervenir que comme paramètre dans le calcul des fractions rationnelles ϕ_i. (On peut penser qu'on travaille temporairement dans l'algèbre de Ore $A' = \mathbb{C}(\nu, x)\langle S; \sigma, 0\rangle$.)

Par construction, on a alors la relation $h' + \eta h = (S-1)\cdot g$, puis, après réduction de chaque occurence des dérivées et décalées de J par la base de Gröbner $\{p_2, p_3\}$,

$$\begin{aligned}2J_\nu J'_\nu + \eta J_\nu^2 &= (S-1)\cdot\big(\phi_0(\nu)J_\nu^2 + \phi_1(\nu)J_{\nu+1}^2 + \phi_2(\nu)J_\nu J_{\nu+1}\big)\\ &= \phi_0(\nu+1)J_{\nu+1}^2 - \phi_0(\nu)J_\nu^2 + \phi_1(\nu+1)x^{-2}\big(2(\nu+1)J_{\nu+1} - xJ_\nu\big)^2 - \phi_1(\nu)J_{\nu+1}^2\\ &\qquad + \phi_2(\nu+1)x^{-1}J_{\nu+1}\big(2(\nu+1)J_{\nu+1} - xJ_\nu\big) - \phi_2(\nu)J_\nu J_{\nu+1},\end{aligned}$$

laquelle se récrit

$$\begin{aligned}&\big(2vx^{-1}+\eta\big)J_\nu^2 - 2J_\nu J_{\nu+1}\\ &\quad= \big(\phi_1(\nu+1) - \phi_0(\nu)\big)J_\nu^2 - \big(4(\nu+1)x^{-1}\phi_1(\nu+1) + \phi_2(\nu+1) + \phi_2(\nu)\big)J_\nu J_{\nu+1}\\ &\qquad + \big(\phi_0(\nu+1) + 4(\nu+1)^2x^{-2}\phi_1(\nu+1) - \phi_1(\nu) + 2(\nu+1)x^{-1}\phi_2(\nu+1)\big)J_{\nu+1}^2.\end{aligned}$$

Comme les fonctions J_ν^2, $J_{\nu+1}^2$ et $J_\nu J_{\nu+1}$ sont linéairement indépendantes sur $\mathbb{C}(\nu, x)$, on déduit les relations nécessaires

$$\begin{aligned}-\phi_0(\nu) + \phi_1(\nu+1) &= (2\nu x^{-1} + \eta),\\ 4(\nu+1)x^{-1}\phi_1(\nu+1) + \phi_2(\nu) + \phi_2(\nu+1) &= 2,\\ \phi_0(\nu+1) - \phi_1(\nu) + 4(\nu+1)^2x^{-2}\phi_1(\nu+1) + 2(\nu+1)x^{-1}\phi_2(\nu+1) &= 0.\end{aligned}$$

En résolvant les deux premières respectivement en ϕ_0 et en ϕ_1, puis en substituant dans la dernière, on trouve la récurrence

$$\begin{aligned}&x^2(\nu+1)\phi_2(\nu+3) - (\nu+1)(4\nu^2+20\nu+24-x^2)\phi_2(\nu+2)\\ &\quad + (\nu+3)(4\nu^2+12\nu+8-x^2)\phi_2(\nu+1) - x^2(\nu+3)\phi_2(\nu) = -4x(\eta\nu^2 + 4\eta\nu + 3\eta + x).\end{aligned}$$

Nous résolvons maintenant celle-ci en ses solutions rationnelles par l'algorithme d'Abramov, dans sa variante paramétrée qui résoud en $\phi_2 \in \mathbb{C}(n, \nu)$ et simultanément en $\eta \in \mathbb{C}$. Les coefficients extrêmes de la partie homogène indiquent que toute solution rationnelle doit être polynomiale, puisque le P. G. C. D. $\big((\nu-3)+1\big) \wedge \big(x+3\big)$ vaut 1. La mise sous forme de différences finies de la partie homogène indique que l'opérateur associé accroît de 2 le degré d'un polynôme. Le degré de la partie inhomogène étant 2, toute solution rationnelle ne peut être qu'une constante. On trouve $\phi_2 = 1$ et $\eta = 0$, d'où après report $\phi_1 = 0$ et $\phi_0 = -2\nu/x$. Autrement dit, on a

$$D\cdot J_\nu^2 = (S-1)\cdot\big(J_\nu J_{\nu+1} - 2\nu x^{-1}J_\nu^2\big),$$

qui par sommation fournit

$$D\cdot\sum_{\nu=0}^{N} J_\nu^2 = J_{N+1}\big(J_{N+2} - 2(N+1)x^{-1}J_{N+1}\big) - J_0J_1.$$

Comme la série $J_N \in \mathbb{C}[[x]]$ a valuation N, le membre droit tend vers $-J_0J_1 = \frac{1}{2}D\cdot J_0^2$ quand N tend vers ∞ pour la topologie usuelle donnée par la métrique $|s| = 2^{-v}$ pour toute série non-nulle s de valuation v. On a donc

$$D\cdot\big(\tfrac{1}{2}J_0(x)^2 + J_1(x)^2 + J_2(x)^2 + \cdots\big) = 0$$

qui caractérise la somme par une condition initiale. Une simple évaluation en 0 montre que la somme vaut $\frac{1}{2}$, ce qui achève la preuve de l'identité annoncée.

3. Bases de Gröbner de modules et découplage de systèmes

Dans l'exemple qui précède, on a pour le moment effectué le découpler « à la main », mais un procédé systématique et automatique est disponible, par le biais des bases de Gröbner de modules, qui généralise la notion de base de Gröbner pour les idéaux. Cette notion existe tant dans le domaine des polynômes commutatifs que dans le cadre non-commutatif des algèbres de Ore ; nous la présentons directement dans ce second cas.

Dans le cas d'un idéal I d'une algèbre de Ore A, les éléments de I s'interprètent comme autant d'équations vérifiées par une fonction inconnue ϕ. Dans une perspective algorithmique, chaque idéal est engendré par un nombre fini de générateurs. Une question naturelle est celle de systèmes linéaires sur un vecteur de fonctions inconnues $(\phi_1, \dots, \phi_d)$, à coefficients dans A. On considère des systèmes d'un nombre fini d'équations de la forme $g_i = g_{i,1} \cdot \phi_1 + \cdots + g_{i,d} \cdot \phi_d$ pour des opérateurs $g_{i,j}$ de A. Un tel système se représente de façon compacte un par une matrice $(g_{i,j})$ à entrées dans A. Les questions qui sont alors naturelles sont celle de l'algèbre linéaire pour ces matrices, dont en particulier celle de donner un algorithme du pivot de Gauss pour des coefficients dans A, c'est-à-dire non plus dans un corps, mais dans un anneau, et non-commutatif de surcroît.

Pour ce faire, au lieu de considérer simplement l'algèbre de Ore $A = k(x)\langle\partial; \sigma, \delta\rangle$ munie d'un ordre monomial sur les ∂^a, pour lequel tout idéal à gauche admet une base de Gröbner, on s'intéresse plus généralement à un module libre de rang fini sur A, donné par une base sous la forme $A^d = Ae_1 + \cdots + Ae_d$ et muni d'un ordre sur les $\partial^a e_i$, dans lequel on va étendre la notion de base de Gröbner pour des sous-modules à gauche de A^d. La notion d'ordre monomial conserve formellement la même définition, si ce n'est que les e_i ne peuvent apparaître que linéairement dans les monômes $\partial^a e_i$ et qu'ils ne peuvent servir pour des multiplications à gauche. La notion de S-polynôme s'étend aussi mot pour mot, à ceci près que deux polynômes de monômes de tête $\partial^a e_i$ et $\partial^b e_j$ ont un S-polynôme nul dès lors que i et j sont différents. Les définitions et caractérisations équivalentes des bases de Gröbner d'idéaux restent alors valables pour les sous-modules du module libre A^d. L'algorithme de Buchberger, modifié pour suivre ces nouvelles définitions, termine sur tout sous-module en fournissant une base de Gröbner. Pour certains ordres, ce calcul correspond à l'algorithme de Gauss.

Un point de vue presque équivalent, mais qui donne une variante des calculs avec un peu plus de réductions, est que le calcul est celui d'une base de Gröbner dans l'anneau $A[e_1, \dots, e_d]$ des polynômes en les indéterminées commutatives e_i à coefficients dans l'anneau A pour l'idéal à gauche engendré par les g_i initiaux et tous les produits $e_i e_j = 0$.

Reprenons l'exemple du découplage des relations entre les coordonnées ϕ_i donnant g dans la section précédente sur la somme des carrés fonctions de Bessel. Ces relations se recodent par les éléments

$$\begin{aligned}
g_1 &:= -e_0 + Se_1 - (2\nu x^{-1} + \eta)e_3,\\
g_2 &:= 4(\nu+1)x^{-1}Se_1 + (S+1)e_2 - 2e_3,\\
g_3 &:= Se_0 + \big(4(\nu+1)^2x^{-2}S - 1\big)e_1 + 2(\nu+1)x^{-1}Se_2,\\
g_4 &:= (S-1)e_3,
\end{aligned}$$

du module libre A^4 pour l'algèbre de Ore $A = \mathbb{C}(n,k)\langle S_n, S_k; \sigma_n, \sigma_k, 0, 0\rangle$. Ici, chaque e_i représente la fonction rationnelle inconnue ϕ_i, et l'on a astucieusement représenté les second membres de équations inhomogènes d'origine comme multiple d'une nouvelle inconnue représentée par e_3 et contrainte par g_4 à être constante.

Le découplage effectué dans la section précédente revient au calcul d'une base de Gröbner pour l'ordre $\mathsf{lex}(e_0, e_1, e_2, e_3, S)$. Les monômes de tête respectifs des g_i sont e_0, Se_1, Se_0 et Se_3, si bien que le seul S-polynôme non nul est $\mathrm{Spoly}(g_1, g_3)$. Il est donné par

$$\mathrm{Spoly}(g_1, g_3) = Sg_1 + g_3 = \big(S^2 + 4(\nu+1)^2x^{-2}S - 1\big)e_1 + 2(\nu+1)x^{-1}Se_2 - \big(2(\nu+1)x^{-1} + \eta\big)Se_3.$$

Après réductions par g_2 et g_3, ce polynôme devient

$$g_5 = -e_1 - \left(\frac{x}{4(\nu+2)}S^2 + \frac{x^2 - 4\nu^2 - 12\nu - 8}{4x(\nu+2)}S + \frac{\nu+1}{x}\right)e_2 + \left(\frac{x}{2(\nu+2)} - \eta\right)e_3,$$

qui est adjoint à la base de Gröbner en cours de calcul. L'unique nouvel S-polynôme à considérer est celui entre ce g_5 et g_2, qui est $\mathrm{Spoly}(g_2, g_4) = g_2 + 4(\nu+1)x^{-1}Sg_5$ et a pour monôme de tête S^3e_2. Après réduction par e_3 et renormalisation, le dernier polynôme introduit dans la base de Gröbner est

$$\big((\nu+1)x^2S^3 + (\nu+1)(x^2 - 4\nu^2 - 20\nu - 24)S^2 - (\nu+3)(x^2 - 4\nu^2 - 12\nu - 8)S - (\nu+3)x^2\big)e_2 \\ + 4\big((\nu^2 + 4\nu + 3)\eta + x\big)xe_3.$$

Ce polynôme n'est autre qu'un recodage de l'équation inhomogène du troisième ordre qui a permis de déterminer ϕ_2 dans la section précédente.

Table des matières