

Mathématiques Expérimentales

En cours de rédaction

Notes du cours de Calcul Formel en M1 en 2^e année de l'ENS-Paris

Version préliminaire – début 2011

Alin Bostan & Bruno Salvy¹

1. La première version de la plupart des chapitres a été rédigée par nos brillants élèves de l'année 2008–2009 : Samuel Baumard, Roland Casalis, Sary Drappeau, Pierre Lairez, Bruno Le Floch, Henri Guenancia, Nicolas Mascot, Arnaud de Mesmay, Michaël Monereau, Aurel Page, Guillaume Scerri, Olivier Taïbi.

Table des matières

Cours 1. Introduction au calcul formel	1
1. Le théorème de Richardson	1
2. Structures et constructions de base	1
3. Équations comme structures de données	4
Notes	5
Bibliographie	6
TP : Coefficients de $(\sqrt{x^2 - 1})^{(n)}$	7
 Première partie. Calculs d'approximations	 9
Cours 2. Accélération de convergence	11
1. Introduction	11
2. Exemple : Archimède, Huygens et le calcul de π	11
3. Méthodes d'extrapolation linéaires	12
4. Méthodes non-linéaires	14
Bibliographie	16
TP : Calcul numérique de quelques sommes	17
 Cours 3. Calculs de séries par itération formelle de Newton	 19
1. Séries formelles	19
2. Méthode de Newton : principe	21
3. Itération de Newton pour l'inversion de série	21
4. Énoncé général	22
5. Applications	22
TP : Calcul de séries par itération de Newton	24
 Cours 4. Les approximants de Padé	 27
1. Le tableau de Padé	27
2. Fractions continues	30
3. Convergence	31
TP : Approximants de Padé	34
 Deuxième partie. De l'approximation à la formule : aides à la conjecture	 37
Cours 5. D'une série à une fraction rationnelle : approximants de Padé	39
1. Reconstruction rationnelle	39
2. Algorithme d'Euclide étendu	40
3. Calcul de la reconstruction rationnelle	41
4. Approximants de Padé	42

5. Algorithme de Berlekamp-Massey	43
6. Interpolation rationnelle de Cauchy	44
Complexité	45
TP : Un gros déterminant	46
Cours 6. Les approximants de Padé-Hermite ou la reconstruction d'équations	49
1. Premières définitions et premiers résultats	49
2. Algorithme de Derksen : idées et résultats préliminaires	50
3. Algorithme de Derksen : fonctionnement	52
4. Applications	53
5. Approximants de Padé-Hermite de type arbitraire	55
TP : Singularités d'une intégrale	56
Cours 7. Du flottant à la forme close : LLL	59
1. Introduction	59
2. Rappels et compléments	59
3. Quelques résultats préliminaires	61
4. L'algorithme BasePropre	62
5. L'algorithme BaseRéduite	62
6. Application : relations linéaires entre flottants	64
TP : Les décimales de π en base 16	65
Troisième partie. Preuves automatiques	67
Cours 8. Identités de fonctions spéciales et séries D-finies	69
1. Définitions	69
2. Équivalence entre séries D-finies et suites P-récurrentes	70
3. Test d'égalité	71
4. Somme et Produit	71
5. Séries algébriques	72
TP : La moyenne arithmético-géométrique et les séries hypergéométriques	74
Cours 9. Sommation hypergéométrique	77
1. Sommation indéfinie	77
2. Sommation définie	80
TP : Séries pour la fonction Zêta de Riemann aux entiers positifs	83
Cours 10. Résultants : propriétés et calcul euclidien	85
1. Définition et premières propriétés	85
2. Applications	89
3. Calcul	91
Bibliographie	92
TP : Utilisation de résultants	93
Cours 11. Bases de Gröbner	95
1. Définitions	95
2. Applications	97
TP : Bases de Gröbner et coloriage de graphes	100
Cours 12. Bases de Gröbner II : Calcul et Géométrie	103
1. Radicaux et Nullstellensatz	103

2. Calcul effectif des bases de Gröbner	105
TP : Bases de Gröbner pour la géométrie	108
Quatrième partie. Compléments	111
Cours 13. Systèmes linéaires et algorithme de Gauss-Jordan	113
Introduction	113
1. Formes réduites	113
2. Opérations élémentaires	114
3. L'algorithme de Gauss-Jordan	115
4. Exploitation de l'algorithme de Gauss-Jordan	117
5. L'algorithme de Wiedemann	119
Cours 14. Systèmes linéaires sur des anneaux euclidiens	121
1. Algorithme de Bareiss	121
2. Formes normales de Hermite et de Smith	123
3. Applications	125
Cours 15. Localisation de racines de polynômes	127
1. La méthode de Newton	127
2. Distance aux racines	128
3. Comptage de racines	130

Introduction au calcul formel¹

Résumé

L'indécidabilité n'est pas loin du calcul formel, mais il est possible de construire des classes d'objets assez sophistiquées dans lesquelles mener des calculs.

1. Le théorème de Richardson

D'une certaine manière, le calcul formel est fondé sur une contrainte d'origine logique.

Théorème 1 (Richardson, 1968). *Dans la classe des expressions obtenues à partir de $\mathbb{Q}(x)$, π , $\log 2$ par les opérations $+$, $-$, \times et la composition avec \exp , \sin et $|\cdot|$, le test d'équivalence à 0 est indécidable.*

Autrement dit, il n'existe pas d'algorithme permettant pour toute expression de cette classe de déterminer en temps fini si elle vaut 0 ou non. Plus généralement tout test d'égalité peut bien entendu se ramener à tester l'égalité à zéro dès que la soustraction existe. Cette limitation de nature théorique explique la difficulté et parfois la frustration que rencontrent les utilisateurs débutants des systèmes de calcul formel face à des fonctions de « simplification », qui ne peuvent être qu'heuristiques.

Pour effectuer un calcul, il est pourtant souvent crucial de déterminer si des expressions représentent 0 ou non, en particulier pour évaluer une fonction qui possède des singularités (comme la division). L'approche du calculateur formel expérimenté consiste à se ramener autant que faire se peut à des opérations d'un domaine dans lequel le test à zéro est décidable. Le calcul formel repose ainsi de manière naturelle sur des constructions algébriques qui préservent la décidabilité du test à 0. En particulier, les opérations courantes sur les vecteurs, matrices, polynômes, fractions rationnelles, ne nécessitent pas d'autre test à 0 que celui des coefficients.

2. Structures et constructions de base

Les objets les plus fondamentaux sont assez faciles à représenter en machine de manière exacte. Nous considérons tour à tour les plus importants d'entre eux, en commençant par les plus basiques. Ils s'assemblent ensuite à l'aide de tableaux ou de listes pour en former de plus complexes.

1. Ce chapitre est une version allégée de notre premier cours de M2 : les bases du calcul formel ne dépendent pas de l'année où on les voit. Le cours de M2 fait ensuite porter l'accent sur la complexité et l'efficacité alors que le cours de M1 insiste sur les calculs effectifs et l'approche expérimentale en mathématiques.

Entiers machine. Les entiers fournis par les processeurs sont des entiers modulo une puissance de 2 (le nombre de bits d'un mot machine, typiquement 32 ou 64). Ils sont appelés des *entiers machine*. Les opérations rendues disponibles par le processeur sont l'addition, la soustraction, la multiplication et parfois la division. La norme ANSI du langage C fournit au programmeur la division et le modulo pour ces entiers, c'est-à-dire que le compilateur implante ces opérations si le processeur ne le fait pas.

Entiers. Pour manipuler des entiers dont la taille dépasse celle d'un mot machine, il est commode de les considérer comme écrits dans une base B assez grande :

$$N = a_0 + a_1B + \dots + a_kB^k.$$

L'écriture est unique si l'on impose $0 \leq a_i < B$. (Le signe est stocké séparément.) Ces nombres peuvent être stockés dans des tableaux d'entiers machine. Les objets obtenus sont des entiers de taille arbitraire appelés parfois *bignums*.

L'addition et le produit peuvent alors être réduits à des opérations sur des entiers inférieurs à B^2 , au prix de quelques opérations de propagation de retenue. Le choix de B dépend un peu du processeur. Si le processeur dispose d'une instruction effectuant le produit de deux entiers de taille égale à celle d'un mot machine, renvoyant le résultat dans deux mots machines, alors B pourra être pris aussi grand que le plus grand entier tenant dans un mot machine. Sinon, c'est la racine carré de ce nombre qui sera utilisée pour B .

Entiers modulaires. Les calculs avec des polynômes, des fractions rationnelles ou des matrices à coefficients entiers souffrent souvent d'une maladie propre au calcul formel : la croissance des expressions intermédiaires. Les entiers produits comme coefficients des expressions intervenant lors du calcul sont de taille disproportionnée par rapport à ceux qui figurent dans l'entrée et dans la sortie.

Exemple 1. Voici le déroulement typique du calcul du plus grand diviseur commun (pgcd) de deux polynômes à coefficients entiers par l'algorithme d'Euclide :

$$P_0 = 7x^5 - 22x^4 + 55x^3 + 94x^2 - 87x + 56,$$

$$P_1 = 62x^4 - 97x^3 + 73x^2 + 4x + 83,$$

$$P_2 = \text{rem}(P_0, P_1) = \frac{113293}{3844}x^3 + \frac{409605}{3844}x^2 - \frac{183855}{1922}x + \frac{272119}{3844},$$

$$P_3 = \text{rem}(P_1, P_2) = \frac{18423282923092}{12835303849}x^2 - \frac{15239170790368}{12835303849}x + \frac{10966361258256}{12835303849},$$

$$P_4 = \text{rem}(P_2, P_3) = -\frac{216132274653792395448637}{44148979404824831944178}x - \frac{631179956389122192280133}{88297958809649663888356},$$

$$P_5 = \text{rem}(P_3, P_4) = \frac{20556791167692068695002336923491296504125}{3639427682941980248860941972667354081}.$$

Chaque étape calcule le reste (noté *rem* pour *remainder*) de la division euclidienne des deux polynômes précédents. Les coefficients de ces polynômes intermédiaires font intervenir des entiers qui croissent de manière exponentielle, alors que le résultat recherché est 1.

Les entiers modulaires remédient à ce problème de deux manières. D'une part, pour un calcul de décision, de dimension, ou de degré, l'exécution de l'algorithme

sur la réduction de l'entrée modulo un nombre premier donne un algorithme *probabiliste* répondant à la question. Cette technique peut aussi servir de base à un algorithme *déterministe* lorsque les nombres premiers pour lesquels la réponse est fausse peuvent être maîtrisés. C'est le cas du pgcd : en évitant les premiers qui divisent les coefficients de tête des deux polynômes, le degré du pgcd modulaire est le même que le degré du pgcd exact.

D'autre part, les entiers modulaires sont utilisés dans les algorithmes reposant sur le théorème des restes chinois. Ce théorème indique qu'un entier inférieur au produit de nombres premiers $p_1 \cdots p_k$ peut être reconstruit à partir de ses réductions modulo p_1, \dots, p_k . Lorsqu'une borne sur la taille du résultat est disponible, il suffit d'effectuer le calcul modulo suffisamment de nombres premiers (choisis assez grands pour que leur nombre soit faible et assez petits pour que les opérations tiennent dans un mot machine), pour ensuite reconstruire le résultat, court-circuitant de la sorte toute croissance intermédiaire.

Vecteurs et matrices. Une fois donnée une représentation exacte pour des coefficients, il est facile de construire des vecteurs ou matrices comme des tableaux, ou plus souvent comme des tableaux de pointeurs sur les coefficients. Les opérations de produit par un scalaire, de produit de matrices ou de produit d'une matrice par un vecteur se réduisent aux opérations d'addition et de multiplication sur les coefficients. Il en va de même de la recherche de noyau ou d'inverse de matrices.

Polynômes. Les polynômes peuvent être stockés de plusieurs manières, et la meilleure représentation dépend des opérations que l'on souhaite effectuer. Pour un polynôme en une variable, les choix principaux sont :

- la représentation dense : comme pour les entiers, le polynôme est représenté comme un tableau de (pointeurs sur les) coefficients ;
- la représentation creuse : le polynôme est représenté comme une liste de paires (coefficient, exposant) généralement triée par les exposants.

Par exemple, le système Maple utilise la seconde représentation par défaut, sans trier les exposants. En outre, il ne développe pas les produits automatiquement, et il faut le lui demander explicitement par la commande `expand`. Une autre commande utile sur les polynômes est `collect` qui regroupe les coefficients et permet d'y appliquer une fonction.

Récursivement, on construit bien sûr les polynômes multivariés.

Fractions rationnelles. Les rationnels peuvent être stockés comme des paires où numérateur et dénominateur sont des entiers de taille arbitraire. Les opérations d'addition et de multiplication se réduisent aux opérations analogues sur les entiers et le test d'égalité à zéro se réduit au test d'égalité à 0 sur le numérateur. De même, les fractions rationnelles sont représentées par des paires de polynômes. Les opérations d'addition, produit, division se réduisent aux additions et multiplications sur les coefficients.

Ces constructions sont possibles dès que les coefficients sont disponibles. Il est donc possible par exemple de manipuler des polynômes dont les coefficients sont des rationnels, des entiers modulaires, ou des matrices.

En Maple, les rationnels sont simplifiés automatiquement ; les fractions rationnelles le sont par la commande `normal`.

Séries tronquées. Les séries tronquées

$$\sum_{k=0}^N a_k X^k + O(X^{N+1})$$

se représentent pratiquement comme des polynômes. La différence principale apparaît lors du produit : les coefficients des termes d'exposant au moins $N + 1$ n'ont pas besoin d'être calculés, ni stockés. Cette structure de données joue un rôle très important non seulement pour des calculs d'approximations, mais aussi comme une représentation *exacte*. En voici trois exemples importants qui seront abordés dans le cours :

1. Une fraction rationnelle dont les numérateurs et dénominateurs ont degré borné par d peut être reconstruite à partir d'un développement en série à l'ordre $2d + 1$. Cette représentation joue ainsi un rôle clé dans l'algorithmique des suites récurrentes linéaires.
2. Un polynôme en deux variables peut être reconstruit à partir du développement en série d'une solution.
3. Il est possible de reconstruire une équation différentielle linéaire à coefficients polynomiaux à partir du développement en série d'une solution et de bornes sur l'ordre et le degré des coefficients. De façon analogue, il est possible de reconstruire une récurrence linéaire à coefficients polynomiaux à partir des premières valeurs d'une de ses solutions.

3. Équations comme structures de données

Une fois construits les objets de base que sont les polynômes, les séries ou les matrices, il est possible d'aborder des objets mathématiques construits *implicitement*. Ainsi, il est bien connu qu'il n'est pas possible de représenter toutes les solutions de polynômes de haut degré par radicaux, mais de nombreuses opérations sur ces solutions sont aisées en prenant le polynôme lui-même comme structure de données. Ce point de vue permet d'étendre le domaine d'application du calcul formel pourvu que des algorithmes soient disponibles pour effectuer les opérations souhaitées (typiquement addition, multiplication, multiplication par un scalaire, test d'égalité) par manipulation des équations elles-mêmes.

Nombres algébriques. C'est ainsi que l'on nomme les solutions de polynômes univariés. Les opérations d'addition et de multiplication peuvent être effectuées à l'aide de résultants (Cours 10). La division s'obtient par l'algorithme d'Euclide sur les polynômes (Cours 5), et le test à zéro se déduit du pgcd. Par exemple, il est possible de prouver assez facilement une identité comme

$$(1) \quad \frac{\sin \frac{2\pi}{7}}{\sin^2 \frac{3\pi}{7}} - \frac{\sin \frac{\pi}{7}}{\sin^2 \frac{2\pi}{7}} + \frac{\sin \frac{3\pi}{7}}{\sin^2 \frac{\pi}{7}} = 2\sqrt{7}$$

une fois que l'on reconnaît qu'il s'agit d'une égalité entre nombres algébriques.

Systèmes polynomiaux. De nombreuses questions naturelles sur un système de polynômes, comme l'existence de solutions, la dimension de l'espace des solutions (qui indique s'il s'agit d'une surface, d'une courbe, ou de points isolés), le degré, ou le calcul d'une paramétrisation de l'ensemble des solutions trouvent une réponse algorithmique en utilisant comme structure de données des bases de Gröbner, qui seront abordées dans les Cours 11-12.

Il est également possible d'éliminer une ou des variables entre des polynômes. Cette opération peut s'interpréter géométriquement comme une projection. Dans le cas le plus simple, elle permet de calculer un polynôme s'annulant sur les abscisses des intersections de deux courbes. Une autre application est l'implicitisation, qui permet par exemple de calculer une équation pour une courbe donnée sous forme paramétrée.

Équations différentielles linéaires. Cette structure de données permet de représenter de nombreuses fonctions usuelles transcendantes (exponentielle, fonctions trigonométriques et trigonométriques hyperboliques, leurs réciproques) ainsi que de nombreuses fonctions spéciales de la physique mathématique (fonctions de Bessel, de Struve, d'Anger, . . . , fonctions hypergéométriques et hypergéométriques généralisées), ainsi bien sûr que de multiples fonctions auxquelles n'est pas attaché un nom classique. Les opérations d'addition et de produit sont effectuées par des variantes noncommutatives du résultant qui se ramènent à de l'algèbre linéaire élémentaire (Cours 8). Le test à zéro se ramène à tester l'égalité d'un nombre fini de conditions initiales. En d'autres termes, des structures de données finies permettent de manipuler ces objets infinis et d'en tester l'égalité ou la nullité.

Ainsi, des identités élémentaires comme $\sin^2 x + \cos^2 x = 1$ sont non seulement facilement prouvables algorithmiquement, mais elles sont également calculables, c'est-à-dire que le membre droit se calcule à partir du membre gauche. Les relations étroites entre équations différentielles linéaires et récurrences linéaires — les séries solutions des unes ont pour coefficients les solutions des autres — amènent aux mêmes réponses algorithmiques à des questions sur des suites. Par exemple, l'identité de Cassini sur les nombres de Fibonacci

$$F_{n+2}F_n - F_{n+1}^2 = (-1)^{n+1}, \quad n \geq 0$$

est exactement du même niveau de difficulté que $\sin^2 x + \cos^2 x = 1$.

En conclusion, les exemples ci-dessus illustrent bien la manière dont le calcul formel parvient à effectuer de nombreux calculs utiles dans les applications malgré l'indécidabilité révélée par le théorème de Richardson.

Notes

Les références générales sur les algorithmes du calcul formel sont deux livres : celui de von zur Gathen et Gerhard [6] et celui, plus élémentaire, de Geddes, Czapor et Labahn [2].

Le théorème de Richardson [4] s'applique à des fonctions. Pour des constantes, l'approche la plus récente [5] réduit le test à zéro à une conjecture de théorie des nombres due à Schanuel qui exprime que les seules relations entre exponentielles et logarithmes sont celles qui découlent des formules d'addition et de multiplication.

L'implantation d'une arithmétique efficace pour les entiers longs (bignums) est un travail très délicat. Une des meilleures arithmétiques disponibles est fournie par GMP, le *Gnu Multiprecision Package* [3]. Elle est le résultat d'un travail de nombreuses années, qui comporte une partie importante de code assembleur consacré à la multiplication sur chacun des processeurs produits dans une période récente. Les entiers de GMP sont ceux qui sont utilisés dans Maple pour les grandes tailles. D'autres entiers très efficaces sont implantés dans le système Magma.

La jolie identité (1) est tirée de [1].

Bibliographie

- [1] Beck (Matthias), Berndt (Bruce C.), Chan (O-Yeat), and Zaharescu (Alexandru). – Determinations of analogues of Gauss sums and other trigonometric sums. *International Journal of Number Theory*, vol. 1, n° 3, 2005, pp. 333–356.
- [2] Geddes (Keith O.), Czapor (Stephen R.), and Labahn (George). – *Algorithms for Computer Algebra*. – Kluwer Academic Publishers, 1992.
- [3] Granlund (Torbjörn). – *GNU Multiple Precision Arithmetic Library*. – <http://swox.com/gmp>, 2006.
- [4] Richardson (Daniel). – Some undecidable problems involving elementary functions of a real variable. *Journal of Symbolic Logic*, vol. 33, n° 4, 1968, pp. 514–520.
- [5] Richardson (Daniel). – How to recognize zero. *Journal of Symbolic Computation*, vol. 24, n° 6, 1997, pp. 627–645.
- [6] von zur Gathen (Joachim) and Gerhard (Jürgen). – *Modern computer algebra*. – Cambridge University Press, New York, 2003, 2nd edition, xiv+785p.

TP 1

Coefficients de $(\sqrt{x^2 - 1})^{(n)}$

L'objectif de ce TP est double : d'une part il s'agit d'effectuer des premiers pas en Maple en alternant manipulations simples et recherches dans l'aide en ligne ; d'autre part, il montre comment utiliser le calcul formel pour conjecturer puis prouver une formule. L'exemple traité est celui de la dérivée n ème de $\sqrt{x^2 - 1}$.

1. Calculer les dix premières dérivées de $\sqrt{x^2 - 1}$ et observer qu'elles sont de la forme

$$(E) \quad \frac{d^n}{dx^n} \sqrt{x^2 - 1} = \frac{P_n(x)}{(x^2 - 1)^{\alpha_n}},$$

où P_n est un polynôme. Conjecturer les valeurs du degré de P_n et de α_n .

Dans la suite, on se concentrera sur le cas où n est un entier pair, le cas des valeurs impaires se traite de manière similaire.

2. Calculer le polynôme P_{100} .
3. Conjecturer une récurrence pour les valeurs des coefficients de P_{100} (à l'aide de la fonction `seriestorec` du package `gfun`). L'algorithme utilisé sera présenté plus tard dans le cours.
4. Factoriser les coefficients de cette récurrence, et en déduire une récurrence plausible pour les coefficients de P_n pour n pair arbitraire.
5. Résoudre cette récurrence.
6. Il reste à déterminer les conditions initiales $P_n(0)$. Pour cela, à nouveau, calculer les premières valeurs, conjecturer une récurrence et la résoudre.
7. Combiner ces conditions initiales avec les valeurs trouvées plus tôt pour donner une formule plausible pour l'équation (E) lorsque n est pair.
8. Utiliser le système pour prouver cette formule par récurrence.

Première partie

Calculs d'approximations

COURS 2

Accélération de convergence

Résumé

L'accélération de convergence est une technique d'analyse numérique qui s'avère utile en calcul formel, en conjonction avec la précision arbitraire et les outils de conjecture à base de l'algorithme LLL. Ce cours présente les principes de base des grandes familles de méthodes d'accélération de convergence.

1. Introduction

Le principe de l'accélération de convergence est assez simple : on connaît une suite réelle $(S_n)_{n \in \mathbb{N}}$ qui converge vers une valeur S_∞ et on cherche une nouvelle suite T_n qui tende aussi vers S_∞ , mais (beaucoup) plus rapidement. Autrement dit, $T_n - S_\infty = o(S_n - S_\infty)$ lorsque $n \rightarrow \infty$. Il est possible de trouver de telles suites T_n si l'on dispose d'hypothèses supplémentaires sur la régularité avec laquelle S_n tend vers sa limite.

2. Exemple : Archimède, Huygens et le calcul de π

2.1. La suite à accélérer. Pour encadrer π , le point de départ de la méthode d'Archimède consiste à considérer deux polygones réguliers, l'un inscrit et l'autre circonscrit à un cercle de rayon 1. Une petite figure permet de se convaincre que pour tout n , ces polygones ont pour périmètres $2n \sin \pi/n$ et $2n \tan \pi/n$, d'où découle l'inégalité suivante

$$n \sin \frac{\pi}{n} < \pi < n \tan \frac{\pi}{n}.$$

Ensuite, Archimède, avec bien moins d'outils que ce dont nous disposons, se rend compte qu'il est possible de calculer simultanément les éléments des deux suites $s_k := \sin(\alpha/2^k)$ et $t_k := \tan(\alpha/2^k)$ à l'aide des relations

$$\frac{1}{\tan \frac{x}{2}} = \frac{1}{\tan x} + \frac{1}{\sin x} \quad \text{et} \quad \sin^2 \frac{x}{2} = \frac{1}{1 + \frac{1}{\tan^2 \frac{x}{2}}}.$$

À l'aide de ces relations, et en partant de $\alpha = \pi/3$, Archimède pousse le calcul jusqu'à $k = 5$, ce qui correspond à des polygones à 96 côtés !

Exercice 1. Réaliser ce calcul en Maple. En déduire un encadrement de π qui donne ses 3 premières décimales.

2.2. Principe de l'accélération. Le développement de Taylor de \tan donne directement le développement asymptotique

$$n \tan \frac{\pi}{n} = \pi + \frac{\pi^3}{3n^2} + O\left(\frac{1}{n^4}\right), \quad n \rightarrow \infty.$$

Lorsque l'angle est divisé par 2, n est doublé, ce qui mène à

$$2n \tan \frac{\pi}{2n} = \pi + \frac{1}{4} \frac{\pi^3}{3n^2} + O\left(\frac{1}{n^4}\right), \quad n \rightarrow \infty.$$

L'idée de la méthode consiste à éliminer le terme en $1/n^2$ par une combinaison linéaire. Si $t_k^{(0)} := 3 \cdot 2^k \tan(\pi/(3 \cdot 2^k))$, une suite à convergence plus rapide est ainsi fournie par

$$t_k^{(1)} := \frac{4t_{k+1}^{(0)} - t_k^{(0)}}{3} = \pi + O\left(\frac{1}{16^k}\right), \quad k \rightarrow \infty.$$

Cette idée, jointe à une manipulation analogue sur le sinus, est due à Huygens qui s'en est servi en 1654 pour calculer 35 décimales de π .

Exercice 2. Calculer à l'aide de Maple les éléments de ces suites que l'on déduit de ceux de l'exercice précédent.

De nombreuses années plus tard, en 1936, Kommerel se rend compte qu'il est possible de réitérer cette transformation, en considérant cette fois

$$t_k^{(2)} := \frac{16t_{k+1}^{(1)} - t_k^{(1)}}{15} = \pi + O\left(\frac{1}{64^k}\right), \quad k \rightarrow \infty.$$

Exercice 3. Calculer non seulement les premiers éléments de cette suite et ceux de la suite analogue pour les polygones inscrits, mais aussi ceux des suites $t_k^{(3)}, t_k^{(4)}, t_k^{(5)}$ dont il faut d'abord déterminer la bonne définition.

3. Méthodes d'extrapolation linéaires

Le calcul de Huygens est un cas particulier des méthodes d'extrapolation linéaires. Le terme *linéaire* qualifie ici l'application d'accélération envoyant une suite sur une suite accélérée, et on parle d'extrapolation pour signifier que l'on cherche à déterminer une valeur en dehors du domaine des valeurs de départ.

3.1. Méthode d'Euler. Il s'agit sans doute de la plus vieille des méthodes d'accélération. Elle s'applique à des suites dont le comportement asymptotique est de la forme suivante (ou est supposé l'être) :

$$S_n = S_\infty + r^n \varphi(n) \quad \text{avec} \quad \frac{\varphi(n+1)}{\varphi(n)} \rightarrow 1 \quad \text{et} \quad r \neq 1 \quad \text{connu.}$$

(La convergence exige en outre $|r| \leq 1$.)

L'accélération repose sur le principe simple suivant.

Proposition 1. Dans une telle situation, la suite $T_n := (S_{n+1} - rS_n)/(1 - r)$ vérifie $T_n - S_\infty = o(S_n - S_\infty)$.

DÉMONSTRATION. Il suffit de développer :

$$\begin{aligned} T_n &= \frac{S_{n+1} - rS_n}{1 - r} \\ &= S_\infty + \frac{r^{n+1}}{1 - r} (\varphi(n+1) - \varphi(n)) \\ &= S_\infty + \frac{r^{n+1}}{1 - r} \underbrace{\varphi(n) \left(\frac{\varphi(n+1)}{\varphi(n)} - 1 \right)}_{o(\varphi(n))}. \end{aligned}$$

□

Exemple 1. La série géométrique peut être accélérée par cette méthode. Si $S_n = 1 + \alpha + \alpha^2 + \dots + \alpha^n$, alors le choix $r = \alpha$ donne $T_n = S_\infty$.

La méthode d'Euler consiste à réitérer ce procédé pour obtenir une suite de suites accélérées en posant

$$(1) \quad T_n^{(0)} := S_n, \quad T_n^{(k)} := \frac{T_{n+1}^{(k-1)} - rT_n^{(k-1)}}{1 - r}, \quad (k \geq 1).$$

Exercice 4. Utiliser cette méthode (en Maple) avec $r = -1$ pour calculer des décimales de $\ln 2$ donné comme limite de la suite

$$S_n = \sum_{i=0}^n \frac{(-1)^i}{i+1}.$$

On demande de calculer les 6 premières décimales de $\ln 2$ en partant seulement des 10 premières valeurs S_1, \dots, S_{10} .

Exercice 5. Appliquer les mêmes opérations sur la suite pourtant divergente

$$\sum_{i=0}^n (-1)^i \frac{2^{i+1}}{i+1}.$$

Comparer à $\ln 3$.

3.2. Méthode de Richardson (1910). Il s'agit d'une généralisation de la méthode précédente au cas où la suite se comporte comme

$$(2) \quad S_n = S_\infty + c_1 r_1^n + \dots + c_k r_k^n,$$

les r_i étant connus, distincts, différents de 1, et de module décroissant. Le principe est ici d'éliminer les r_i les uns après les autres par la Proposition 1. Pour ceci, on suit un procédé de construction itératif :

$$(3) \quad T_n^{(0)} := S_n, \quad T_n^{(i)} = \frac{T_{n+1}^{(i-1)} - r_i T_n^{(i-1)}}{1 - r_i}, \quad i = 1, \dots, k.$$

Exemple 2. L'accélération du calcul de π dans la section précédente est une accélération de Richardson avec $r_1 = 1/4, r_2 = 1/16, \dots$

Le terme $T_n^{(k)}$ est appelé *transformée de Richardson* de S_n . La suite $T_n^{(k)}$ converge (en n) vers S_∞ plus vite que S_n puisque les termes résiduels ont été amortis un à un. Plus la valeur de k est élevée, plus l'accélération est rapide (on a amorti plus de termes). Une autre formulation de $T_n^{(k)}$ est fournie par les formules

de Cramer : le système linéaire est fourni par (2) évalué en $n, n+1, \dots, n+k$ ayant pour inconnues S_∞ et les c_i . Il s'ensuit une expression en terme de déterminants :

$$T_n^{(k)} = \frac{\begin{vmatrix} S_n & 1 & \dots & 1 \\ \vdots & \vdots & & \vdots \\ S_{n+k} & r_1^k & \dots & r_k^k \end{vmatrix}}{\begin{vmatrix} 1 & 1 & \dots & 1 \\ \vdots & \vdots & & \vdots \\ 1 & r_1^k & \dots & r_k^k \end{vmatrix}}.$$

3.3. Convergence plus lente. Dans le cas très fréquent où la convergence de la suite est de la forme

$$S_n = S_\infty + \frac{\alpha}{n} + \frac{\beta}{n^2} + \dots,$$

la méthode d'Euler ne s'applique pas directement puisqu'il faudrait prendre $r = 1$. L'idée est alors de poser d'abord $\widetilde{S}_n = S_{2^n}$, ce qui permet alors d'utiliser la méthode de Richardson avec $r_i = 1/2^i$. L'équation (3) devient alors

$$T_n^{(0)} = S_{2^n}, \quad T_n^{(k+1)} := \frac{2^{k+1}T_{n+1}^{(k)} - T_n^{(k)}}{2^{k+1} - 1}.$$

Exercice 6. Employer cette méthode pour calculer une dizaine de décimales de la constante γ d'Euler, définie comme limite de la suite

$$S_n = \sum_{k=1}^n \frac{1}{k} - \log n.$$

3.4. Méthode de Romberg (1955). C'est la méthode précédente, appliquée au calcul d'une intégrale par la méthode des trapèzes. La suite S_n est alors

$$S_n = \frac{h}{2}(f(a) + f(b)) + h \sum_{k=1}^{n-1} f(a + kh), \quad h = \frac{b-a}{n}.$$

Il faut cependant faire attention que si la fonction ne se comporte pas aimablement sur l'intervalle $[a, b]$, la convergence peut ne pas être assez bonne pour appliquer la méthode.

Exercice 7. Calculer de cette manière quelques décimales de $\int_0^\pi \sqrt{x} dx$.

4. Méthodes non-linéaires

4.1. Méthode Δ^2 d'Aitken (1926). Cette méthode s'applique aux suites dont le comportement est de la forme

$$S_n = S_\infty + r^n \varphi(n), \quad \text{avec} \quad \frac{\varphi(n+1)}{\varphi(n)} \longrightarrow 1,$$

mais cette fois-ci, r n'est pas supposé connu. La méthode peut être vue comme un calcul simultané de r et de l'accélération d'Euler.

Pour approcher r , on considère la suite

$$\Delta S_n = S_{n+1} - S_n = r^n (r\varphi(n+1) - \varphi(n)).$$

La notation Δ est classique pour cet opérateur aux différences. On observe alors que $\Delta S_{n+1}/\Delta S_n \longrightarrow r$. Cette propriété est très utile, même en relation avec des

méthodes linéaires : si l'on parvient à identifier r en contemplant les premières valeurs de cette suite, alors il vaut mieux exploiter cette valeur.

La *transformée d'Aitken* consiste donc naturellement à appliquer la Proposition 1, mais en remplaçant r par sa valeur approchée ci-dessus, ce qui donne

$$T_n = \frac{S_{n+1} - \frac{\Delta S_{n+1}}{\Delta S_n} S_n}{1 - \frac{\Delta S_{n+1}}{\Delta S_n}} = S_n - \frac{(\Delta S_n)^2}{\Delta^2 S_n}.$$

Ici, Δ^2 désigne la composition de l'opérateur Δ avec lui-même : $\Delta^2 u_n = \Delta(u_{n+1} - u_n) = u_{n+2} - 2u_{n+1} + u_n$.

Exercice 8. Calculer numériquement la limite de

$$S_n = \sum_{k=0}^n \frac{F_k}{2^k},$$

où les F_k sont les nombres de Fibonacci définis par $F_0 = 0$, $F_1 = 1$ et $F_{k+2} = F_{k+1} + F_k$ pour $k \geq 0$.

4.2. Méthode de Shanks (1949). La méthode de Shanks est à la méthode d'Aitken ce que la méthode de Richardson est à la méthode d'Euler : il s'agit d'éliminer plusieurs termes perturbateurs, mais cette fois-ci les r_i sont inconnus. Alors que la méthode d'Aitken peut se récrire

$$T_n = \frac{\begin{vmatrix} S_n & S_{n+1} \\ S_{n+1} & S_{n+2} \end{vmatrix}}{|\Delta^2 S_n|},$$

le cas général devient

$$T_n^{(k)} = \frac{\begin{vmatrix} S_n & \dots & S_{n+k} \\ \vdots & & \vdots \\ S_{n+k} & \dots & S_{n+2k} \end{vmatrix}}{\begin{vmatrix} \Delta^2 S_n & \dots & \Delta^2 S_{n+k-1} \\ \vdots & & \vdots \\ \Delta^2 S_{n+k-1} & \dots & \Delta^2 S_{n+2k-2} \end{vmatrix}}.$$

Comme ci-dessus, la vitesse de convergence augmente avec k . Cette méthode était semble-t-il déjà connue de Jacobi.

4.3. Algorithme ε de Wynn (1956). L'algorithme de Wynn réalise la transformation de Shanks par une récurrence qui évite le calcul de déterminants.

$$\varepsilon_{-1}^{(n)} = 0, \quad \varepsilon_0^{(n)} = S_n, \quad \varepsilon_{k+1}^{(n)} = \varepsilon_{k-1}^{(n+1)} + \frac{1}{\varepsilon_k^{(n+1)} - \varepsilon_k^{(n)}}.$$

Nous admettons alors le résultat suivant.

Théorème 2. $\varepsilon_{2k}^{(n)}$ est le $T_n^{(k)}$ de la méthode de Shanks.

Bibliographie

- [1] Brezinski (Claude). – *Accélération de la convergence en analyse numérique*. – Springer-Verlag, 1977, *Lectures Notes in Mathematics*, vol. 584.
- [2] Laurie (Dirk). – *The SIAM 100-digit challenge*, Chapter Convergence acceleration, pp. xii+306. – Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA, 2004. A study in high-accuracy numerical computing, With a foreword by David H. Bailey.

Calcul numérique de quelques sommes

L'objectif de ce TP est l'utilisation de méthodes d'accélération de convergence, ainsi que de récurrences, pour l'évaluation numérique de sommes sur l'exemple

$$S := \lim_{N \rightarrow \infty} S_N \quad \text{où} \quad S_N := \sum_{k=1}^N \frac{(-1)^{k+1}}{k^2} \sum_{i=1}^k \frac{(-1)^{i+1}}{i}.$$

1. Estimer l'ordre de grandeur du nombre de termes à utiliser pour calculer 30 décimales de la somme *sans* accélération de convergence. (Utiliser le caractère alterné de la somme).
2. Calculer les 1100 premiers termes de la suite S_n et déterminer une méthode d'accélération de convergence adaptée. Estimer empiriquement le nombre de décimales de S ainsi obtenu.

Pour obtenir encore plus de décimales par cette méthode, il faut pouvoir évaluer S_{2^m} pour $m > 10$ en un temps raisonnable.

3. Calculer exactement, c'est-à-dire comme des rationnels, les nombres S_0, \dots, S_{100} .
4. Conjecturer une récurrence pour les valeurs de S_N à l'aide de `gfun[listtorec]`.
5. Traduire cette récurrence en une procédure permettant d'évaluer numériquement S_N (avec la procédure `gfun[rectoproc]` et son option `evalfun`).
6. Conforter la conjecture en vérifiant la valeur de S_{200} .
7. Utiliser la procédure pour calculer S_{2^m} pour m jusqu'à 15.
8. Appliquer alors une accélération de convergence comme en question (3).
9. À l'aide de la valeur ainsi obtenue, il est également possible de conjecturer *une forme close* pour S . Pour cela, utiliser la fonction `identify`. Pour aider `identify`, il faut exploiter (par l'option `BasisSumConst`) le fait que la somme

$$\sum_{k=1}^{\infty} \frac{(-1)^{k+1}}{k^2} \sum_{i=1}^{\infty} \frac{(-1)^{i+1}}{i}$$

admet une expression simple, que Maple peut calculer, et dont une combinaison linéaire avec S a une forme simple.

Finalement, ce calcul a permis d'obtenir suffisamment de décimales pour arriver à la conjecture suivante¹

$$S = \frac{1}{4}\pi^2 \ln(2) - \frac{5}{8}\zeta(3).$$

1. Des preuves de cette identité et d'autres similaires existent (R. Sitaramachandrarao. A formula of S. Ramanujan, *Journal of Number Theory* 25 (1987), no. 1, 1–19. P. Flajolet and B. Salvy, Euler sums and contour integral representations, *Experimental Mathematics* 7 (1998), no. 1, 15–35.), mais aucune n'est vraiment simple.

Calculs de séries par itération formelle de Newton

Résumé

L'objectif de ce cours est de donner une vision unifiée du calcul sur les séries, de plus ces calculs se feront avec une bonne complexité.

1. Séries formelles

Définition 1. Soit A un anneau commutatif unitaire. L'ensemble $A[[X]]$ des séries formelles sur A est l'ensemble des suites (u_n) d'éléments de A , notées $U(X) = \sum_{n \geq 0} u_n X^n$ et muni des opérations

$$(1) \quad \sum a_n X^n + \sum b_n X^n = \sum (a_n + b_n) X^n,$$

$$(2) \quad \left(\sum a_n X^n \right) \times \left(\sum b_n X^n \right) = \sum \left(\sum_{i+j=n} a_i b_j \right) X^n.$$

Comme pour les séries entières, on appelle u_0 le terme constant de la série $U(X)$, noté aussi $U(0)$.

- Les premières propriétés élémentaires qui découlent des identités (1-2) sont :
- la série $1 = 1 + 0 \cdot X + 0 \cdot X^2 + \dots$ est élément neutre pour la multiplication ;
 - la série $1 + X + X^2 + \dots$ est inverse de $1 - X$.

Définition 2. La valuation $\text{val}(f)$ d'une série formelle f est l'indice de son premier coefficient non nul, et par convention $\text{val}(0) = \infty$.

Cette valuation permet de définir une métrique et donc une notion de convergence dans les séries formelles :

Proposition 1. Pour $f, g \in A[[X]]$ on pose :

$$d(f, g) = 2^{-\text{val}(f-g)}.$$

Avec cette fonction comme distance, $A[[X]]$ est un espace métrique complet.

DÉMONSTRATION. Le seul point à observer est l'inégalité triangulaire, qui provient de

$$\text{val}(F + G) \geq \min(\text{val } F, \text{val } G)$$

appliqué à $F = f - g$ et $G = g - h$.

Avec cette métrique, une suite Cauchy de séries formelles (F_k) est telle que pour tout n , il existe K tel que pour tous k_1, k_2 supérieurs à K , on ait $\text{val } F_{k_1} - F_{k_2} > n$. Autrement dit, les n premiers coefficients de F_k ne bougent plus pour $k > K$. Ceci permet pour tout n de définir le n ° coefficient d'une série formelle, qui est par construction limite de cette suite. Ceci prouve la complétude de l'espace. \square

On définit ensuite la composition.

Définition 3. Soient $f, g \in A[[X]]$ avec $g(0) = 0$, on définit $f \circ g$ comme limite de la suite

$$F_k = f_0 + f_1g + f_2g^2 + \dots + f_kg^k.$$

La limite existe parce que $g(0)$ étant nul, la valuation de f_kg^k vaut au moins k , ce qui fait de la suite une suite de Cauchy.

Lemme 1. Si $f = 1 - g$ avec $\text{val } g > 0$, alors l'inverse de f vaut

$$f^{-1} = 1 + g + g^2 + g^3 + \dots$$

DÉMONSTRATION. Il suffit de composer l'identité $(1 + X + X^2 + \dots)(1 - X) = 1$ avec g . \square

Si $f(0) = a$ est inversible, alors $f^{-1} = (a^{-1}f)^{-1}a^{-1}$, où l'inverse de $a^{-1}f$ est donné par la formule du lemme.

À ce stade, nous avons obtenu le résultat suivant.

Proposition 2. $A[[X]]$ est un anneau commutatif unitaire. Ses inversibles sont les séries de terme constant inversible.

(Le fait qu'un inversible doit avoir un terme constant inversible se constate en regardant le terme constant de $ff^{-1} = 1$).

Définition 4. On supposera $\mathbb{Q} \subset A$.

Soit $f = \sum a_n X^n$, on définit la dérivation comme suit :

$$f' = \sum n a_n X^{n-1}$$

On définit une primitive comme suit :

$$F = \sum \frac{1}{n+1} a_n X^{n+1}$$

Proposition 3. (Formule de Taylor)

On suppose $\mathbb{Q} \subset A$. Soient $f, g, h \in A$ tels que $\text{val}(g), \text{val}(h) > 0$, alors on a :

$$f(g+h) = f(g) + \sum_{i \geq 1} \frac{f^{(i)}(g)}{i!} h^i$$

Définition 5. (Exponentielle et logarithme)

Supposons $\mathbb{Q} \subset A$, $\text{val}(f) > 0$.

$$\exp(f) = \sum_{i \geq 0} \frac{1}{i!} f^i$$

$$\log(1+f) = \sum_{i \geq 0} \frac{(-1)^{i-1}}{i} f^i$$

Remarque 1. – Les deux définitions précédentes ne sont qu'un cas particulier de composition.

– On a les propriétés usuelles :

$$(\exp(f))' = f' \exp(f) \text{ et } \exp(f+g) = \exp(f) \exp(g)$$

Définition 6. (Séries tronquées)

Soit $f = \sum a_i X^i \in A[[X]]$, $N \in \mathbb{N}$, on note $f \bmod X^N = \sum_{i=0}^{N-1} a_i X^i \in A[X]$
Soient $f, g \in A[[X]]$ ou $A[X]$, on note : $g = f + O(X^N)$ si $(f-g) \bmod X^N = 0$

2. Méthode de Newton : principe

La méthode de Newton consiste en le processus itératif suivant : on remplace l'équation ou le système à résoudre par le linéarisé au voisinage de la dernière solution.

Version numérique (Newton, 1669). Soit ϕ une application dérivable, on cherche u de façon approché tel que $\phi(y + u) = 0$. On voudrait :

$$0 = \phi(y + u) = \phi(y) + u\phi'(y) + O(u^2)$$

En négligeant le $O(u^2)$, on obtient $u = -\frac{\phi(y)}{\phi'(y)}$. Ceci nous donne l'itération de Newton numérique partant de y_0 :

$$y_{n+1} = y_n - \frac{\phi(y_n)}{\phi'(y_n)}, \quad n \geq 0$$

La convergence ainsi obtenue est quadratique sous de bonnes hypothèses.

3. Itération de Newton pour l'inversion de série

Soit $f = \sum a_n X^n \in A[[X]]$, inversible ($a_0 \in A^*$). L'inverse $g \in A[[X]]$ de f est solution de $\phi(g) = 0$ avec $\phi(y) = \frac{1}{y} - f$. On a $\phi'(y) = -\frac{1}{y^2}$.

l'opérateur de Newton est ici $h \mapsto N(h)$ avec

$$\begin{aligned} N(h) &= h - \frac{\frac{1}{h} - f}{-\frac{1}{h^2}} \\ &= h + h(1 - fh). \end{aligned}$$

On veut montrer la convergence de l'itération de Newton.

Lemme 2. Si $h = f^{-1} \pmod{X^k}$, alors $N(h) = f^{-1} \pmod{X^{2k}}$

DÉMONSTRATION. $h = f^{-1} \pmod{X^k} \Rightarrow hf = 1 \pmod{X^k}$. On écrit alors $hf = 1 + X^k R$ avec $R \in A[[X]]$, ainsi

$N(h)f = hf + hf(1 - fh) = hf(1 - X^k R) = (1 + X^k R)(1 - X^k R) = 1 + X^{2k} R^2$,
d'où le résultat en multipliant par f^{-1} . □

On prend alors $h_0 = f(0)^{-1}$ et $h_{n+1} = N(h_n) \pmod{X^{2^{n+1}}}$. Le lemme nous donne : $h_n \rightarrow \frac{1}{f}$ à vitesse quadratique.

Corollaire 1. Soit $f \in A[[X]]$ tel que $val(f) > 0$. On a $\log(1 + f) = \int \frac{f'}{1+f}$. La méthode précédente nous donne un calcul de $\log(1 + f)$ par itération de Newton, qui converge plus vite que $\sum \frac{(-1)^{i-1}}{i} f^i$

Remarque 2. (Inversion d'une matrice de séries)

Soit $M \in \mathcal{M}_n(A[[X]])$ inversible. Posons $Y_0 = M(0)^{-1}$,

$$y_{k+1} = Y_k + Y_k(Id - MY_k) \pmod{X^{2^{k+1}}}$$

Alors Y_k converge vers M^{-1} à vitesse quadratique. (La démonstration de ce résultat est laissée en exercice au lecteur)

4. Énoncé général

Proposition 4. Soit $\phi(X, Y) \in A[[X, Y]]$ avec $\phi(0, 0) = 0$ et $\frac{\partial \phi}{\partial Y} \in A^*$.

1. (Existence) Il existe une unique solution $y \in A[[X]]$ telle que $\phi(X, y(X)) = 0$ dans $A[[X]]$.
2. (Calcul) Posons $y_0 = 0$, $y_{k+1} = y_k - \frac{\phi(X, y_k)}{\frac{\partial \phi}{\partial Y}(X, y_k)} \pmod{X^{2^{k+1}}}$. Alors, on a convergence quadratique de y_k vers y (ie : $y_k - y = O(X^{2^k})$).

DÉMONSTRATION. On va démontrer la propriété suivante par récurrence sur k ce qui nous donnera le résultat :

$$\begin{cases} \phi(X, y_k(X)) &= O(X^{2^k}), \\ y_k - y_{k+1} &= O(X^{2^k}). \end{cases}$$

Le résultat est clair pour $k = 0$.

Supposons la propriété vérifiée pour k . La démonstration repose sur la formule de Taylor :

$$\phi(X, y_{k+1}) = \underbrace{\phi(X, y_k) - (y_k - y_{k+1}) \frac{\partial \phi}{\partial Y}(X, y_k)}_{O(X^{2^{k+1}})} + \underbrace{O((y_k - y_{k+1})^2)}_{O(X^{2^{k+1}})}$$

Avec la définition de y_{k+2} , ceci donne $y_{k+1} - y_{k+2} = O(X^{2^{k+1}})$, ce qui termine la preuve de récurrence. En découle la convergence de (y_k) et, comme $A[[X]]$ est complet, l'existence de la limite $y \in A[[X]]$. Cette limite est bien solution de l'équation :

$$\underbrace{\phi(X, y_k)}_{\rightarrow 0} = \phi(X, y) - \underbrace{(y_k - y) \frac{\partial \phi}{\partial Y}(X, y_k)}_{\rightarrow 0} + \underbrace{O((y - y_k)^2)}_{\rightarrow 0}.$$

□

5. Applications

5.1. $P(x, y) = 0$. Soit $P \in A[X, Y]$, λ tel que $P(0, \lambda) = 0$ et $P_y(0, \lambda)$ soit inversible.

La méthode de Newton nous donne une représentation $y(x)$ des solutions par :

$$y_0 = \lambda, y_{n+1} = y_n - \frac{P(x, y_n)}{P_y(x, y_n)} \pmod{X^{2^{n+1}}}$$

tel que $y_n \rightarrow y$ à vitesse quadratique.

5.2. Exponentielle d'une série (Brent 1975). $\phi(X, Y) = \log(1+Y) - f(X)$ avec $f(0) = 0$.

$$\phi(0, 0) = 0, \frac{\partial \phi}{\partial Y}(X, Y) = \frac{1}{1+Y}, \frac{\partial \phi}{\partial Y}(0, 0) = 1$$

Posons $y_0 = 0$ et $y_{k+1} = y_k - (1 + y_k)(\log(1 + y_k) - f) \pmod{X^{2^{k+1}}}$, alors $y_k \rightarrow \exp(f) - 1$.

5.3. Inverse compositionnelle (Brent-Kung, 1978). Soit $f \in A[[X]]$ telle que $f(0) = 0$, $f'(0) = 1$. On veut s telle que $f \circ s = s \circ f = X$.

Posons $\phi(y) = f(y) - X$, $y_0 = 0$ et

$$y_{n+1} = y_n - \frac{f(y_n) - X}{f'(y_n)} \pmod{X^{2^{n+1}}}.$$

On a : $y_n \rightarrow s$.

5.4. Équations différentielles linéaires d'ordre 1. $f' + Bf = C$ avec $B, C \in A[[X]]$.

Équation homogène : $f'_0 + Bf_0 = 0$ soit $f'_0/f_0 = -B$, soit $f_0 = \exp(\int -B)$, cette solution se calcule avec la méthode de Newton.

Posons $f = \lambda f_0$, on a $\lambda' f_0 = C$ soit $\lambda = \int \frac{C}{f_0}$ qui se calcule encore par la méthode de Newton.

5.5. Équations non linéaires. Soit $P \in A[X]$, on s'intéresse à l'équation $y' = P(y)$.

Posons $\phi : y \mapsto y' - P(y)$

$$\begin{aligned} \underbrace{\phi(y+u) - \phi(y)}_{=0} &= y' + u' - P(y+u) - y' + P(y) \\ &= u' + P(y) - P(y+u) \\ &\approx u' + P(y) - (P(y) + uP'(y)) \\ &= u' - P'(y)u \end{aligned}$$

On pose $\tilde{y} = y + u$, ce qui nous donne encore un itérateur.

5.6. Système algébrique polynomial. On veut résoudre avec $X, Y \in A[[X]]$ le système suivant :

$$\begin{cases} f(X, Y) = 0 \\ g(X, Y) = 0. \end{cases}$$

$$0 = f(x+u, y+v) = f(x, y) + u \frac{\partial f}{\partial x} + v \frac{\partial f}{\partial y} + 0(u^2 + v^2)$$

$$0 = g(x+u, y+v) = g(x, y) + u \frac{\partial g}{\partial x} + v \frac{\partial g}{\partial y} + 0(u^2 + v^2)$$

En négligeant le terme quadratique on obtient :

$$\begin{pmatrix} \frac{\partial f}{\partial x} & \frac{\partial f}{\partial y} \\ \frac{\partial g}{\partial x} & \frac{\partial g}{\partial y} \end{pmatrix} = - \begin{pmatrix} f(x, y) \\ g(x, y) \end{pmatrix}.$$

On construit un itérateur exactement de la même façon que précédemment, c'est à dire :

$$\begin{pmatrix} x_{n+1} \\ y_{n+1} \end{pmatrix} = \begin{pmatrix} x_n \\ y_n \end{pmatrix} - \begin{pmatrix} \frac{\partial f}{\partial x} & \frac{\partial f}{\partial y} \\ \frac{\partial g}{\partial x} & \frac{\partial g}{\partial y} \end{pmatrix}^{-1} (x_n, y_n) \begin{pmatrix} f(x_n, y_n) \\ g(x_n, y_n) \end{pmatrix}$$

Comme précédemment, on a $\begin{pmatrix} x_n \\ y_n \end{pmatrix} \rightarrow \begin{pmatrix} x \\ y \end{pmatrix}$ à vitesse quadratique.

Calcul de séries par itération de Newton

L'objectif de ce TP est l'utilisation de l'itération de Newton pour la résolution en série de plusieurs types d'équations ou de systèmes. Une motivation importante est fournie par les séries génératrices d'énumération en combinatoire : il s'agit de séries formelles de la forme

$$\sum_{n=0}^{\infty} a_n z^n \quad \text{ou} \quad \sum_{n=0}^{\infty} a_n \frac{z^n}{n!},$$

où a_n désigne le nombre d'objets de taille n dans une certaine famille. Dans le premier cas, la série génératrice est dite *ordinaire* ; elle est *exponentielle* dans le second.

1. Introduction : arbres binaires

Un arbre binaire à n sommets est soit vide ($n = 0$), soit composé d'une racine et de deux sous-arbres binaires de taille k et $n - k - 1$. Les nombres C_n d'arbres binaires de taille n s'appellent les nombres de Catalan.

1. Écrire une procédure prenant une taille n en argument et renvoyant C_n à partir d'une récurrence non-linéaire simple.
2. À partir de C_0, \dots, C_{10} , deviner un polynôme dont la série génératrice $\sum C_n z^n$ est solution.
3. Résoudre ce polynôme et vérifier que sa solution donne bien les C_n jusqu'à $n = 100$.
4. Deviner une récurrence linéaire satisfaite par les C_n , et la résoudre.

2. Arbres d'arité 5 et équations algébriques

Ces arbres sont définis comme les arbres binaires, chaque sommet interne ayant cinq sous-arbres et non plus 2.

5. Calculer le nombre B_n d'arbres d'arité 5 à n sommets, pour $n = 0, \dots, 20$.
6. Deviner un polynôme dont la série génératrice $\sum B_n z^n$ est solution (il faudra changer des réglages par défaut de `gfun`, voir `?gfun, Parameters`).
7. À l'aide de ce polynôme et d'une itération de Newton, calculer les nombres B_n , $n = 0, \dots, 63$.
8. (À la fin, s'il reste du temps). Utiliser les 50 premiers de ces nombres pour deviner une récurrence sur les B_n , la résoudre, et confirmer la solution en comparant la valeur pour $n = 60$.

3. Des arbres bicolorés et un système

Il est possible de considérer des règles plus ou moins complexes de construction. Par exemple, des arbres dont les sommets peuvent être bleus ou verts, ont une arité finie mais illimitée, avec la contrainte qu'une racine bleue a au moins un fils vert, alors qu'une racine verte a au moins deux fils bleus. Ces arbres ont des séries génératrices d'énumération qui sont solution de

$$B(z) = z + \frac{zV(z)}{(1-V(z))(1-B(z))}, \quad V(z) = z + \frac{zB(z)^2}{(1-V(z))(1-B(z))}.$$

9. Construire la matrice jacobienne du système ;
10. l'utiliser pour former une itération de Newton ;
11. calculer le nombre d'arbres bicolorés dont la racine est bleue pour les tailles $n = 0, \dots, 31$.

4. Arbres ordonnés et équations différentielles

Un arbre ordonné de taille n est un arbre dont chaque nœud porte une étiquette distincte entre 1 et n , de telle sorte que l'étiquette d'une racine soit toujours plus petite que les étiquettes de ses fils. Pour les arbres ternaires (d'arité 3) ordonnés, l'équation satisfaite par la série génératrice exponentielle est

$$Y'(z) = 1 + Y(z)^3.$$

12. Déterminer l'équation différentielle linéaire à utiliser à chaque itération de Newton, et écrire une procédure permettant de la résoudre en série ;
13. calculer ensuite par itération de Newton les 32 premiers coefficients.

5. Arbres 2-3 et équation fonctionnelle

Les arbres 2-3 sont une structure de données utilisée en bases de données. Ils permettent la recherche, l'insertion et la suppression en temps amorti logarithmique. Les nœuds internes ont deux ou trois fils, et les nœuds externes sont tous au même niveau. Leur série génératrice d'énumération vérifie l'équation fonctionnelle

$$T(z) = z + T(z^2 + z^3).$$

14. Déterminer l'équation linéarisée à résoudre pour l'itération de Newton ;
15. écrire une procédure permettant de résoudre cette équation par itération ;
16. écrire l'itération de Newton et calculer les 64 premiers coefficients de la série.

Les approximants de Padé

Résumé

Il est toujours utile de posséder des outils pour approximer une fonction donnée par d'autres fonctions plus aisément calculables. Les développements limités constituent un bon exemple de cette approche, la simplicité des polynômes permet de résoudre aisément un grand nombre des problèmes que l'on peut se poser sur des fonctions a priori alambiquées (limites, zéros, calculs d'intégrales, ...). Les approximants de Padé s'inscrivent dans la continuité de cette démarche : pour approcher précisément une fonction, on peut substituer à la traditionnelle approche polynomiale une approximation par des fractions rationnelles. Si celles-ci sont moins pratiques à manier que les polynômes, elles ont néanmoins un avantage crucial : elles incluent la notion de pôles, là où les polynômes s'affolent. C'est cette propriété qui en fait l'outil idéal pour approximer les fonctions analytiques et leur caractère méromorphe. C'est également le bon outil pour discuter de problèmes d'algébricité, et les premières démonstrations historiques de la transcendance de e ou de π utilisent de tels approximants.

1. Le tableau de Padé

1.1. Définition. Un approximant de Padé d'ordre (m, n) d'une série $U \in k[[X]]$ est une fraction rationnelle

$$U_m^n(x) = \frac{\sum_{k=0}^n a_k x^k}{\sum_{k=0}^m b_k x^k}$$

avec $b_0 = 1$ et $U - U_m^n = O(x^{m+n+1})$.

1.2. Calcul. Pour calculer l'approximant de Padé, partons du résultat :

$$U - \frac{A}{B} = O(x^{m+n+1}) \Leftrightarrow BU - A = O(x^{m+n+1})$$

(car $b_0 \neq 0$).

Ce qui donne les systèmes :

$$\left\{ \begin{array}{l} b_0 u_0 = a_0, \\ b_1 u_0 + b_0 u_1 = a_1, \\ \vdots \\ b_m u_m u_{n-m} + \cdots + b_0 u_n = a_n. \end{array} \right. \quad \left\{ \begin{array}{l} b_m u_{n-m+1} + \cdots + b_0 u_{n+1} = 0, \\ \vdots \\ \vdots \\ b_m u_n + \cdots + b_0 u_{n+m} = 0. \end{array} \right.$$

Celui de droite est un système de m équations à m inconnues, et une fois celui-ci résolu les coefficients a_i se lisent sur celui de gauche. D'où la proposition suivante.

Proposition 1. *Si le déterminant*

$$\begin{vmatrix} u_{n-m+1} & \cdots & u_n \\ \vdots & & \vdots \\ u_n & \cdots & u_{n+m-1} \end{vmatrix}$$

est non nul, alors U_m^n existe et est unique.

De plus, la complexité du calcul se réduit à la résolution d'un système linéaire $m \times m$.

1.3. Le tableau de Padé. Le tableau de Padé est le tableau suivant :

$$\begin{array}{cccc} U_0^0 & U_0^1 & U_0^2 & \cdots \\ U_1^0 & U_1^1 & U_1^2 & \\ U_2^1 & U_2^1 & U_2^2 & \\ \vdots & & & \ddots \end{array}$$

Définition 1. *La suite $U_0^0, U_1^0, U_1^1, U_2^1, U_2^2, \dots$ est dite normale si les déterminants sont non nuls.*

1.4. Reconstruction de fractions rationnelles. : L'idée est simple et intuitive : lorsqu'on essaye d'approximer une fraction rationnelle par des approximants de Padé, à partir d'un certain rang, on retombe sur cette fraction rationnelle !

Proposition 2. *Si $U = \frac{P}{Q}$ avec $Q(0) \neq 0$, alors pour tout $m \geq \deg Q, n \geq \deg P$,*

$$U_m^n = U.$$

DÉMONSTRATION.

$$\begin{aligned} U_m^n &= \frac{A}{B} = U + O(x^{m+n+1}) \\ \underbrace{AQ}_{\deg \leq m+n} &= \underbrace{PB}_{\deg \leq m+n} + O(x^{m+n+1}) \\ AQ &= BP \\ \frac{A}{B} &= \frac{P}{Q} \end{aligned}$$

□

Application : Reconnaissance des suites récurrentes linéaires. (à détailler)

1.5. Application numérique. : Regardons maintenant si numériquement, les approximants de Padé fournissent de bonnes approximations aux fonctions usuelles, $\ln(1+x)$ pour $x = 2$ dans cet exemple. Le calcul du développement limité et des premiers approximants de Padé donne :

$$\begin{aligned} \ln(1+x) &= x - \frac{x}{2} + \frac{x}{3} + \cdots \\ \ln(1+x)_n^n &= \frac{x}{1 + \frac{x}{2}}, \frac{x + \frac{x^2}{2}}{1 + x + \frac{x^2}{6}} + \cdots \\ \ln(1+x)_{n+1}^n &= \frac{x}{1 + \frac{x}{2}}, \frac{x}{1 + \frac{x}{2} - \frac{x^2}{12}} + \cdots \end{aligned}$$

Calculons les approximations correspondantes en $z = 2$:

- Pour la première ligne : 1 ; 1.0909 ; 1.0980 ; 1.09857, la suite est croissante
- Pour la deuxième ligne : 1.2 ; 1.14 ; 1.101 ; 1.0988 ; 1.09862, la suite est décroissante

Dans cet exemple, les approximants de Padé fournissent donc une bonne approximation (car on a à la fois une minoration et une majoration) *même en dehors du disque de convergence*.

1.6. Lien avec l'accélération de convergence. La méthode d'Aitken consiste, pour accélérer la convergence d'une suite $S_n = l + c\rho^n$, à introduire $\Delta S_n = c(\rho - 1)\rho^n$, ce qui donne donc

$$\frac{\Delta S_{n+1}}{\Delta S_n} = \rho$$

puis à considérer

$$T_n := \frac{S_n - \frac{\Delta S_{n+1}}{\Delta S_n} S_{n-1}}{1 - \frac{\Delta S_{n+1}}{\Delta S_n}}$$

La suite T_n considérée converge alors "mieux" (cf. cours sur l'accélération de convergence) que S_n . L'opération réalisée sur S_n étant de nature algébrique (manipulations de fractions rationnelles), on peut s'attendre à la retrouver en calculant un approximant de Padé de la bonne fonction, c'est le résultat qu'on obtient dans la proposition suivante :

Proposition 3. *En prenant $S_n = U_0 + \dots + U_n x^n$,*

$$T_n = U_1^n.$$

DÉMONSTRATION. Écrivons U_1^n sous la forme $\frac{A}{1+b_1x}$, alors on a

$$\begin{aligned} \frac{A}{1+b_1x} &= S_{n+1} + O(x^{n+2}) \\ A &= S_{n+1}(1+b_1x) + O(x^{n+2}) \\ 0 &= u_{n+1} + b_1 u_n \\ b_1 &= -\frac{u_{n+1}}{u_n} \end{aligned}$$

A étant de degré n,

$$\begin{aligned} A &= S_n + b_1 x S_{n-1} \\ \Delta S_n &= S_{n+1} - S_n = u_{n+1} x^{n+1} \\ \frac{\Delta S_n}{\Delta S_{n-1}} &= -b_1 x \\ \frac{A}{B} &= \frac{S_n - \frac{\Delta S_n}{\Delta S_{n-1}} S_{n-1}}{1 - \frac{\Delta S_n}{\Delta S_{n-1}}} \end{aligned}$$

□

En fait l'approximant d'ordre (k, n) est lié à l'accélération de convergence par la remarque plus générale suivante.

Proposition 4. *Avec $S_n = U_0 + \dots + U_n x^n$, la k e accélération de Shanks donne U_k^n .*

2. Fractions continues

Un outil efficace d'approximation de fonctions par des fractions rationnelles est celui des fractions continues (ou continuées). Nous allons voir que les deux outils coïncident, et même que les fractions continues fournissent un bon algorithme de calcul de certains approximants de Padé.

2.1. Définitions. Une fraction continue est une suite de fractions rationnelles dont le $(n + 1)$ e terme s'écrit :

$$F_n = \frac{c_0}{1 + \frac{c_1 z}{1 + \frac{c_2 z}{\ddots \frac{c_{n-1} z}{1 + c_n z}}}}$$

Le développement en fraction continue d'une série est une telle suite dont le n^e terme coïncide avec la série à l'ordre n .

2.2. Calcul. Le calcul d'un développement en fraction continue se fait par récurrence :

On part de

$$N_0(z) = U_0 + U_1 z + \dots$$

$$D_0(z) = 1.$$

On construit une suite de séries par

$$\begin{aligned} \frac{N_i(z)}{D_i(z)} &= \frac{\frac{N_i(0)}{D_i(0)}}{1 + z \frac{N_{i+1}(z)}{D_{i+1}(z)}} \\ \frac{N_{i+1}(z)}{D_{i+1}(z)} &= \frac{1}{z} \left(\frac{N_i(0) D_i(z)}{D_i(0) N_i(z)} - 1 \right) \\ &= \frac{N_i(0)}{N_i(z)} \frac{1}{z} \left(\frac{D_i(z)}{D_i(0)} - \frac{N_i(z)}{N_i(0)} \right) \end{aligned}$$

Et donc

$$D_{i+1}(z) = \frac{N_i(z)}{N_i(0)}$$

et

$$N_{i+1}(z) = \frac{1}{z} \left(\frac{D_i(z)}{D_i(0)} - \frac{N_i(z)}{N_i(0)} \right)$$

d'où on déduit $c_{i+1} = N_{i+1}(0)/D_{i+1}(0)$.

On a ainsi un algorithme explicite de calcul des F_n , et en évaluant le nombre d'opérations utilisées, on a la majoration suivante :

Proposition 5. *On peut calculer c_0, \dots, c_n en $O(n^2)$ opérations.*

2.3. Numérateur et dénominateur. On pose $F_n := P_n/Q_n$.

Les fractions continues vérifient de très nombreuses propriétés de récurrence, la suivante nous sera utile, qui n'est que du calcul :

Proposition 6.

$$\begin{pmatrix} P_{n+1} \\ Q_{n+1} \end{pmatrix} = \begin{pmatrix} P_n \\ Q_n \end{pmatrix} + x_{n+1}z \begin{pmatrix} P_{n-1} \\ Q_{n-1} \end{pmatrix}.$$

DÉMONSTRATION. On démontre la propriété par récurrence : Pour $n = 0$,

$$\begin{pmatrix} P_1 \\ Q_1 \end{pmatrix} = \begin{pmatrix} P_0 \\ Q_0 \end{pmatrix} + c_1z \begin{pmatrix} 0 \\ 1 \end{pmatrix}$$

correspond à

$$\begin{pmatrix} c_0 \\ 1 + c_1z \end{pmatrix} = \begin{pmatrix} c_0 \\ 1 \end{pmatrix} + c_1z \begin{pmatrix} 0 \\ 1 \end{pmatrix},$$

ce qui est évident.

Supposons la propriété vraie au rang n , on a

$$\begin{aligned} \frac{P_{n+1}}{Q_{n+1}} &= \frac{P_n}{Q_n} \Big|_{c_n \rightarrow \frac{c_n}{1+c_{n+1}z}} \\ &= \frac{P_{n-1} + c_nzP_{n-2}}{Q_{n-1} + c_nzQ_{n-2}} \Big|_{c_n \rightarrow \frac{c_n}{1+c_{n+1}z}} \\ &= \frac{P_{n-1} + \frac{c_nz}{1+c_{n+1}z}P_{n-2}}{Q_{n-1} + \frac{c_nz}{1+c_{n+1}z}Q_{n-2}} \\ &= \frac{P_n + c_{n+1}zP_{n-1}}{Q_{n-1} + c_{n+1}zQ_{n-1}}. \end{aligned}$$

□

2.4. Lien avec les approximants de Padé. Si la suite $U_0^0, U_1^0, \dots, U_{n+1}^n$ est normale, son $(n+1)^e$ élément vaut F_n . On obtient ainsi un bon algorithme de calcul des approximants de Padé diagonaux et sous-diagonaux dans le tableau.

3. Convergence

Comme pour les développements limités, où l'on a longtemps cherché des conditions nécessaires et suffisantes pour que la série de Taylor converge vers la fonction, il est naturel ici de se demander sous quelles conditions les approximants de Padé vont converger, ce qui justifiera leur utilisation comme outils d'approximation numérique.

3.1. Le terme général.

Proposition 7. On a

$$F_n = c_0 - \frac{c_0c_1z}{Q_0(z)Q_1(z)} + \dots + \frac{c_0 \cdots c_n(-z)^n}{Q_{n-1}(z)Q_n(z)}.$$

DÉMONSTRATION. Il suffit de montrer que

$$F_{n+1} - F_n = \frac{c_0 \cdots c_{n+1}(-z)^{n+1}}{Q_n(z)Q_{n+1}(z)}.$$

Multiplions par $-Q_n$, puis par P_n , on obtient :

$$P_{n+1} = P_n + c_{n+1}zP_{n-1}, \quad Q_{n+1} = Q_n + c_{n+1}zQ_{n-1}.$$

Et donc

$$\begin{aligned} \omega_n &:= -P_{n+1}Q_n + P_nQ_{n+1} = -c_{n+1}z \underbrace{(P_{n-1}Q_n - P_nQ_{n-1})}_{=\omega_{n-1}} \\ &= c_{n+1} \cdots c_1 (-z)^{n+1} (P_{-1}Q_0 - P_0Q_{-1}) \end{aligned}$$

d'où le résultat. \square

3.2. Les fractions continues à coefficients positifs. On part de la remarque évidente suivante : si $c_i z > 0$, alors les P_i et les Q_i aussi, qui nous fournit un bon argument de convergence par des suites adjacentes :

Proposition 8. *Si $(c_i z)_i$ est une suite positive, alors U_n^n décroît, U_{n+1}^n croît et $U_1^0 \leq U_2^1 \leq \cdots \leq U_{n+2}^{n+1} \leq \cdots \leq U_0^0$, et donc U_n^n converge et U_{n+1}^n aussi.*

Exemple 1. On considère la fonction suivante :

$$\frac{\ln(1+z)}{z}$$

Le calcul des coefficients c_i donne $c_0 = 1$ et

$$c_n = \frac{(n+1)^2 + (-1)^{n+1}(2n+1)}{8n(n+1)} > 0.$$

On peut donc bien réaliser une approximation de cette fonction par des approximants de Padé pour les valeurs de z positives.

3.3. Séries de Stieltjes. Les séries de Stieltjes sont des séries particulières où cette propriété est toujours vérifiée, et réciproquement, la positivité des coefficients c_i les relie naturellement à une série de Stieltjes, c'est pourquoi elles interviennent naturellement ici :

Définition 2. *Une série $\sum u_n (-z)^n$ est de Stieltjes s'il existe une fonction $\rho(x) > 0$ pour $x \geq 0$ telle que*

$$u_k = \int_0^\infty t^k \rho(t) dt$$

pour tout k .

Proposition 9. *Si U est de Stieltjes, elle est le développement asymptotique de*

$$f(z) = \int_0^\infty \frac{\rho(t)}{a + tz} dt$$

quand $z \rightarrow 0$ avec $|\arg z| < \pi$.

Cette proposition nous permet de capter l'essentiel des propriétés des séries de Stieltjes qui nous intéressent ici :

Proposition 10. :

- Les coefficients de la fraction continue sont ≥ 0 .
- Si une fraction continue a des coefficients ≥ 0 , il existe une fonction de Stieltjes les ayant pour coefficients.
- Si la suite f_n à coefficients > 0 converge, alors c'est vers une fonction de Stieltjes.

- Si $\sum_{n \geq 1} c_n^{-\frac{1}{2n}}$ diverge, alors la suite F_n converge.
- Si $F_n(z_0)$ converge pour un z_0 dans $\mathbb{C} - \mathbb{R}^-$ alors $F_n(z)$ converge pour tout z

Approximants de Padé

L'objectif de ce TP est l'exploration des capacités d'approximation des approximants de Padé bien au-delà du disque de convergence d'une série formelle. Le point de départ est une équation différentielle :

$$(E) \quad y' = 1 + y^2, \quad y(0) = 0.$$

La solution est bien entendu la fonction tangente $y(x) = \tan(x)$, sur laquelle on pourra s'appuyer pour estimer la qualité des approximations, mais le but du TP est de ne pas tenir compte de cette connaissance et de traiter le problème en partant de l'équation différentielle.

1. Développement en série

Dans un premier temps, sauter cette partie et utiliser pour S la série tronquée de $\tan(x)$ à l'ordre 32.

Le calcul du développement de la solution de (E) peut s'obtenir par itération de Newton.

1. Déterminer l'équation différentielle linéaire à utiliser à chaque itération de Newton, et écrire une procédure permettant de la résoudre en série ;
2. calculer ensuite par itération de Newton les 32 premiers coefficients. Soit S la série tronquée obtenue.

2. Approximation

3. Calculer un approximant de Padé (15,16) de S (voir ?pade). On appelle ensuite F la fraction obtenue.
4. Tracer sur un même dessin les graphes de F , de S et de la fonction tangente, pour $-20 \leq x \leq 20$.
5. Évaluer la qualité des approximations de \tan par F et par S en $x = 1$, c'est-à-dire à l'intérieur du disque de convergence de S .
6. Estimer la vitesse de convergence de la suite d'approximants de Padé diagonaux en $x = 1$.
7. Effectuer les mêmes opérations en $x = 10$, c'est-à-dire *en dehors* du disque de convergence de la série.

Les approximants de Padé sont souvent employés pour localiser les singularités et les zéros de fonctions sur lesquelles on dispose de peu d'information.

8. Calculer numériquement les racines du dénominateur de F , et comparer les plus petites en valeur absolue aux valeurs attendues.
9. Faire de même pour le numérateur.
10. Construire une animation permettant de visualiser la convergence des approximants diagonaux vers la tangente.

3. Validité des approximations

La qualité empirique des estimations obtenues par approximants de Padé est en général difficile à justifier rigoureusement. Dans le cas particulier de la tangente, cependant, il est possible de se ramener aux résultats de convergence pour les séries de Stieltjes.

11. Calculer le début du développement en fraction continue de la tangente hyperbolique $\tanh(x)$ et deviner la forme générale des coefficients. (Une preuve de cette conjecture sera vue plus tard dans le cours ; le résultat dans ce cas particulier est dû à Lambert qui en avait déduit la première preuve de l'irrationalité de π).
12. Analyser la convergence de cette fraction continue, puis conclure en observant les transformations liant les approximants de Padé de \tan et ceux de \tanh .

Deuxième partie

De l'approximation à la formule :
aides à la conjecture

D'une série à une fraction rationnelle : approximants de Padé

Résumé

L'algorithme d'Euclide étendu permet le calcul d'approximants de Padé. Plus généralement, il permet d'effectuer la reconstruction des fractions rationnelles.

Le problème abordé dans ce cours est le calcul des approximants de Padé. Plus généralement, on s'intéresse au problème suivant (*reconstruction rationnelle*) : trouver une fraction rationnelle de degré donné qui soit congrue à un polynôme donné modulo un autre polynôme donné. Un autre cas particulier important est l'interpolation de Cauchy des fractions rationnelles. Dans la section 1, nous définissons le problème de la reconstruction rationnelle et faisons quelques remarques préliminaires. Dans la section 2, nous rappelons brièvement le fonctionnement de l'algorithme d'Euclide étendu et quelques propriétés utiles. Dans la section 3, nous montrons comment la reconstruction rationnelle peut se ramener à l'algorithme d'Euclide étendu. Les applications au calcul d'approximants de Padé, à la devinette de récurrences (algorithme de Berlekamp-Massey) et à l'interpolation de Cauchy sont traitées dans les dernières sections 4, 5 et 6.

1. Reconstruction rationnelle

Soit \mathbb{K} un corps, $f \in \mathbb{K}[X]$ un polynôme de degré $n > 0$ et $g \in \mathbb{K}[X]$ de degré $< n$. Pour un $k \in \{1, \dots, n\}$ fixé, on se pose la question de trouver un couple $(r, t) \in \mathbb{K}[X]^2$ vérifiant :

$$(RR) \quad \text{pgcd}(t, f) = 1, \quad \deg(r) < k, \quad \deg(t) \leq n - k \quad \text{et} \quad \frac{r}{t} \equiv g \pmod{f}.$$

Les cas particuliers les plus importants correspondent aux choix $f = X^n$ (*approximation de Padé*) et $f = \prod_i (X - u_i)$, avec $u_i \in \mathbb{K}$ distincts deux à deux (*interpolation de Cauchy*).

Faisons quelques observations préliminaires :

- (i) Si $k = n$, alors clairement $(r, t) = (g, 1)$ est une solution de (RR).
- (ii) Si $k < n$, il est possible que (RR) n'admette pas de solution. Ceci est le cas en prenant $n = 3, k = 2$ et $f = X^3, g = X^2 + 1 \in \mathbb{K}[X]$. En effet, si $t(X) = aX + b$ avec $b \neq 0$, alors $r \equiv (aX + b)(X^2 + 1) = bX^2 + aX + b \pmod{X^3}$, ce qui est incompatible avec $\deg(r) \leq 1$.
- (iii) Par ailleurs, si (RR) admet une solution (r, t) , alors la fraction rationnelle r/t est forcément unique. En effet, si $(r_1, t_1) \in \mathbb{K}[X]^2$ est une autre solution de (RR), alors $r_1/t_1 \equiv r/t \pmod{f}$, donc f divise $r_1 t - t_1 r$. Or, le polynôme

$r_1t - t_1r$ ayant un degré strictement inférieur à celui de f , il doit être identiquement nul. Donc les fractions r/t et r_1/t_1 coïncident.

Un problème plus simple. Si $r, t \in \mathbb{K}[X]$ sont tels que (r, t) est solution du problème (RR), alors (r, t) vérifie aussi le problème plus simple

$$(RRS) \quad \deg(r) < k, \quad \deg(t) \leq n - k \quad \text{et} \quad r \equiv tg \pmod{f},$$

où, à la différence de (RR), on a mis de côté la contrainte sur le pgcd de f et de t .

Remarquons tout de suite que le problème (RRS) admet *toujours* une solution non-triviale $(r, t) \neq (0, 0)$: en effet, il se traduit en termes d'algèbre linéaire en un système linéaire homogène ayant $k + (n - k + 1) = n + 1$ inconnues (les coefficients de r et de t) et n équations. Par ailleurs, la solution r/t du problème (RRS) est encore unique (même idée de preuve que pour (RR) : si f divise à la fois $r - tg$ et $r_1 - t_1g$, alors il divise également la combinaison linéaire $rt_1 - r_1t = (r - tg)t_1 - (r_1 - t_1g)t$).

Lien avec le pgcd étendu. Nous allons prouver que le problème (RRS) peut être résolu en utilisant l'algorithme d'Euclide étendu AEE rappelé dans §2. Nous allons en déduire ensuite dans §3 une procédure de décision et calcul pour (RR).

L'intuition du lien entre le problème (RRS) et le calcul du pgcd étendu s'appuie sur la remarque simple suivante : la congruence $r \equiv tg \pmod{f}$ équivaut à l'existence d'un polynôme s tel que $r = sf + tg$; or, cette dernière égalité est une *relation de type Bézout*.

Pour préciser un peu cette remarque, traitons brièvement le problème (RRS) dans le cas particulier $k = 1$. Si f et g sont premiers entre eux, alors on prend $r = 1$ et la relation de Bézout $r = sf + tg$ avec $\deg(t) < \deg(f) = n$ fournit la réponse. Sinon, on prend $r = 0$ et comme le ppcm de f et g a un degré $< m + n$, alors $t = \text{ppcm}(f, g)/g$ est de degré $\leq n - 1$ et vérifie $tg = 0 \pmod{f}$.

2. Algorithme d'Euclide étendu

Étant donnés deux polynômes f et g à coefficients dans un corps \mathbb{K} , l'algorithme d'Euclide étendu (AEE, Fig. 1) calcule une suite de restes successifs dont le degré décroît, jusqu'à atteindre le pgcd de f et g . Il calcule également une suite de cofacteurs, les derniers fournissant l'identité de Bézout qui exprime $\text{pgcd}(f, g)$ comme combinaison linéaire polynomiale de f et g .

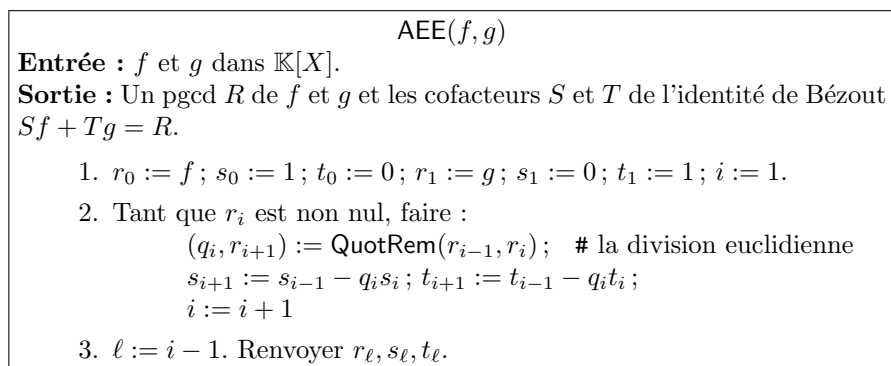


FIGURE 1. L'algorithme d'Euclide étendu

La terminaison de cet algorithme provient de la décroissance stricte des degrés des r_i . La correction se déduit de la relation $\text{pgcd}(f, g) = \text{pgcd}(h, g)$ pour $h := f \bmod g$. Par récurrence, il s'ensuit que $\text{pgcd}(f, g) = \text{pgcd}(r_i, r_{i+1})$ pour tout i et que les éléments de la i -ième itération vérifient $s_i f + t_i g = r_i$. En particulier, $s_\ell f + t_\ell g = r_\ell = \text{pgcd}(f, g)$. L'écriture matricielle de l'itération

$$\begin{bmatrix} s_i & t_i \\ s_{i+1} & t_{i+1} \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ 1 & -q_i \end{bmatrix} \times \begin{bmatrix} s_{i-1} & t_{i-1} \\ s_i & t_i \end{bmatrix}$$

permet de déduire aisément que $s_i t_{i+1} - t_i s_{i+1} = (-1)^i$ et en particulier que s_i et t_i sont premiers entre eux, quel que soit i . Enfin, cela entraîne une dernière propriété utile : $\text{pgcd}(r_i, t_i) = \text{pgcd}(f, t_i)$.

Le résultat suivant montre que les cofacteurs t_i dans AEE ont des degrés modérés et bien maîtrisés.

Lemme 1. Soient $n_0 := \deg(f) > n_1 := \deg(g) > n_2 > \dots > n_\ell$ les degrés des restes r_i dans AEE. Alors :

$$\deg(t_i) = n - n_{i-1} \quad \text{pour} \quad 1 \leq i \leq \ell + 1.$$

DÉMONSTRATION. On montre par récurrence sur $k = 1, \dots, \ell + 1$ que

$$(1) \quad \deg(t_i) = n - n_{i-1} \quad \text{pour} \quad 1 \leq i \leq k.$$

Pour $k = 1$, on a en effet que $\deg(t_1) = 0$. Prouvons maintenant $k \rightarrow k + 1$. On a d'abord $\deg(t_{k-1}) = n - n_{k-2} < n - n_{k-1} = \deg(t_k)$. Du coup, $\deg(t_{k+1}) = \max\{\deg(t_{k-1}), \deg(q_k t_k)\}$ vaut $\deg(q_k t_k) = \deg(q_k) + \deg(t_k) = (n_{k-1} - n_k) + (n - n_{k-1}) = n - n_k$. \square

Remarquons que, dans une situation générique, les quotients successifs q_i ont tous degré 1 et donc $n_{i+1} = n_i - 1$ (la suite des restes est appelée *normale*) et l'on a $\deg(s_i) = i - 2$ et $\deg(t_i) = i - 1$.

3. Calcul de la reconstruction rationnelle

Nous sommes en mesure de prouver que le problème de reconstruction rationnelle (RR) peut être résolu à l'aide de l'algorithme AEE.

Théorème 3. Soit $f \in \mathbb{K}[X]$ de degré $n > 0$ et soit $g \in \mathbb{K}[X]$ de degré $< n$. Soit $k \in \{1, 2, \dots, n\}$ et soit (r_j, s_j, t_j) la j -ième ligne dans AEE(f, g), où j est choisi minimal tel que $\deg(r_j) < k$.

1. Il existe un couple $(r, t) \neq (0, 0)$ solution de (RRS), à savoir $r = r_j$ et $t = t_j$. Si de plus $\text{pgcd}(r_j, t_j) = 1$, alors (r, t) est aussi solution du problème (RR).
2. Si (RR) admet une solution et si $r/t \in \mathbb{K}(X)$ en est une forme irréductible, alors il existe une constante $\alpha \in \mathbb{K} \setminus \{0\}$ telle que $r = \alpha r_j$ et $t = \alpha t_j$.

En particulier, le problème (RR) admet une solution si et seulement si $\text{pgcd}(r_j, t_j) = 1$.

DÉMONSTRATION. (1) Par propriété, r_j vaut $s_j f + t_j g \equiv t_j g$ modulo f . Par ailleurs, le Lemme 1 montre que $\deg(t_j) = n - \deg(r_{j-1})$, ce qui, par la minimalité de j , est borné par $n - k$. Donc $(r, t) = (r_j, t_j)$ vérifie bien (RRS).

De plus, on a que $\text{pgcd}(f, t_j) = \text{pgcd}(r_j, t_j)$, donc si ce dernier vaut 1, alors t_j est inversible modulo f et donc $(r, t) = (r_j, t_j)$ vérifie aussi (RR).

(2) Pour la seconde partie, supposons que (r, t) est une solution de (RR). Alors r s'écrit $sf + tg$ pour un certain $s \in \mathbb{K}[X]$.

Nous allons prouver que $(r, t) = (\alpha r_j, \alpha t_j)$ pour un certain $\alpha \in \mathbb{K}[X] \setminus \{0\}$. Par l'hypothèse, r et t étant premiers entre eux, on en déduira que α est une constante de $\mathbb{K} \setminus \{0\}$, et que donc $\text{pgcd}(r_j, t_j) = 1$.

Il nous reste à prouver que $(r, t) = (\alpha r_j, \alpha t_j)$. Nous commençons par montrer que $s_j t = s t_j$. Supposons le contraire et considérons le système linéaire (de Cramer)

$$\begin{pmatrix} s_j & t_j \\ s & t \end{pmatrix} \times \begin{pmatrix} f \\ g \end{pmatrix} = \begin{pmatrix} r_j \\ r \end{pmatrix}.$$

La matrice du système étant inversible, on peut appliquer la règle de Cramer et déduire :

$$f = \frac{\begin{vmatrix} r_j & t_j \\ r & t \end{vmatrix}}{\begin{vmatrix} s_j & t_j \\ s & t \end{vmatrix}}.$$

L'hypothèse $\deg(r_j) < k \leq \deg(r_{j-1})$ et le Lemme 1 montrent que le membre droit de l'égalité précédente a un degré majoré par

$$\begin{aligned} \deg(r_j t - r t_j) &\leq \max\{k - 1 + \deg(t), \deg(r) + n - \deg(r_{j-1})\} \\ &\leq \max\{n - 1, (k - 1) + n - \deg(r_{j-1})\} = n - 1, \end{aligned}$$

ce qui contredit $\deg(f) = n$.

Maintenant, l'égalité $s_j t = s t_j$ implique que t_j divise $s_j t$ et puisque s_j et t_j sont premiers entre eux, il s'ensuit que t_j divise t . On écrit $t = \alpha t_j$, avec $\alpha \neq 0$ (car $t \neq 0$). Alors, on a $s t_j = t s_j = \alpha s_j t_j$, donc $s = \alpha s_j$.

Enfin, $r = s f + t g = \alpha(s_j f + t_j g) = \alpha r_j$. □

L'algorithme qui s'ensuit est donné en figure 2; à noter la ressemblance avec l'algorithme AEE.

ARR(f, g, k)

Entrée : $f, g \in \mathbb{K}[X]$ avec $n = \deg(f) > \deg(g)$, et $k \in \{1, \dots, n\}$.

Sortie : Une solution (r, t) de (RR), ou 0 si une telle solution n'existe pas.

1. $r_0 := f$; $s_0 := 1$; $t_0 := 0$; $r_1 := g$; $s_1 := 0$; $t_1 := 1$; $i := 1$.
2. Tant que $\deg(r_i) \geq k$, faire :
 - $(q_i, r_{i+1}) := \text{QuotRem}(r_{i-1}, r_i)$; # la division euclidienne
 - $s_{i+1} := s_{i-1} - q_i s_i$; $t_{i+1} := t_{i-1} - q_i t_i$;
 - $i := i + 1$
3. Si $\text{pgcd}(f, t_i) = 1$, renvoyer (r_i, t_i) ; sinon renvoyer 0.

FIGURE 2. Algorithme de reconstruction rationnelle.

4. Approximants de Padé

On rappelle que si $n > 0$, $k \in \{1, 2, \dots, n\}$ et si $g \in \mathbb{K}[X]$ est de degré $< n$, alors un *approximant de Padé pour g de type $(k - 1, n - k)$* est une fraction rationnelle

$r/t \in \mathbb{K}[X]$ telle que

$$(2) \quad X \nmid t, \quad \deg(r) < k, \quad \deg(t) \leq n - k \quad \text{et} \quad \frac{r}{t} \equiv g \pmod{X^n}.$$

Comme corollaire du Th. 3 ($f = X^n$) nous tirons

Corollaire 1. *Soit $g \in \mathbb{K}[X]$ de degré $< n$. Soit (r_j, s_j, t_j) la j -ième ligne dans $\text{AEE}(X^n, g)$, où j est choisi minimal tel que $\deg(r_j) < k$. Alors :*

1. *L'équation (2) est soluble si et seulement si $\text{pgcd}(r_j, t_j) = 1$.*
2. *Si $\text{pgcd}(r_j, t_j) = 1$, alors r_j/t_j est l'unique approximant de Padé pour g de type $(k-1, n-k)$.*

L'algorithme qui s'en déduit est un cas particulier de l'algorithme ARR, en prenant $f = X^n$ et en remplaçant le test $\text{pgcd}(f, t_i) = 1$ de l'étape (3) par $t_i(0) \neq 0$.

5. Algorithme de Berlekamp-Massey

L'algorithme décrit dans cette section permet de deviner des récurrences à coefficients constants d'ordre arbitraire à partir de termes consécutifs de la suite. Il peut être vu comme la base de la fonction `listtorec` de `gfun` dans le cas à coefficients constants.

On se donne dans un corps \mathbb{K} les $2n$ premiers éléments d'une suite récurrente linéaire $(a_k)_{k \geq 0}$ pour laquelle on sait qu'il existe un polynôme générateur de degré $d \leq n$, c'est-à-dire un polynôme

$$f(X) = f_d X^d + \cdots + f_0, \quad \text{tel que} \quad f_d a_{i+d} + \cdots + f_0 a_i = 0, \quad \text{pour tous } i \geq 0.$$

Le problème est de calculer le polynôme minimal (*i.e.* le polynôme générateur de degré minimal) de $(a_k)_{k \geq 0}$. Le résultat suivant montre comment ramener ce problème à de l'approximation de Padé.

Lemme 2. *Soit $A(X) = \sum_{i \geq 0} a_i X^i$ la série génératrice de la suite (a_k) . Soit $f \in \mathbb{K}[X] \setminus \{0\}$ de degré d et soit $f^* = f(1/X)X^d$ son polynôme réciproque. Les assertions suivantes sont équivalentes :*

- (i) *f est un polynôme générateur de (a_k) ;*
- (ii) *$A(X) = h/f^*$ pour un certain $h \in \mathbb{K}[X]$ avec $\deg(h) < d$.*

De plus, si f est le polynôme minimal de (a_k) , alors $d = \max\{1 + \deg(h), \deg(f^)\}$ et $\text{pgcd}(h, f^*) = 1$.*

DÉMONSTRATION. Pour l'équivalence, on utilise uniquement le fait que le coefficient de X^{d+i} dans $f^* \cdot A(X)$ est égal à $f_d a_{i+d} + \cdots + f_0 a_i$ et que $f^*(0) \neq 0$, car $f^*(0) = \text{lc}(f)$.

Soit maintenant f le générateur minimal de (a_k) . On a $\deg(f^*) \leq d$ avec égalité si et seulement si $f_0 \neq 0$, c'est-à-dire $X \nmid f$. Donc $d \geq \max\{1 + \deg(h), \deg(f^*)\}$. Supposons par l'absurde que cette inégalité est stricte. Alors $X \mid f$ et on a que f/X est aussi polynôme générateur de (a_k) , ce qui contredit la minimalité de f . Donc $d = \max\{1 + \deg(h), \deg(f^*)\}$.

Soit enfin $u := \text{pgcd}(h, f^*)$. Alors $f_u := f/u^*$ est un polynôme de degré $d - \deg(u)$ qui est générateur de (a_k) , car $f^*/u = (f_u)^*$ et $(f^*/u) \cdot A(X) = h/u$ est un polynôme de degré $< d - \deg(u)$. Par la minimalité, cela implique que $\deg(u) = 0$, donc h et f^* sont bien premiers entre eux. \square

Le Lemme 2 montre que calculer le générateur minimal de (a_k) équivaut à la résolution du problème de Padé

$$(3) \quad \frac{r}{t} \equiv A \pmod{X^{2n}}, \quad X \nmid t, \quad \deg(r) < n, \quad \deg(t) \leq n \quad \text{et} \quad \text{pgcd}(r, t) = 1.$$

En effet, le Lemme 2 implique que $(r, t) = (h, f^*)$ est solution de (3). L'algorithme qui s'en déduit est donné en Fig. 3.

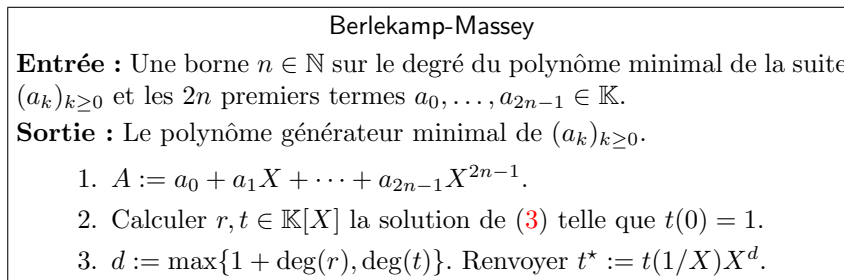


FIGURE 3. L'algorithme de Berlekamp-Massey

6. Interpolation rationnelle de Cauchy

L'interpolation des fractions rationnelles, appelée aussi *interpolation de Cauchy*, est une généralisation naturelle de l'interpolation polynomiale (de Lagrange).

Soit $k \in \{1, \dots, n\}$ et soient u_0, \dots, u_{n-1} des points distincts de \mathbb{K} et $v_0, \dots, v_{n-1} \in \mathbb{K}$. On cherche une fraction rationnelle $r/t \in \mathbb{K}(X)$ avec $r, t \in \mathbb{K}[X]$ tels que

$$(4) \quad t(u_i) \neq 0, \quad \frac{r(u_i)}{t(u_i)} = v_i, \quad \text{pour} \quad 0 \leq i \leq n-1, \quad \deg(r) < k, \quad \deg(t) \leq n-k.$$

L'intérêt de l'interpolation rationnelle est que souvent elle fournit de meilleures approximations numériques que l'interpolation polynomiale. Une autre application est la découverte de récurrences à coefficients polynomiaux d'ordre 1 (voir Corollaire 2 ci-dessous). Le cas $k = n$ correspond à l'interpolation polynomiale et admet toujours une solution; si $k < n$, le problème peut ne pas avoir de solution. Par contre, s'il admet une solution, elle est unique.

Le problème (4) est un cas particulier de reconstruction rationnelle. En effet, si l'on note g l'unique polynôme de degré $< n$ tel que $g(u_i) = v_i$, alors (4) équivaut à la reconstruction rationnelle de g modulo $f := (X - u_0) \cdots (X - u_{n-1})$, puisque $\forall i, r(u_i) = t(u_i)g(u_i)$ équivaut à $r \equiv tg \pmod{f}$.

La procédure de décision et de calcul est la suivante : on calcule le polynôme interpolant g et le polynôme $f = (X - u_0) \cdots (X - u_{n-1})$. On calcule les éléments (r_j, t_j) de la j -ième ligne de AEE(f, g) avec j minimal tel que $\deg(r_j) < k$. Si $\text{pgcd}(r_j, t_j) = 1$, alors r_j/t_j est l'unique forme canonique de la fraction rationnelle cherchée. Sinon, le problème (4) n'a pas de solution (car $\text{pgcd}(r_j, t_j) \neq 1$ implique $\text{pgcd}(f, t_j) \neq 1$, i.e., $t_j(u_i) = 0$ pour un i .) L'algorithme qui s'en déduit est obtenu par spécialisation de l'algorithme ARR en prenant $f = \prod (X - u_i)$.

Le corollaire suivant peut être vu comme la base de la fonction listtorec de gfun dans le cas des suites hypergéométriques.

Corollaire 2 (Devinette de récurrences à coefficients polynomiaux d'ordre 1).
 Soit (p_n) une suite d'éléments non nuls de \mathbb{K} vérifiant une récurrence $a(n)p_{n+1} + b(n)p_n = 0$ avec $a(X), b(X) \in \mathbb{K}[X]$ premiers entre eux. Si $b(j) \neq 0$ pour $0 \leq j \leq 2d + 1$ et si l'on connaît une borne d sur le degré de $a(X)$ et de $b(X)$, alors on peut déterminer a et b par un calcul d'interpolation rationnelle, à partir des $2d + 1$ premiers termes de la suite (p_n) .

DÉMONSTRATION. Prendre $k = d + 1$, $n = 2d + 1$, $u_i = i$ et $v_i = -p_i/p_{i+1}$ pour $i = 0, 1, \dots, 2d$. \square

Complexité

Tous les algorithmes présentés dans ce cours ont une complexité arithmétique *quadratique* : ils utilisent $O(n^2)$ opérations $(+, -, \times, /)$ dans le corps de base \mathbb{K} . Notons qu'il existe des versions rapides de ces algorithmes, reposant sur la multiplication polynomiale par FFT (transformée de Fourier rapide). Leur complexité est de $O(n \log^2(n) \log \log(n))$ opérations dans \mathbb{K} .

À titre de comparaison, l'approche directe par algèbre linéaire demande $O(n^3)$ opérations si le pivot de Gauss est employé.

Un gros déterminant

Soit A_n la matrice $n \times n$ dont les coefficients sont tous nuls sauf les nombres premiers 2,3,5,7,... sur la diagonale et le nombre 1 dans toutes les positions a_{ij} où $|i - j| = 1, 2, 4, 8, \dots$. L'objectif de ce TP¹ est le calcul du déterminant de A_n .

1. Approche directe

1. Écrire une procédure prenant en entrée n et le type des coefficients souhaité (integer ou float) et renvoyant la matrice A_n . La matrice doit être créée en un nombre d'opérations *linéaire* en n .
2. Estimer empiriquement la complexité de la procédure `LinearAlgebra[Determinant]` sur ces matrices en comparant les temps de calculs pour $|A_k|$ et $|A_{2k}|$ pour des valeurs de k bien choisies en fonction du type des coefficients.

2. La méthode de Wiedemann

La méthode de Wiedemann calcule le polynôme minimal de matrices pour lesquelles le produit matrice-vecteur est moindre que quadratique, à l'aide d'un approximant de Padé. Lorsque (comme c'est fréquemment le cas) le polynôme minimal est aussi le polynôme caractéristique, le déterminant est obtenu comme terme constant au signe près.

Pour ce calcul, la méthode commence par tirer aléatoirement deux vecteurs v et w , puis calcule la suite de scalaires $u_i := wA^i v$ pour $i = 0, \dots, 2n$. Ce calcul n'utilise que des produits scalaires et des multiplication de A par des vecteurs. Si le polynôme minimal est $\chi(A) = t^n + c_1 t^{n-1} + \dots + c_n$, alors la suite u_i vérifie la récurrence

$$u_{i+n} + c_1 u_{i+n-1} + \dots + c_n u_i = 0, \quad i \geq 0.$$

Dans un second temps, la méthode reconstruit cette récurrence à partir des $2n + 1$ premières valeurs de u_i par un approximant de Padé de type $(n - 1, n)$ de la série $S_n = u_0 + \dots + u_{2n} z^{2n} + O(z^{2n+1})$.

En toute généralité, pour de mauvais choix de v et w , il est possible (mais peu probable) d'obtenir un diviseur de $\chi(A)$ et non $\chi(A)$ lui-même. Dans ce cas, il suffit de reprendre le calcul avec un autre choix de (v, w) . Le résultat est correct si le degré obtenu est n .

3. Écrire une procédure qui prend en entrée A et une variable z et calcule la série tronquée S_n correspondante. (On obtient un vecteur aléatoire par `LinearAlgebra[RandomVector]`).
4. Estimer empiriquement la complexité de ce calcul.

1. Il s'agit d'une variante d'un des problèmes étudiés dans le livre *The SIAM 100-Digit Challenge*, SIAM Press, 2004.

5. Calculer le déterminant de la matrice A_{10} par la méthode de Wiedemann sur la matrice A_{10} , d'abord en flottants, puis en entiers.

3. Calculs exacts

Pour des raisons de stabilité numérique, la méthode de Wiedemann est réservée au calcul exact (soit sur des entiers, soit sur des entiers modulaires).

6. Écrire une procédure qui prend en entrée les $2n$ premiers coefficients d'une série tronquée ainsi qu'un entier n et renvoie le dénominateur d'un approximant de Padé de type $(n-1, n)$ calculé par l'algorithme d'Euclide étendu. Dans un premier temps on ne se préoccupera pas du type des coefficients.
7. Comparer cette implémentation à `numapprox[pade]` pour calculer $|A_{10}|$.

4. Calculs modulaires exacts

Pour éviter de calculer des entiers intermédiaires trop gros, il est plus efficace de calculer le déterminant modulo plusieurs nombres premiers et de le reconstruire par le théorème des restes chinois. Il faut pour cela disposer d'une borne facile à calculer sur ce déterminant, et on pourra prendre le produit B des éléments diagonaux. Il suffit alors de disposer de nombres premiers (p_1, \dots, p_k) tels que $p_1 \cdots p_k \geq B$. Pour rendre les calculs efficaces, ils doivent être assez gros (pour diminuer k) mais pas trop (pour que les calculs tiennent dans un mot mémoire). Typiquement on prend p_1 le plus grand nombre premier inférieur à 2^{32} et p_2, p_3, \dots les précédents.

8. Récrire la procédure de calcul d'approximants de Padé pour des coefficients modulaires en utilisant les opérations fournies par le package `modp1`.
9. Estimer empiriquement la complexité de ce calcul d'approximants de Padé.
10. Écrire la procédure générale de calcul de déterminant modulaire, en calculant la série sur les entiers, puis les images modulaires du déterminant et en reconstruisant ce déterminant (par `chrem`); estimer sa complexité empirique, et conclure.

Les approximations de Padé-Hermite ou la reconstruction d'équations

Résumé

Les approximations de Padé-Hermite sont une généralisation des approximations de Padé. Leur calcul peut s'effectuer grâce à un algorithme qui peut être vu comme une généralisation de l'algorithme d'Euclide étendu. Il permet de reconstruire une équation linéaire à coefficients polynomiaux reliant des séries formelles. Ce chapitre définit ces approximations, prouve leur existence, présente un algorithme pour les calculer et donne quelques applications.

1. Premières définitions et premiers résultats

Dans toute la suite, \mathbb{K} désigne un corps quelconque.

Définition 1 (approximation de Padé-Hermite). *Soit $n \geq 1$, $\mathbf{F} = {}^t(f_1, \dots, f_n)$ un vecteur de séries formelles de $\mathbb{K}[[X]]$ et soit $\mathbf{d} = (d_1, \dots, d_n) \in \mathbb{N}^n$. Un vecteur non nul $\mathbf{P} = (P_1, \dots, P_n)$ de polynômes de $\mathbb{K}[X]$ est appelé un « approximation de Padé-Hermite de type \mathbf{d} de \mathbf{F} » si :*

1. La valuation $\text{val}(\mathbf{P} \cdot \mathbf{F})$ de la série $\mathbf{P} \cdot \mathbf{F} = \sum_{i=1}^n P_i f_i$ est au moins égale à $\sigma := \sum (d_i + 1) - 1$;
2. $\deg(P_i) \leq d_i$ pour tout $1 \leq i \leq n$.

L'entier σ est alors appelé l'ordre de l'approximation.

Par exemple, dans la terminologie du cours précédent, si $r/t \in \mathbb{K}(X)$ est un approximation de Padé de type $(k, n-k)$ de $g \in \mathbb{K}[[X]]$, alors (r, t) est un approximation de Padé-Hermite pour $(-1, g)$, de type $(k, n-k)$. Plus généralement, il n'est pas difficile de se convaincre que le problème de reconstruction rationnelle RRS introduit au cours précédent

(RRS) : Calculer $r, t \in \mathbb{K}[X]$ tels que : $\deg(r) < k$, $\deg(t) \leq n - k$ et $r \equiv tg \pmod{f}$,

revient à un calcul d'approximation de Padé-Hermite (r, t, s) de $(-1, g, f)$ de type $(k-1, n-k, n-k+1)$.

Le premier résultat de ce cours montre l'existence des approximations de Padé-Hermite et fournit également un premier algorithme pour leur calcul.

Théorème 4. *Tout vecteur de séries formelles $\mathbf{F} = (f_1, \dots, f_n) \in \mathbb{K}[[X]]^n$ admet un approximation de Padé-Hermite de type $\mathbf{d} = (d_1, \dots, d_n) \in \mathbb{N}^n$ donné.*

DÉMONSTRATION. On procède par coefficients indéterminés. En écrivant $P_i = \sum_{j=0}^{d_i} p_{ij} X^j$, on obtient un système linéaire homogène à $\sigma = \sum_i (d_i + 1) - 1$ équations en les $\sigma + 1$ inconnues p_{ij} . Puisqu'il a moins d'équations que d'inconnues, ce système admet forcément une solution non-triviale. \square

L'algorithme qui découle de la preuve du Th. 4 repose sur la recherche d'un élément non-trivial dans le noyau d'une matrice à coefficients dans \mathbb{K} , de taille $\sigma \times (\sigma + 1)$. En utilisant de l'élimination gaussienne, cet algorithme est donc de complexité cubique en σ . Le but de la suite de ce cours est de présenter un algorithme plus efficace, dû à Harm Derksen, de complexité seulement quadratique en σ . En anticipant un peu, cet algorithme revient à effectuer une sorte de pivot de Gauss sur une matrice à coefficients dans $\mathbb{K}[X]$, mais de taille bien plus petite que σ (seulement linéaire en $\max(\mathbf{d})$). Dans le cas particulier $n = 2$, l'algorithme de Derksen a essentiellement la même complexité que le calcul d'approximants de Padé via l'algorithme d'Euclide étendu.

Le problème d'approximation de Padé-Hermite a été introduit par Hermite en 1873 dans sa preuve de la transcendance de e , qui utilise le choix très particulier $\mathbf{F} = (1, e^X, e^{2X}, \dots)$. Deux autres cas particuliers importants sont : les « approximants algébriques » avec $\mathbf{F} = (1, f, f^2, \dots)$ et les « approximants différentiels » avec $\mathbf{F} = (f, f', f'', \dots)$, où f est une série de $\mathbb{K}[[X]]$ donnée. En Maple, le calcul d'approximants algébriques et différentiels se fait grâce aux fonctions `seriestoalgeq` et `seriestodiffeq` du package `gfun`. Le problème général d'approximation peut se traiter en utilisant la fonction `hermite_pade` du package `numapprox`. Ces trois fonctions utilisent toutes (des variantes de) l'algorithme de Derksen.

Il existe diverses généralisations utiles du problème d'approximation de Padé-Hermite, par exemple les « approximants de Padé-Hermite simultanés et matriciels », ou encore des approximants modulo un polynôme arbitraire de degré $\sigma = \sum_i (d_i + 1) - 1$ au lieu de X^σ . Des algorithmes de complexité quadratique existent pour toutes ces généralisations.

Il existe des algorithmes encore plus rapides, de complexité *essentiellement linéaire* en σ . Ces algorithmes, de type *diviser pour régner* et reposant sur la multiplication rapide de polynômes via la *transformée de Fourier rapide* (FFT), dépassent le cadre de ce cours.

2. Algorithme de Derksen : idées et résultats préliminaires

Pour simplifier la présentation, on se restreint dans la suite au cas où le type de l'approximant cherché est de la forme $\mathbf{d} = (d, \dots, d) \in \mathbb{N}^n$, pour un certain $d \in \mathbb{N}$.

L'idée de l'algorithme de Derksen est de construire non pas un seul approximant de Padé-Hermite, mais toute une famille de tels approximants, et cela de manière incrémentale. Plus exactement, pour $s = 0, 1, \dots$, il construit une base d'une forme spéciale, appelée « base minimale », du $\mathbb{K}[X]$ -module

$$V_s := \{\mathbf{P} \in \mathbb{K}[X]^n \mid \text{val}(\mathbf{P} \cdot \mathbf{F}) \geq s\}.$$

Comme nous le verrons plus loin, une telle base de V_σ pour $\sigma = nd + n - 1$ contiendra alors nécessairement un approximant de Padé-Hermite de type $\mathbf{d} = (d, \dots, d)$ de \mathbf{F} .

Observons que, grâce aux inclusions $X^s \mathbb{K}[X]^n \subseteq V_s \subseteq \mathbb{K}[X]^n$, le module V_s est libre de rang n . Précisons ce que l'on entend par « base minimale ». Pour ce

faire, nous introduisons d'abord une notion de *degré* et de *type* d'un vecteur de polynômes.

Définition 2 (degré et type d'un vecteur de polynômes). *Pour tout vecteur de polynômes $\mathbf{P} = (P_1, \dots, P_n) \in \mathbb{K}[X]^n$, on définit*

$$\deg(\mathbf{P}) = \max\{\deg(P_1), \dots, \deg(P_n)\} \quad \text{et} \quad \text{type}(\mathbf{P}) = \max\{i \mid \deg(\mathbf{P}) = \deg(P_i)\}.$$

Définition 3 (base minimale). *Soit $V \subseteq \mathbb{K}[X]^n$ un sous-module libre de rang n . Une suite $\mathbf{Q}_1, \dots, \mathbf{Q}_n$ est appelée « base minimale » de V si pour tout i , le vecteur \mathbf{Q}_i est non-nul, de type i , et de degré minimal parmi les éléments de $V \setminus \{0\}$ de type i .*

Le résultat suivant précise le lien entre une base minimale et l'approximation de Padé-Hermite.

Lemme 1. *Soit $d \geq 1$. Supposons que $\mathbf{Q}_1, \dots, \mathbf{Q}_n$ est une base minimale de V_{nd+n-1} . Soit ℓ tel que $\deg(\mathbf{Q}_\ell)$ est minimal. Alors \mathbf{Q}_ℓ est un approximant de Padé-Hermite de type (d, \dots, d) pour \mathbf{F} .*

DÉMONSTRATION. Par Th. 4, il existe un vecteur non-nul $\mathbf{P} \in V_{nd+n-1}$ tel que $\deg(\mathbf{P}) \leq d$. Soit i le type de \mathbf{P} . On a alors la suite d'inégalités :

$$\deg(\mathbf{Q}_\ell) \leq \deg(\mathbf{Q}_i) \leq \deg(\mathbf{P}) \leq d,$$

qui prouve que \mathbf{Q}_ℓ est un approximant de Padé-Hermite de type (d, \dots, d) de \mathbf{F} . \square

Le résultat suivant montre qu'une base minimale est nécessairement une base du $\mathbb{K}[X]$ -module V au sens usuel.

Théorème 5. *Soit $V \subseteq \mathbb{K}[X]^n$ un sous-module libre de rang n . Toute base minimale de V est une base du $\mathbb{K}[X]$ -module V .*

DÉMONSTRATION. Montrons d'abord qu'il s'agit d'un système de générateurs. Soit $W := \mathbb{K}[X]\mathbf{Q}_1 + \dots + \mathbb{K}[X]\mathbf{Q}_n \subseteq V$. On suppose par l'absurde que $V \neq W$. Soit $\mathbf{P} \in V \setminus W$ un élément minimal dans $V \setminus W$ pour l'ordre $\mathbf{P} < \mathbf{Q}$ défini par

$$(1) \quad \begin{aligned} &\deg(\mathbf{P}) < \deg(\mathbf{Q}), \quad \text{ou} \\ &\deg(\mathbf{P}) = \deg(\mathbf{Q}) \quad \text{et} \quad \text{type}(\mathbf{P}) < \text{type}(\mathbf{Q}). \end{aligned}$$

Autrement dit, \mathbf{P} est de type minimal parmi les éléments de degré minimal de $V \setminus W$.

Soit i le type de \mathbf{P} . Puisque $\text{type}(\mathbf{P}) = \text{type}(\mathbf{Q}_i)$ et $\deg(\mathbf{P}) \geq \deg(\mathbf{Q}_i)$, il existe un monôme $q \in \mathbb{K}[X]$ de degré $\deg(q) = \deg(\mathbf{P}) - \deg(\mathbf{Q}_i)$, tel que $\text{type}(\mathbf{P} - q\mathbf{Q}_i) < \text{type}(\mathbf{P})$. Du coup, comme $\deg(\mathbf{P} - q\mathbf{Q}_i) \leq \deg(\mathbf{P})$, on obtient que

$$\mathbf{P} - q\mathbf{Q}_i < \mathbf{P}.$$

Par la minimalité de \mathbf{P} , il s'ensuit que $\mathbf{P} - q\mathbf{Q}_i$ appartient à W , donc $\mathbf{P} \in W$, ce qui contredit le choix de \mathbf{P} .

Pour conclure la preuve, montrons que les \mathbf{Q}_i forment une famille libre. Si $\sum_i a_i \mathbf{Q}_i = 0$ est une combinaison polynomiale nulle des \mathbf{Q}_i , on a que pour tout i , le vecteur $a_i \mathbf{Q}_i$ est de type i . L'assertion découle du lemme suivant. \square

Lemme 2. *Si \mathbf{P} et \mathbf{Q} sont de type différent, alors $\text{type}(\mathbf{P} + \mathbf{Q}) \in \{\text{type}(\mathbf{P}), \text{type}(\mathbf{Q})\}$.*

DÉMONSTRATION. Supposons $j = \text{type}(\mathbf{P}) > i = \text{type}(\mathbf{Q})$. Si $\deg(\mathbf{P}) \geq \deg(\mathbf{Q})$, alors $\text{type}(\mathbf{P} + \mathbf{Q}) = j$ et si $\deg(\mathbf{P}) < \deg(\mathbf{Q})$, alors $\text{type}(\mathbf{P} + \mathbf{Q}) = i$. \square

3. Algorithme de Derksen : fonctionnement

L'idée de l'algorithme est de construire de proche en proche une base minimale de V_s , partant de la base minimale des $\mathbf{Q}_k = (0, 0, \dots, 0, 1, 0, \dots, 0)$ (avec 1 en position k) de V_0 . Cf. Lemme 1, l'élément de degré minimal dans une base minimale de V_{nd+n-1} fournit un approximant de Padé-Hermite de type (d, \dots, d) de \mathbf{F} .

Le résultat suivant montre comment construire une base minimale de V_{s+1} à partir d'une base minimale de V_s .

Théorème 6. *Soit $\mathbf{Q}_1, \dots, \mathbf{Q}_n$ une base minimale de*

$$V_s = \{\mathbf{P} \in \mathbb{K}[X]^n \mid \text{val}(\mathbf{P} \cdot \mathbf{F}) \geq s\}.$$

1. *Si $\text{val}(\mathbf{Q}_i \cdot \mathbf{F}) \geq s + 1$ quel que soit i , alors $V_{s+1} = V_s$ et $\{\mathbf{Q}_1, \dots, \mathbf{Q}_n\}$ est une base minimale de V_{s+1} .*
2. *Supposons que $1 \leq i \leq n$ est tel que les deux conditions suivantes soient réunies :*
 - $\text{val}(\mathbf{Q}_i \cdot \mathbf{F}) = s$;
 - *Si $\text{val}(\mathbf{Q}_\ell \cdot \mathbf{F}) = s$ pour un $\ell \neq i$, alors $\mathbf{Q}_i < \mathbf{Q}_\ell$, où $<$ est l'ordre (1).*

Alors :

- (a) *Pour $\ell \neq i$, il existe un scalaire $\lambda_\ell \in \mathbb{K}$ tel que $\tilde{\mathbf{Q}}_\ell := \mathbf{Q}_\ell - \lambda_\ell \mathbf{Q}_i$ vérifie $\text{val}(\tilde{\mathbf{Q}}_\ell \cdot \mathbf{F}) > s$.*
- (b) *En posant $\tilde{\mathbf{Q}}_i = X\mathbf{Q}_i$, la suite $\tilde{\mathbf{Q}}_1, \dots, \tilde{\mathbf{Q}}_n$ forme une base minimale de V_{s+1} .*

DÉMONSTRATION. (1) L'inclusion $V_{s+1} \subseteq V_s$ est évidente et inversement, cf. Th. 5, tout $\mathbf{P} \in V_s$ s'écrit comme combinaison linéaire $\sum_i a_i \mathbf{Q}_i$, ainsi $\mathbf{P} \cdot \mathbf{F} = \sum_i a_i (\mathbf{Q}_i \cdot \mathbf{F})$ est de valuation $\geq s + 1$, donc $\mathbf{P} \in V_{s+1}$.

(2a) Si $\text{val}(\mathbf{Q}_\ell \cdot \mathbf{F}) > s$, on pose $\lambda_\ell = 0$; si $\text{val}(\mathbf{Q}_\ell \cdot \mathbf{F}) = s$, alors $\mathbf{Q}_\ell \cdot \mathbf{F} = c_\ell X^s + \dots$ et $\mathbf{Q}_i \cdot \mathbf{F} = c_i X^s + \dots$, avec $c_i \neq 0$, et alors $\lambda_\ell := c_\ell / c_i$ convient.

Pour (2b), commençons par montrer que la suite $\tilde{\mathbf{Q}}_1, \dots, \tilde{\mathbf{Q}}_{i-1}, \mathbf{Q}_i, \tilde{\mathbf{Q}}_{i+1}, \dots, \tilde{\mathbf{Q}}_n$ reste une base minimale de V_s . Il suffit pour cela de montrer que pour $\ell \neq i$, le vecteur $\tilde{\mathbf{Q}}_\ell$ a même type et même degré que \mathbf{Q}_ℓ . Si $\text{val}(\mathbf{Q}_\ell \cdot \mathbf{F}) > s$, c'est évident car $\lambda_\ell = 0$ et donc $\tilde{\mathbf{Q}}_\ell = \mathbf{Q}_\ell$. Sinon, le choix de i assure que $\mathbf{Q}_\ell > \mathbf{Q}_i$, et donc \mathbf{Q}_ℓ et $\mathbf{Q}_\ell - \lambda_\ell \mathbf{Q}_i$ ont le même degré et type.

Montrons maintenant que $(\tilde{\mathbf{Q}}_j)_j$ est une base minimale de V_{s+1} . Comme la multiplication par un polynôme ne change pas le type, celui de $\tilde{\mathbf{Q}}_i = X\mathbf{Q}_i$ est bien i . Il suffit donc de montrer que si $\mathbf{P} \in V_{s+1}$ est de type $\ell \in \{1, 2, \dots, n\}$, alors $\text{deg}(\mathbf{P}) \geq \text{deg}(\mathbf{Q}_\ell)$. Si $\ell \neq i$, ceci est une évidence : comme \mathbf{P} appartient à $V_{s+1} \subseteq V_s$, et comme la suite $\tilde{\mathbf{Q}}_1, \dots, \tilde{\mathbf{Q}}_{i-1}, \mathbf{Q}_i, \tilde{\mathbf{Q}}_{i+1}, \dots, \tilde{\mathbf{Q}}_n$ forme une base minimale de V_s , le degré de \mathbf{P} est nécessairement au moins égal à $\text{deg}(\tilde{\mathbf{Q}}_\ell) = \text{deg}(\mathbf{Q}_\ell)$.

Dans la suite de la preuve, on peut donc supposer que $\mathbf{P} \in V_{s+1}$ est de type i , le but étant de montrer que $\text{deg}(\mathbf{P}) \geq \text{deg}(X\mathbf{Q}_i)$. Par Th. 5, \mathbf{P} s'écrit $\mathbf{P} = \sum_{j \neq i} a_j \tilde{\mathbf{Q}}_j + a_i \mathbf{Q}_i$. Comme $\text{type}(\mathbf{P}) = i$, on a $a_i \neq 0$, par le Lemme 2. De plus, le degré de \mathbf{P} est égal à celui de $a_i \mathbf{Q}_i$, cf. Lemme 3 ci-dessous.

Comme $\text{val}(\mathbf{P} \cdot \mathbf{F}) > s$ et comme pour $k \neq i$, $\text{val}(\mathbf{Q}_k \cdot \mathbf{F}) > s$, on a nécessairement que $\text{val}(a_i) > 0$. En particulier $\text{deg}(a_i) > 0$ et donc $\text{deg}(\mathbf{P}) \geq 1 + \text{deg}(\mathbf{Q}_i)$. \square

Lemme 3. *Si $\text{type}(\mathbf{P}) = \text{type}(\mathbf{Q}) = i$ et $\text{type}(\mathbf{P} + \mathbf{Q}) < i$, alors $\text{deg}(\mathbf{P}) = \text{deg}(\mathbf{Q})$.*

Derksen

Entrée : $\mathbf{F} = (f_1, \dots, f_n) \in \mathbb{K}[[X]]^n$ et $d \geq 1$.

Sortie : Un approximant de Padé-Hermite de \mathbf{F} , de type (d, \dots, d) .

pour k de 1 à n définir
 $\mathbf{Q}_k := (0, 0, \dots, 0, 1, 0, \dots, 0)$, avec 1 en position k .

pour j de 0 à $nd + n - 2$ faire
 $i := 0$
pour k de 1 à n faire
 $c_k := \text{coeff}(\mathbf{Q}_k \cdot \mathbf{F}, j)$
si $c_k \neq 0$ et $(\mathbf{Q}_k < \mathbf{Q}_i$ ou $i = 0)$, alors $i := k$
si $i \neq 0$ alors
 $\mathbf{Q}_i := c_i^{-1} \mathbf{Q}_i$
pour k de 1 à n faire
si $k \neq i$ alors
 $\mathbf{Q}_k := \mathbf{Q}_k - c_k \mathbf{Q}_i$
 $\mathbf{Q}_i := X \mathbf{Q}_i$

$p := 1$
pour k de 2 à n faire
si $\deg(\mathbf{Q}_k) < \deg(\mathbf{Q}_p)$, alors $p := k$

Renvoyer \mathbf{Q}_p .

FIGURE 1. L'algorithme de Derksen

DÉMONSTRATION. Soient $\mathbf{P} = (P_1, \dots, P_n)$ et $\mathbf{Q} = (Q_1, \dots, Q_n)$. L'hypothèse entraîne les égalités $\deg(P_i) = \deg(\mathbf{P})$ et $\deg(Q_i) = \deg(\mathbf{Q})$. Cela implique que P_i et Q_i ont le même degré; sinon, le type de $\mathbf{P} + \mathbf{Q}$ serait égal à i . \square

L'algorithme qui se déduit de la conjonction du Lemme 1 et du Théorème 6 est donné en Fig. 1. Il est de complexité bornée par $O(n\sigma^2)$.

Remarquons que dans le cas « générique » (dit *normal*), la sortie $\mathbf{Q} = \mathbf{Q}_1, \dots, \mathbf{Q}_n$ est de degré

$$\begin{bmatrix} d+1 & d & \cdots & d & d \\ d+1 & d+1 & \cdots & d & d \\ \vdots & \ddots & \ddots & \vdots & \vdots \\ d+1 & d+1 & \cdots & d+1 & d \\ d & d & \cdots & d & d \end{bmatrix}.$$

C'est la dernière ligne \mathbf{Q}_n qui fournit l'approximant désiré.

4. Applications

On rappelle que si une série est connue comme rationnelle de degré au plus d , l'approximation de Padé de type (d, d) suffit pour reconstruire la fraction rationnelle. Une question naturelle est comment se généralise cette observation dans le cadre de l'approximation de Padé-Hermite. Les réponses sont partielles, mais entièrement satisfaisantes dans la pratique.

4.1. Recherche de relations à coefficients polynomiaux entre séries formelles. Supposons qu'il existe une combinaison linéaire $\sum_{i=1}^n P_i f_i = 0$, à coefficients des polynômes $P_i(X) \in \mathbb{K}[X]$ de degrés au plus d . Par ailleurs, supposons calculé un approximant de Padé-Hermite $\mathbf{Q} := (Q_1, \dots, Q_n)$ de $\mathbf{F} = (f_1, \dots, f_n)$ de type $\mathbf{d} = (d, \dots, d)$, via l'algorithme de Derksen. La question est donc : quel lien y a-t-il entre $\mathbf{P} = (P_1, \dots, P_n)$ et \mathbf{Q} ?

D'abord, dans le cas *générique*, la réponse est très simple : \mathbf{P} et \mathbf{Q} sont identiques, à un coefficient scalaire près. En effet, $\mathbf{Q} = \mathbf{Q}_n$ et $\mathbf{Q}_1, \dots, \mathbf{Q}_n$ est une base de V_{nd+n-1} dont les degrés sont décrits après l'algo Derksen. En particulier, \mathbf{P} doit être une combinaison linéaire des \mathbf{Q}_i . Des considérations sur les degrés, utilisant la forme bien particulière des degrés des \mathbf{Q}_i , mènent à la conclusion désirée.

En effet, si $(c_1(X), \dots, c_n(X)) \cdot {}^t(\mathbf{Q}_1, \dots, \mathbf{Q}_n) = \mathbf{P}$, alors $c_1 Q_{11} + \dots + c_n Q_{n1} = P_1$, et comme $\deg(Q_{11}) = d + 1$ et $\deg(Q_{j1}) = d$ pour $j > 1$ et $\deg(P_1) \leq d$, on obtient que $c_1 = 0$. De même, $c_2 = \dots = c_{n-1} = 0$ et c_n doit être une constante $c \in \mathbb{K}$ telle que $\mathbf{P} = c\mathbf{Q}$.

Dans le cas général, un argument de nothérianité permet de prouver que pour $D \gg 0$, V_{nD+n-1} contient la relation \mathbf{P} , qui sera trouvée par l'algorithme de Derksen. Seulement, on ne dispose pas de borne *a priori* en fonction de d , sur le D minimal avec cette propriété. En effet, si on note

$$W_j := \{\mathbf{Q} \in \mathbb{K}[[X]]^n \mid \text{val}(\mathbf{Q} \cdot \mathbf{F}) \geq j \text{ et } \deg(\mathbf{Q}) \leq \deg(\mathbf{P})\},$$

et $W_\infty = \bigcap_{j \geq 0} W_j$, alors W_∞ contient toutes les relations de F en degré d , et en particulier \mathbf{P} . Puisque $W_0 \supseteq W_1 \supseteq W_2 \supseteq \dots$ est une suite décroissante d'espaces vectoriels de dimension finie, elle est stationnaire, donc il existe un N tel que $W_\infty = W_N = W_{N+1} = \dots$. Le cas *normal* correspond à la situation où $\dim(W_{k+1}) = \dim(W_k) - 1$ pour chaque k (noter la ressemblance avec la normalité de la suite des restes dans l'algorithme d'Euclide).

4.2. Reconstruction d'équations algébriques et différentielles. Deux cas particuliers importants, pour lesquels on peut être encore plus précis, sont les approximants algébriques et différentiels.

Soit $f \in \mathbb{K}[[X]]$ une série formelle algébrique. Le problème d'approximation algébrique consiste à retrouver, à partir des premiers termes de f un polynôme $P(X, Y) \in \mathbb{K}[X, Y]$ tel que $P(X, f(X)) = 0$. Si un tel P , de degré d en X et n en Y existe, alors les coefficients des puissances de Y formeront un approximant de Padé-Hermite de type (d, \dots, d) du vecteur de séries $(1, f, \dots, f^n)$. La difficulté vient de ce que un calcul d'approximation de Padé-Hermite ne trouve, *a priori*, qu'un polynôme $Q \in \mathbb{K}[X, Y]$ tel que $Q(X, f(X)) = 0 \pmod{X^\sigma}$, où $\sigma = (n+1)d - 1$. On dit alors qu'on a *deviné* un polynôme annulateur Q de f .

Il se pose donc la question de la *certification a posteriori* de Q , c'est-à-dire qu'on souhaiterait déduire que non seulement $Q(X, f(X)) = 0 \pmod{X^\sigma}$, mais aussi $Q(X, f(X)) = 0$. Pour les approximants différentiels, la problématique est la même, la seule différence étant qu'on calcule un approximant de Padé-Hermite du vecteur des dérivées successives $(f, f', \dots, f^{(n)})$.

Le résultat suivant apporte une réponse partielle à la question de la certification. Il sera prouvé grâce à des techniques de résultant au Chapitre 10. Son avantage est qu'il ne dépend pas de l'algorithme utilisé pour produire l'approximant de Padé-Hermite.

Théorème 7. *Supposons que $f \in \mathbb{K}[[X]]$ est racine d'un polynôme irréductible de $\mathbb{K}[X, Y]$ de degré au plus d en X et au plus n en Y . Soit $\mathbf{Q} = (Q_0, Q_1, \dots, Q_n)$ un approximant de Padé-Hermite de type (d, \dots, d) de $\mathbf{F} = (1, f, \dots, f^n)$.*

Si $\text{val}(\mathbf{Q} \cdot \mathbf{F}) \geq 2dn$, alors $\mathbf{Q} \cdot \mathbf{F} = 0$, c'est-à-dire que f est racine du polynôme $Q = \sum_{i=1}^n Q_i Y^i$.

4.3. Reconstruction de récurrences. Soit $(a_n)_n \in \mathbb{K}^{\mathbb{N}}$ une suite vérifiant une récurrence linéaire à coefficients polynomiaux. Comment, à partir des premiers termes de la suite, retrouver les coefficients de cette récurrence ?

L'idée consiste à trouver une équation différentielle, à coefficients polynomiaux, portant sur la série génératrice de la suite. Cette équation peut être devinée grâce à la méthode proposée au paragraphe 4.2. Il suffit ensuite de passer de l'équation différentielle à l'expression de la récurrence, ce qui n'est pas trivial mais purement formel et sera vu au Chapitre 8.

5. Approximants de Padé-Hermite de type arbitraire

Pour calculer un approximant de Padé-Hermite de type $\mathbf{d} = (d_1, \dots, d_n)$, il suffit de remplacer dans l'algorithme donné en Fig. 1, deg par $\text{deg}_{\mathbf{d}}$ et type par $\text{type}_{\mathbf{d}}$, où :

$$\text{deg}_{\mathbf{d}}(\mathbf{P}) = \max\{\text{deg}(P_1) - d_1, \dots, \text{deg}(P_n) - d_n\}$$

et

$$\text{type}_{\mathbf{d}}(\mathbf{P}) = \max\{i \mid \text{deg}(P_i) - d_i = \text{deg}_{\mathbf{d}}(\mathbf{P})\}.$$

Singularités d'une intégrale

Une version simplifiée d'un calcul de susceptibilité magnétique d'un modèle d'Ising a conduit des physiciens à s'interroger sur la position des singularités de la fonction

$$\phi_n(w) = \frac{1}{\pi} \int_{-1}^1 F_n(w, t) \frac{dt}{\sqrt{1-t^2}} \quad \text{où} \quad F_n(w, t) = \frac{1}{1 - x(w, t)^{n-1} x(w, T_{n-1}(t))},$$

$T_{n-1}(t)$ est un polynôme de Tchebychev de 1ère espèce (c'est un polynôme de degré $n-1$ défini par $\cos((n-1)t) = T_{n-1}(\cos(t))$) et

$$x(w, t) = \frac{2w}{1 - 2wt + \sqrt{(1 - 2wt)^2 - 4w^2}}.$$

Des arguments généraux permettent d'assurer que $\phi_n(w)$ satisfait une équation différentielle linéaire à coefficients polynomiaux, c'est-à-dire de la forme

$$(E) \quad a_k(w)\phi_n^{(k)}(w) + \dots + a_0(w)\phi_n(w) = 0,$$

où les a_i sont des polynômes. La théorie de ces équations affirme alors que les singularités de $\phi_n(w)$ ne peuvent se trouver que parmi les racines du terme de tête $a_k(w)$ de (E).

Le but de ce TP est de calculer une telle équation dans le cas d'intérêt le plus simple, c'est-à-dire lorsque $n = 3$. La taille des calculs est déjà telle qu'il faudra souvent aider Maple dans les étapes intermédiaires. L'approche consiste à calculer un développement en série de $\phi_n(w)$, puis reconstruire (E) par un approximant de Padé-Hermite de $(\phi_n, \phi_n', \dots, \phi_n^{(k)})$.

Malheureusement, la taille des séries (et de leurs coefficients) sont telles que ni `seriestodiffeq`, ni `numapprox[hermite.pade]`, ne sont capables de calculer les approximants de Padé-Hermite dont nous aurons besoin. Pour parvenir au résultat :

- (0) Sur la page du cours (<http://algo.inria.fr/salvy/M1ENS>), récupérer le fichier `gfun.mla` qui contient une version plus récente de `gfun`; changer la variable globale `libname` pour qu'elle commence par le chemin où se trouve ce fichier, et vérifier que cela a fonctionné en testant la valeur de `gfun:-version()`.

1. Premiers essais d'approximation

Tout d'abord, il est possible de se faire une idée approximative de la position des singularités par un simple calcul d'approximant de Padé.

1. Calculer un développement en série de $F_3(w, t)$ à l'ordre 15 par rapport à w ;
2. intégrer terme à terme pour obtenir le développement de $\phi_3(w)$ (on ne demande pas de justifier l'interversion des signes somme et intégrale);
3. calculer un approximant de Padé de cette série et évaluer numériquement la position des pôles de l'approximant.

La suite du TP consiste à trouver un polynôme à coefficients entiers dont ces pôles approchent les racines qui correspondent vraiment aux singularités de ϕ_3 .

2. Développement en série à grande précision

Pour obtenir l'équation (E) que nous cherchons par un calcul d'approximant de Padé-Hermite, il est nécessaire de disposer d'une bonne centaine de termes du développement en série de $\phi_n(w)$ (cette valeur est trouvée en tâtonnant). Les méthodes directes utilisées dans la section précédente ne peuvent pas aller à de très grands ordres. Il faut aider Maple à développer F_3 par rapport à w , puis à intégrer terme à terme.

Pour développer F_n , l'idée est d'utiliser son caractère algébrique, qui entraîne l'existence d'une récurrence linéaire sur ses coefficients (les preuves et algorithmes seront présentés dans un cours ultérieur).

4. Calculer un polynôme $P(w, t, y)$ tel que $P(w, t, F_3(w, t)) = 0$ (?algfuntoalgeq);
5. en déduire une équation différentielle (?algeqtodiffeq), en précisant que la solution qui nous intéresse est celle qui satisfait $y(0) = 1, y'(0) = 0$;
6. en déduire une récurrence sur les coefficients (?diffeqtorec), puis une procédure pour dérouler cette récurrence (?rectoproc, avec comme dans un TP précédent, l'option evalfun à positionner à expand);
7. calculer enfin les 200 premiers coefficients du développement

$$F_3(w, t) = 1 + w^3 + (4t^2 + 4t - 2)w^4 + \dots$$

8. Pour aider Maple à intégrer terme à terme cette série, calculer symboliquement l'intégrale

$$I_k = \frac{1}{\pi} \int_{-1}^1 \frac{t^k}{\sqrt{1-t^2}} dt;$$

transformer la série ci-dessus en un polynôme en t à coefficients des polynômes en w (?collect), puis intégrer terme à terme et retransformer en série en w . On doit trouver les premiers termes

$$\phi_3(w) = 1 + w^3 + 11w^5 + 7w^6 + \dots$$

3. Approximant de Padé-Hermite

9. Calculer l'approximant de Padé-Hermite souhaité (?seriestodiffeq);
10. En déduire une conjecture sur les positions des singularités de ϕ_3 ;
11. Certaines des singularités trouvées à la question précédente ne sont pas vraiment des singularités des solutions; on les appelle des *singularités apparentes*. Pour s'en débarrasser, calculer une équation différentielle d'ordre plus élevé (à l'aide de gfun:-Parameters), et calculer le pgcd des coefficients de tête. Ceci mène à une meilleure conjecture;
12. Pour conforter cette conjecture, calculer les pôles d'un approximant de Padé (40,40) de la série obtenue en question (8) et les afficher dans le plan complexe, avec les singularités de la question ci-dessus.

Du flottant à la forme close : LLL

1. Introduction

L'algorithme LLL, dont le nom provient des initiales de ses inventeurs (A. Lenstra, H. Lenstra, et L Lovász) est un algorithme de réduction des réseaux, historiquement¹ introduit pour factoriser les polynômes à plusieurs variables sur \mathbf{Q} . Il a aussi des applications dans la recherche de dépendance linéaire entre des constantes réelles flottantes.

Le problème auquel on donnera une réponse dans ces notes est celui de trouver un vecteur de norme «presque» minimale dans un réseau donné par une de ces bases. Dans la première section on redonne quelques définitions générales. Dans la seconde on s'attaque à quelques résultats plus spécifiques à notre problème. Dans la troisième et la quatrième section, on donnera un algorithme répondant à nos exigences en temps polynomial. Enfin on appliquera cet algorithme à la question des relations linéaires entre flottants.

2. Rappels et compléments

Définition 1. Soit V un espace euclidien². Un réseau de V est un sous-groupe discret L de V .

Certains auteurs réclament de plus que L engendre V comme \mathbb{R} -espace vectoriel. Dès la propriété ci-dessous prouvée, on se restreindra au cas où $V = \mathbb{R}^n$.

Proposition 1. Soit L un réseau de V . Alors il existe (e_1, \dots, e_k) des vecteurs de V tels que $L = \{\sum_{j=0}^k \lambda_j e_j \mid \lambda_j \in \mathbb{Z}\}$, i.e. tels que L soit un \mathbb{Z} -module libre de base (e_1, \dots, e_k) . De plus, les (e_j) forment une famille \mathbb{R} -libre (donc $k \leq \dim(V)$).

DÉMONSTRATION. On va raisonner par récurrence sur la dimension de V . L'idée est de choisir un vecteur non nul $e_1 \in L$ de norme (presque) minimale, de considérer $V' = V/\mathbb{R}.e_1$, et $\pi : V \rightarrow V'$ la surjection canonique. On montre que $L' = \pi(L)$ est un réseau de l'espace euclidien V' , puis on utilise l'hypothèse de récurrence sur $L' \subset V'$.

L'initialisation à $n = 0$ ne pose aucun problème. De même on peut exclure le cas $L = \{0\}$, car alors $k = 0$ convient.

Par définition de la borne inférieure $\alpha = \inf_{v \in L - \{0\}} \|v\|$, on peut choisir $e_1 \in L - \{0\}$ tel que $\|e_1\| \leq \frac{3}{2}\alpha$. L'espace vectoriel $V' = V/\mathbb{R}.e_1$ est alors naturellement

1. Le lecteur intéressé pourra consulter l'article original dans Mathematische Annalen, à l'adresse

https://openaccess.leidenuniv.nl/dspace/bitstream/1887/3810/1/346_050.pdf

2. Un espace euclidien est un \mathbb{R} -espace vectoriel de dimension finie, muni d'un produit scalaire

muni de la norme quotient : pour $w = \pi(v) \in V'$,

$$\|w\|_{V'} = d(v, \mathbb{R}.e_1) = \inf_{u \in \mathbb{R}.e_1} \|v - u\|_V.$$

Cela en fait un espace euclidien. On considère aussi L' l'image de L par la projection canonique $\pi : V \rightarrow V'$.

Montrons que L' est un réseau de V' . Puisque π est un morphisme de groupes, L' est un groupe. Il suffit donc de vérifier que 0 est isolé dans L' . J'affirme que pour tout $w = \pi(v) \in L' - \{0\}$ on a $\|w\|_{V'} \geq \frac{1}{4}\alpha$. En effet, pour tout point λe_1 de $\mathbb{R}.e_1$, on a, par l'inégalité triangulaire,

$$\begin{aligned} \|v - \lambda.e_1\| &\geq \|v - [\lambda]e_1\| - \|\lambda e_1 - [\lambda]e_1\| \\ &\geq \alpha - \frac{1}{2}\|e_1\| \\ &\geq \frac{1}{4}\alpha, \end{aligned}$$

où $[\lambda]$ désigne l'entier le plus proche de λ , et où on utilise le fait que $v - [\lambda]e_1$ est un vecteur non-nul de L .

On peut donc appliquer l'hypothèse de récurrence à $L' \subset V'$: c'est un \mathbb{Z} -module libre de base (f_2, \dots, f_k) pour un certain k , et les (f_j) sont même linéairement indépendants sur \mathbb{R} . Prenons alors des antécédents (e_2, \dots, e_k) des (f_j) par π . On va finalement montrer que L est un \mathbb{Z} -module libre de base (e_1, \dots, e_k) , ce qui conclura.

Soit $v \in L$. On sait déjà que son image s'écrit $\pi(v) = \sum_{j=2}^k \lambda_j \pi(e_j)$. La différence $v - \sum_{j=2}^k \lambda_j e_j$ est donc un élément de $L \cap \mathbb{R}.e_1$, c'est-à-dire $^3 \mathbb{Z}.e_1$. Ainsi la famille des (e_j) est génératrice de L .

Supposons $\sum_{j=1}^k \lambda_j e_j = 0$, avec $\lambda_j \in \mathbb{R}$. Appliquant π , comme $\pi(e_1) = 0$ et $\pi(e_j) = f_j$, il vient $\sum_{j=2}^k \lambda_j f_j = 0$. Or les (f_j) sont \mathbb{R} -indépendants. D'où $\lambda_j = 0$ pour $j \geq 2$. Il reste alors $\lambda_1 e_1 = 0$, d'où $\lambda_1 = 0$. Ainsi la famille des (e_j) est \mathbb{R} -libre, et on a terminé la récurrence. \square

La proposition ci-dessus indique que pour spécifier un réseau de V il suffit de se donner k vecteurs linéairement indépendants de V : une base du réseau. Il n'y a bien sûr pas unicité de la base, et le but de l'algorithme LLL est justement celui-ci : étant donnée une famille libre (v_1, \dots, v_k) de vecteurs de \mathbb{R}^n , trouver une base «plus simple» 4 du réseau qu'elle engendre.

Commençons par donner une proposition indiquant le lien entre deux bases d'un réseau.

Proposition 2. *Soient (e_1, \dots, e_k) et (f_1, \dots, f_l) deux bases d'un même réseau L de \mathbb{R}^n . Alors $k = l$, et il existe $U \in \mathcal{GL}_k(\mathbb{Z})$ telle que $E = FU$, où $E, F \in \mathcal{GL}_{n,k}(\mathbb{R})$ sont les matrices formées des vecteurs colonne (e_j) , resp. (f_j) .*

DÉMONSTRATION. Par la proposition précédente, $k = \dim(\text{Vect}(e_1, \dots, e_k)) = \dim(\text{Vect}(L)) = \dim(\text{Vect}(f_1, \dots, f_l)) = l$, où $\text{Vect}(P)$ désigne le \mathbb{R} -espace vectoriel engendré par $P \subset \mathbb{R}^n$. 5

3. En tant que sous-groupe discret de $\mathbb{R}.e_1$, il est de la forme $\mathbb{Z}.v$. Comme $\|v\| \geq \alpha \geq \frac{2}{3}e_1$ et $e_1 \in \mathbb{Z}.v$, on a $e_1 = \pm v$.

4. En terme des normes euclidiennes des v_j .

5. En général on voit ce résultat comme corollaire du résultat classique très général suivant : si A est un anneau principal (par exemple \mathbb{Z}), toutes les bases d'un A -module libre ont le même

Pour la seconde affirmation, on écrit les (e_j) sur la base des (f_i) :

$$e_j = \sum_{i=1}^k u_{i,j} f_i,$$

de sorte qu'on a $E = FU$ avec $U \in \mathcal{M}_k(\mathbb{Z})$. Par symétrie entre e et f , on dispose aussi d'une matrice $V \in \mathcal{M}_k(\mathbb{Z})$ telle que $F = EV$.

On a alors $E = FU = EVU$. Or E est de rang k , i.e. la matrice d'une application injective, donc d'un monomorphisme⁶, i.e. on peut simplifier par E dans l'équation précédente : $VU = Id$. De la même façon, $UV = Id$. \square

On déduit de ceci que la définition suivante fait sens.

Définition 2. Soit L un réseau de base (e_j) dans \mathbb{R}^n . Comme dans la proposition ci-dessus, on note $E \in \mathcal{M}_{n,k}(\mathbb{R})$ la matrice de colonnes (e_j) . Le déterminant de L est $d(L) = \sqrt{\det({}^t E.E)}$.

3. Quelques résultats préliminaires

Dans les applications pratiques, les réseaux sont toujours donnés par des bases à coefficients entiers, de sorte qu'ils sont toujours inclus dans $\mathbb{Z}^n \subset \mathbb{R}^n$. Dans la suite on fixe un réseau $L \subset \mathbb{Z}^n$ de base (f_1, \dots, f_k) . Par le procédé d'orthonormalisation de Gram-Schmidt, on dispose d'une «meilleure» \mathbf{Q} -base de $Vect_{\mathbf{Q}}(L)$:

$$f_j^* = f_j - \sum_{i=1}^{j-1} \mu_{i,j} f_i^*, \quad \text{où} \quad \mu_{i,j} = \frac{\langle f_j, f_i^* \rangle}{\langle f_i^*, f_i^* \rangle}.$$

On a l'égalité matricielle $F^{(*)}M = F$, où $F^{(*)}$ a pour colonnes les (f_j^*) , F a pour colonnes les (f_j) , et M est la matrice triangulaire supérieure avec des 1 sur la diagonale donnée par $M_{i,j} = \mu_{i,j}$ si $i < j$, $M_{i,j} = 1$ si $i = j$, et $M_{i,j} = 0$ sinon.

L'idée de l'algorithme LLL est de trouver une base de L presque orthogonale, puis de choisir le plus petit vecteur de cette base. Mais regardons de plus près les propriétés des vecteurs (f_j^*) .

Proposition 3. Soit $(f_j)_{j \leq k}$ une base du réseau $L \subset \mathbb{Z}^n$. Alors $\min_{i \leq k} \|f_i^*\| \leq \|f\|$ pour tout $f \in L - \{0\}$.

DÉMONSTRATION. On écrit $f = \sum_{j=1}^k \lambda_j f_j$ avec $\lambda_j \in \mathbb{Z}$. Soit $l \leq k$ le plus grand indice tel que $\lambda_l \neq 0$. On se rappelle que pour tout $j \leq k$, on a $f_j = \sum_{i=1}^j \mu_{i,j} f_i^*$. En remplaçant dans l'expression de f , il vient :

$$f = \sum_{i=j}^l \left(\lambda_j \sum_{i=1}^j \mu_{i,j} f_i^* \right) = \lambda_l f_l^* + v, \quad \text{où} \quad v \in Vect_{\mathbf{Q}}((f_i)_{i < l}).$$

Le procédé de Gram-Schmidt étant ce qu'il est, on peut appliquer le théorème de Pythagore pour conclure :

$$\|f\|^2 = \|\lambda_l f_l^*\|^2 + \|v\|^2 \geq |\lambda_l|^2 \|f_l^*\|^2 \geq \|f_l^*\|^2 \geq \left(\min_{j \leq k} \|f_j\| \right)^2$$

cardinal. Ici, on a affaire à des groupes abéliens plongés dans un monde avec beaucoup de structure, ce qui simplifie beaucoup.

6. Dans une catégorie, on dit qu'un morphisme $f : A \rightarrow B$ est un monomorphisme si pour tout objet C et tous morphismes $g_1, g_2 \in \mathcal{H}om(B, C)$ on a $g_1 \circ f = g_2 \circ f \Rightarrow g_1 = g_2$.

□

Proposition 4 (Inégalité de Hadamard). *Si $L \subset \mathbb{Z}^n$ a pour base (f_1, \dots, f_k) , alors*

$$d(L) = \sqrt{\det({}^t F.F)} = \prod_{j=1}^k \|f_j^*\| \leq \prod_{j=1}^k \|f_j\|.$$

DÉMONSTRATION. On a $\det({}^t F.F) = \det({}^t M {}^t F^{(*)} F^{(*)} M) = \det(M)^2 \det({}^t F^{(*)} F^{(*)})$. Or $\det(M) = 1$, et ${}^t F^{(*)} F^{(*)} = \text{diag}(\|f_j\|^2)$, car la base (f_j^*) est orthogonale. D'où l'égalité dans la proposition. Pour l'inégalité il suffit de voir que $\|f_j^*\| \leq \|f_j\|$, puisque f_j^* est un projeté orthogonal de f_j . □

4. L'algorithme BasePropre

Définition 3. *Une base (f_j) d'un réseau est dite propre si $|\mu_{i,j}| \leq \frac{1}{2}$ pour tout $i < j$.*

Proposition 5 (Réduction faible). *Pour toute base $(f_j)_{j \leq k}$ d'un réseau L , il existe une base propre $(g_j)_{j \leq k}$ de L telle que $g_j^* = f_j^*$ pour tout $j \leq k$.*

DÉMONSTRATION. Voici l'algorithme. La preuve de la correction se fait par récurrence, et est laissée en exercice au lecteur.

Algorithme 1 Algorithme BasePropre

```

pour j=1..k faire
   $g_j := f_j$ ;
fin pour
pour j=2..k faire
  pour i=(j-1)..1 faire
     $g_j := g_j - \lfloor \langle g_j, g_i^* \rangle / \langle g_i^*, g_i^* \rangle \rfloor g_i$ ;
  fin pour
fin pour

```

Attention à bien faire décroître i dans la seconde boucle : si on ne le fait pas dans le bon sens, l'algorithme ne marche pas. □

5. L'algorithme BaseRéduite

Théorème 8 (LLL). *On peut déterminer $v \in L - \{0\}$ tel que*

$$\|v\| \leq 2^{\frac{k-1}{2}} \min_{e \in L - \{0\}} \|e\|$$

en temps polynomial⁷ en n et en $\log B$, où $B = \max \|f_j\|$.

Pour obtenir un tel vecteur, on construit par l'algorithme qui va suivre une base «réduite» de L , et on se rend compte que le premier vecteur de cette base convient.

Définition 4. *Une base réduite de L est une base propre $(f_i)_{i \leq k}$ telle que pour chaque i , on ait $\|f_i^*\|^2 \leq 2\|f_{i+1}^*\|^2$.*

7. $O(n^{5+\epsilon}(\log B)^{2+\epsilon})$.

Proposition 6. *Si (f_i) est réduite, alors pour tout $f \in L - \{0\}$, on a $\|f_1\| \leq 2^{\frac{k-1}{2}} \|f\|$.*

DÉMONSTRATION. La proposition 3 implique que

$$\|f\| \geq \min_{j \leq k} \|f_j^*\| \geq \min_{j \leq k} 2^{-\frac{j-1}{2}} \|f_1^*\| \geq 2^{-\frac{k-1}{2}} \|f_1\|,$$

car $f_1^* = f_1$. □

Pour répondre à notre problème, il suffit donc de donner un algorithme de réduction de la base. (On admet que la proposition 5 est correcte.)

Algorithme LLL(F)

G ← BasePropre(F).

tant que G n'est pas réduite,

échanger f_i et f_{i+1} pour un rang tel que $|f_i|^2 > 2|f_{i+1}|^2$;

G ← BasePropre(G);

Proposition 7. *LLL termine, et calcule bien une base réduite de L en temps polynomial.*

DÉMONSTRATION. La correction de l'algorithme est évidente. Pour la terminaison, pour une base propre F de L donnée par l'étape d'initialisation, on définit $\mathcal{V}(F) = \prod_{j=1}^k V_j(F)$, où $V_j(F) = d(L(f_1, \dots, f_j)) = \prod_{i=1}^j \|f_i^*\|$.

Alors $\mathcal{V}(F)$ n'est pas modifié lors d'une étape de réduction faible, car il ne dépend que des f_j^* , qui ne sont pas modifiés. Il faut ensuite surveiller ce qui se passe lors d'une étape d'échange. On va montrer qu'une telle étape multiplie $\mathcal{V}(F)^2$ par un facteur $< \frac{4}{3}$. Comme c'est un entier (en tant que produit des déterminants $\det({}^t(f_1, \dots, f_j). (f_1, \dots, f_j))$, entiers), positif, non nul (car les f_j sont libres), l'algorithme finira forcément.

Soit $(f_j)_{j \leq k}$ une base propre non réduite, et $l \leq k$ tel que $\|f_l^*\|^2 > 2\|f_{l+1}\|^2$. On note (g_j) la famille obtenue à partir de (f_j) en permutant les termes en place l et $l+1$. On a clairement $V_j(g) = V_j(f)$ dès que $j \neq l$. Pour V_l il faut travailler un peu plus : f_l^* est remplacé par

$$g_l^* = g_l - \sum_{i=1}^{l-1} \frac{\langle g_l, g_i^* \rangle}{\langle g_i^*, g_i^* \rangle} g_i^* = f_{l+1} - \sum_{i=1}^{l-1} \frac{\langle f_{l+1}, f_i^* \rangle}{\langle f_i^*, f_i^* \rangle} f_i^* = f_{l+1}^* + \mu_{l,l+1} f_l^*.$$

Ainsi, $\|g_l^*\|^2 = \|f_{l+1}^*\|^2 + \mu_{l,l+1}^2 \|f_l^*\|^2 \leq ((1/2) + (1/2)^2) \|f_l^*\|^2 = \frac{3}{4} \|f_l^*\|^2$. Finalement, on obtient $\frac{V_l(g)}{V_l(f)} = \frac{\|f_l^*\|}{\|g_l^*\|} \leq \frac{\sqrt{3}}{2}$. D'où

$$\mathcal{V}(g)^2 \leq \frac{3}{4} \mathcal{V}(f)^2.$$

Ceci conclut quand à la terminaison. Il ne manque pas grand chose pour obtenir la complexité : simplement de remarquer qu'initialement,

$$\mathcal{V}(F) = \|f_1^*\|^k \dots \|f_k^*\|^1 \leq \|f_1\|^k \dots \|f_k\|^1 \leq M^{\frac{k(k+1)}{2}},$$

où $M = \max \|f_j\|$. Ainsi, le nombre d'itérations est borné par

$$\log_{4/3}(M^{\frac{k(k+1)}{2}}) = O(\log(M)n^2).$$

Comme l'algorithme BasePropre est lui-même polynomial en $\log(M)$ et n , on obtient finalement bien que l'algorithme LLL est polynomial. □

6. Application : relations linéaires entre flottants

Soient x_1, \dots, x_n des constantes réelles (des flottants). On cherche à déterminer des relations linéaires entre celles-ci. Pour cela, on applique l'algorithme LLL à la matrice

$$A_N = \begin{pmatrix} 1 & 0 & \cdots & 0 \\ 0 & 1 & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ 0 & \cdots & 0 & 1 \\ \hline [10^N x_1] & \cdots & [10^N x_{n-1}] & [10^N x_n] \end{pmatrix}.$$

On obtient une combinaison linéaire $v = \sum \lambda_j e_j$ des colonnes e_j , qui est «courte» au sens du Théorème. Qu'elle soit courte implique notamment que sa dernière composante l'est, ce qui sous-entend qu'on a pu trouver une relation linéaire entre les x_j . Plus explicitement, si les x_j vérifient une relation linéaire $\sum_j \lambda_j x_j$ avec $\lambda_j \in \mathbb{Z}$, alors le vecteur colonne

$$u = \begin{pmatrix} \lambda_1 \\ \vdots \\ \lambda_n \\ \sum_j \lambda_j [10^N x_j] \end{pmatrix}$$

est dans le réseau L associé à la matrice ci-dessus. À cause des problèmes d'arrondi, on a $\|u\|^2 \leq 4\Lambda^2$ en posant $\Lambda = \sum \lambda_j$. D'après le théorème, l'algorithme LLL renvoie donc un vecteur v tel que $\|v\|^2 \leq 2^{n+1}\Lambda^2$.

La remarque clé, ici, est que cette quantité ne dépend pas de N . Notons la K^2 , de sorte que LLL renvoie v tel que $\|v\| \leq K$. Un tel $v = \sum \alpha_j e_j$, où $\alpha_j \in \mathbb{Z}$ et e_j est la j -ème colonne de la matrice A_N , a pour coordonnées

$$v = \begin{pmatrix} \alpha_1 \\ \vdots \\ \alpha_n \\ \sum_j \alpha_j [10^N x_j] \end{pmatrix}.$$

Comme $\|v\| \leq K$, on a $\alpha_i \leq K$ pour chaque i , i.e. on ne peut obtenir sur la dernière coordonnée qu'un nombre fini de combinaisons linéaires des $[10^N x_j]$ quand N varie. Lorsqu'on fait tendre N vers l'infini, on distingue entre les vraies relations et les fausses relations sur les x_j . En effet, si $\sum \alpha_j x_j$ est non nul, alors pour N assez grand, $\sum \alpha_j [10^N x_j]$ est toujours plus grand que K .

Finalement, on a prouvé que si les x_j vérifient une relation linéaire, alors l'algorithme LLL appliqué à la matrice A_N avec N assez grand trouve une relation linéaire (pas forcément la même). Il serait intéressant de connaître la stabilité de cet algorithme.

Les décimales de π en base 16

La fascination pour les décimales de π a mené à de nombreuses expérimentations. En 1995, Simon Plouffe fait ainsi la découverte d'une formule étonnante :

$$(E) \quad \pi = \sum_{i=0}^{\infty} \frac{1}{16^i} \left(\frac{4}{8i+1} - \frac{2}{8i+4} - \frac{1}{8i+5} - \frac{1}{8i+6} \right).$$

Cette formule permet de calculer le n ième chiffre en base 16 de π (et même quelques chiffres à partir du n ième) sans calculer les chiffres précédents, avec peu de mémoire. Il est donc utile de comprendre comment découvrir des formules de ce type. Dans l'article écrit par Simon Plouffe avec David Bailey et Peter Borwein où (E) est présentée et prouvée, sa découverte est décrite comme le résultat de « divination inspirée et recherche intensive ». Le but du TP est de montrer comment une telle recherche peut être menée, et indiquer comment cette identité et des identités similaires peuvent aussi être prouvées automatiquement. S'il reste du temps, l'utilisation de cette formule pour le calcul de décimales lointaines pourra aussi être abordée.

1. Découvertes automatiques

Le principe est simple, il s'agit de calculer numériquement un certain nombre de constantes et d'utiliser LLL pour trouver une relation linéaire entre elles.

1. Calculer avec cent décimales de précision les constantes

$$\Sigma_j := \sum_{i=0}^{\infty} \frac{16^{-i}}{8i+j}, \quad j = 1, \dots, 8.$$

2. Utiliser l'algorithme LLL pour « découvrir » la relation (E).
(?IntegerRelations, LinearDependency)
3. Écrire une procédure prenant en argument une constante, une liste d'expressions, une précision, et renvoyant l'identité que suggère LLL utilisé sur cette constante et ces expressions évaluées à cette précision.
4. Utiliser cette procédure pour conjecturer des identités pour $\ln 2$, $\ln 3$, $\ln 5$, $\arctan 2$, $\arctan 3$, $\sqrt{2} \arctan(1/\sqrt{2})$, $\sqrt{2} \ln(1 + \sqrt{2})$.

D'autres identités peuvent être conjecturées avec des jeux de constantes différents.

5. Obtenir des conjectures pour les expressions de π^2 , $\ln 7$, $\ln^2 2$ en fonction des séries

$$\sum_{i=1}^{\infty} \frac{2^{-ji}}{i^m}, \quad j = 1, \dots, 6, \quad m = 1, \dots, 5.$$

2. Preuves

La partie difficile du travail se situe dans la découverte : trouver la bonne classe de constantes dans laquelle l'identité a des chances d'exister. La phase de preuve est plus facile et plusieurs preuves existent. La méthode utilisée ci-dessous suggère aussi les constantes qui ont des chances d'être obtenues.

6. Calculer une forme close pour les sommes

$$S_j(z) := \sum_{i=0}^{\infty} \frac{z^{8i+j}}{8i+j}, \quad j = 1, \dots, 8.$$

Pour aider Maple dans cette sommation, il pourra être utile de calculer d'abord les sommes des dérivées, puis d'intégrer.

7. En déduire des expressions symboliques pour les Σ_j , puis une preuve de (E). Les autres sommes trouvées en question (5) se prouvent de la même manière.

3. Calcul rapide de décimales

Le calcul se déroule de la même manière pour tous ces nombres se décomposant en combinaison linéaire à coefficients entiers de séries du type

$$S = \sum_{i=0}^{\infty} \frac{p(i) \cdot b^{-i}}{q(i)},$$

où p et q sont des polynômes à coefficients entiers. Les chiffres en base b de S à partir du N ième sont donnés par la partie fractionnaire de $b^N S$, que nous noterons $b^N S \bmod 1$. La somme se décompose pour donner

$$b^N S \bmod 1 = \sum_{i=0}^N \frac{p(i) \cdot b^{N-i} \bmod q(i)}{q(i)} + \sum_{i>N} \frac{p(i)}{b^{i-N} q(i)} \bmod 1.$$

La seconde somme converge géométriquement et peu de termes sont nécessaires pour obtenir des décimales. La première s'évalue rapidement en calculant chacun des sommands par exponentiation binaire.

8. Écrire une procédure prenant deux entiers m et k et calculant $16^m \bmod k$ par exponentiation binaire ;
 9. Écrire deux procédures prenant deux entiers N et q , un polynôme p et sa variable k et renvoyant les premières décimales de

$$\sum_{i=0}^N \frac{q \cdot 16^{N-i} \bmod p(i)}{p(i)} \quad \text{et de} \quad \sum_{i>N} \frac{q}{16^{i-N} p(i)}.$$

10. Écrire enfin une procédure prenant en entrée des entiers N, b_1, \dots, b_8 et renvoyant les premières décimales en base 16 à partir de la N ième de

$$\sum_{i \geq 0} \frac{1}{16^i} \sum_{j=1}^8 \frac{b_j}{8i+j}.$$

Tester cette procédure sur π .

11. Estimer la complexité de ce calcul en fonction de N .

Troisième partie

Preuves automatiques

Identités de fonctions spéciales et séries D-finies

Résumé

Les équations différentielles linéaires et les récurrences linéaires fournissent des structures de données permettant de calculer avec des fonctions ou des suites, et en particulier de prouver des identités sur ces objets.

1. Définitions

1.1. Rappels et complément sur les séries formelles. Si \mathbb{K} désigne un corps, l'anneau des séries formelles à coefficients dans \mathbb{K} est noté $\mathbb{K}[[X]]$. Ses principales propriétés ont été présentées au Cours 3. Son corps des fractions, noté $\mathbb{K}((X))$ est égal à $\mathbb{K}[[X]][1/X]$. Ses éléments sont appelés des séries de Laurent formelles. C'est une algèbre non seulement sur \mathbb{K} , mais aussi sur le corps des fractions rationnelles $\mathbb{K}(X)$.

Dans tout ce cours on suppose \mathbb{K} de caractéristique nulle. On peut donc penser sans rien perdre aux idées à $\mathbb{K} = \mathbb{Q}$.

1.2. Séries D-finies.

Définition 1. Une série formelle $A(X)$ à coefficients dans un corps \mathbb{K} est dite différentiellement finie (ou D-finie) lorsque ses dérivées successives A, A', \dots , engendrent un espace vectoriel de dimension finie sur le corps $\mathbb{K}(X)$ des fractions rationnelles.

De manière équivalente, cette série est solution d'une équation différentielle linéaire à coefficients dans $\mathbb{K}(X)$: si c'est le cas alors l'équation différentielle permet de récrire toute dérivée d'ordre supérieur à celui de l'équation en termes des dérivées d'ordre moindre (en nombre borné par l'ordre), à l'inverse, si l'espace est de dimension finie, alors pour m suffisamment grand, $A, A', \dots, A^{(m)}$ sont liées et une relation de liaison entre ces dérivées est une équation différentielle linéaire.

En pratique, l'équation différentielle est utilisée pour les calculs, et la caractérisation de la définition pour les preuves d'existence.

1.3. Suites P-récurrentes.

Définition 2. Une suite $(a_n)_{n \geq 0}$ d'éléments d'un corps \mathbb{K} est appelée suite polynomialement récurrente (ou P-récurrente) si elle satisfait une récurrence de la forme

$$(1) \quad p_d(n)a_{n+d} + p_{d-1}(n)a_{n+d-1} + \dots + p_0(n)a_n = 0, \quad n \geq 0,$$

où les p_i sont des polynômes de $\mathbb{K}[X]$.

De la même manière, on peut parler de l'espace vectoriel de dimension finie engendré par la suite et ses décalées.

2. Équivalence entre séries D-finies et suites P-récurrentes

Théorème 9. *Une série formelle est D-finie si et seulement si la suite de ses coefficients est P-récurrente.*

DÉMONSTRATION. Soit $A(X) = a_0 + a_1X + \dots$ une série D-finie et

$$(2) \quad q_0(X)A^{(m)}(X) + \dots + q_m(X)A(X) = 0$$

une équation différentielle qui l'annule. En notant $[X^n]f(X)$ le coefficient de X^n dans la série $f(X)$ avec la convention que ce coefficient est nul pour $n < 0$, on a pour $n \geq 0$

$$(3) \quad [X^n]f'(X) = (n+1)[X^{n+1}]f(X), \quad [X^n]X^k f(X) = [X^{n-k}]f(X).$$

Par conséquent, l'extraction du coefficient de X^n de (2) fournit une récurrence linéaire sur les a_n valide pour tout $n \geq 0$ avec la convention $a_k = 0$ pour $k < 0$. Pour obtenir une récurrence de la forme (1) il faut décaler les indices de $n_0 := \max_{0 \leq i \leq m} (\deg q_i + i - m)$ s'il est strictement positif. Les équations obtenues alors pour les indices moindres fournissent des contraintes linéaires sur les premiers coefficients a_n pour qu'ils correspondent aux coefficients d'une série solution de (2).

À l'inverse, soit (a_n) une suite vérifiant la récurrence (1). Les identités analogues à (3) sont maintenant

$$\sum_{n \geq 0} n^k a_n X^n = \left(X \frac{d}{dX} \right)^k A(X), \quad \sum_{n \geq 0} a_{n+k} X^n = (A(X) - a_0 - \dots - a_{k-1} X^{k-1}) / X^k,$$

où A est la série génératrice des coefficients a_n et la notation $(Xd/dX)^k$ signifie que l'opérateur Xd/dX est appliqué k fois. En multipliant (1) par X^n et en sommant pour n allant de 0 à ∞ , puis en multipliant par une puissance de X on obtient donc une équation différentielle linéaire de la forme

$$q_0(X)A^{(d)}(X) + \dots + q_d(X)A(X) = p(X),$$

où le membre droit provient des conditions initiales. Il est alors possible, quitte à augmenter l'ordre de l'équation de 1, de faire disparaître ce membre droit, par une dérivation et une combinaison linéaire. \square

Ce calcul permet aussi d'observer le résultat suivant.

Lemme 1. *Si $A(X)$ est une série D-finie solution de (2) et $q_0(0) \neq 0$, alors le coefficient de tête de la récurrence (1) satisfaite par ses coefficients est le polynôme $q_0(0)(n+1) \cdots (n+m)$.*

DÉMONSTRATION. D'après (3), un terme $cX^i A^{(j)}(X)$ intervient dans la récurrence sur les coefficients sous la forme $c(n-i+1) \cdots (n-i+j)a_{n-i+j}$. L'indice maximal est donc atteint pour $j-i$ maximal et donc pour $j=m$ si $i=0$. \square

Exemple 1. L'équation $y' - x^k y = 0$ ($k \in \mathbb{N}$) donne la récurrence $(n+1)a_{n+1} - a_{n-k} = 0$ valide pour tout $n \geq 0$ avec la convention que les a_n d'indice négatif sont nuls. On en déduit que a_0 est libre, puis les contraintes $a_1 = \dots = a_k = 0$, et les coefficients suivants sont fournis par la récurrence décalée $(n+k+1)a_{n+k+1} - a_n = 0$, valide pour $n \geq 0$. On reconnaît ainsi les coefficients de $a_0 \exp(x^{k+1}/(k+1))$.

3. Test d'égalité

Lemme 2. *Si (u_n) et (v_n) sont deux suites solutions de (1), $u_0 = v_0, \dots, u_{d-1} = v_{d-1}$ et $0 \notin p_d(\mathbb{N})$, alors ces suites sont égales.*

DÉMONSTRATION. Par récurrence, puisque $0 \notin p_d(\mathbb{N})$ permet d'inverser le coefficient de tête de la récurrence et donc d'exprimer chaque terme à partir de l'indice d en fonction des précédents. \square

Corollaire 1. *Si $f(X)$ et $g(X)$ sont deux séries formelles solutions de (2), $f(0) = g(0), \dots, f^{(m-1)}(0) = g^{(m-1)}(0)$ et $q_0(0) \neq 0$, alors ces séries sont égales.*

DÉMONSTRATION. D'après le Lemme 1, le coefficient de tête de la récurrence sur les coefficients des solutions séries formelles de (2) ne s'annule pas sur \mathbb{N} . Les contraintes linéaires sur les conditions initiales jusqu'à l'indice n_0 introduit dans la preuve du Théorème 9 définissent les coefficients d'indice m à $m + n_0$ à partir des précédents et le Lemme 2 s'applique pour les valeurs suivantes. \square

4. Somme et Produit

Théorème 10. *L'ensemble des séries D-finies à coefficients dans un corps \mathbb{K} est une algèbre sur \mathbb{K} . L'ensemble des suites P-récurrentes d'éléments de \mathbb{K} est aussi une algèbre sur \mathbb{K} .*

DÉMONSTRATION. Les preuves pour les suites et les séries sont similaires. Les preuves pour les sommes sont plus faciles que pour les produits, mais dans le même esprit. Nous ne donnons donc que la preuve pour le produit $h = fg$ de deux séries D-finies f et g . Par la formule de Leibniz, toutes les dérivées de h s'écrivent comme combinaisons linéaires de produits entre une dérivée $f^{(i)}$ de f et une dérivée $g^{(j)}$ de g . Les dérivées de f et de g étant engendrées par un nombre fini d'entre elles, il en va de même pour les produits $f^{(i)}g^{(j)}$, ce qui prouve la D-finitude de h . \square

En outre, cette preuve permet de borner l'ordre des équations : l'ordre de l'équation satisfaite par une somme est borné par la somme des ordres des équations satisfaites par les sommants, et l'ordre de l'équation satisfaite par un produit est borné par le produit des ordres.

Cette preuve donne également un algorithme pour trouver l'équation différentielle (resp. la récurrence) cherchée : il suffit de calculer les dérivées (resp. les décalées) successives en les récrivant sur un ensemble fini de générateurs. Une fois leur nombre suffisant (c'est-à-dire au pire égal à la dimension plus 1), il existe une relation linéaire entre elles. À partir de la matrice dont les lignes contiennent les coordonnées des dérivées successives (resp. des décalés successifs) sur cet ensemble fini de générateurs, la détermination de cette relation se réduit alors à celle du noyau de la transposée.

Exemple 2. Voici comment prouver (et même découvrir) l'identité

$$\arcsin(x)^2 = \sum_{k \geq 0} \frac{k!}{\left(\frac{1}{2}\right) \cdots \left(k + \frac{1}{2}\right)} \frac{x^{2k+2}}{2k+2}.$$

Le calcul consiste à partir d'une équation satisfaite par $\arcsin(x)$, en déduire une équation satisfaite par son carré, traduire cette équation en récurrence sur les coefficients, et conclure en constatant que cette récurrence est satisfaite par les coefficients de la série.

Le point de départ est la propriété $(\arcsin(x))' = 1/\sqrt{1-x^2}$, qui permet de représenter \arcsin par l'équation différentielle $(1-x^2)y'' - xy' = 0$ avec les conditions initiales $y(0) = 0$, $y'(0) = 1$.

Ensuite, en posant $h = y^2$, les dérivations et réductions successives donnent

$$\begin{aligned} h' &= 2yy', \\ h'' &= 2y'^2 + 2yy'' = 2y'^2 + \frac{2x}{1-x^2}yy', \\ h''' &= 4y'y'' + \frac{2x}{1-x^2}(y'^2 + yy'') + \left(\frac{2}{1-x^2} + \frac{4x^2}{(1-x^2)^2} \right) yy', \\ &= \left(\frac{4x}{1-x^2} + \frac{2x^2}{(1-x^2)^2} + \frac{2}{1-x^2} + \frac{4x^2}{(1-x^2)^2} \right) yy' + \frac{2x}{1-x^2}y'^2. \end{aligned}$$

Les quatre vecteurs h, h', h'', h''' sont combinaisons linéaires des trois vecteurs y^2, yy', y'^2 . Ils sont donc liés et une relation de liaison s'obtient en calculant le noyau de la matrice 3×4 qui découle de ce système. Le résultat est

$$(1-x^2)h''' - 3xh'' - h' = 0.$$

La récurrence qui s'en déduit est

$$(n+1)(n+2)(n+3)a_{n+3} - (n+1)^3a_{n+1} = 0.$$

Comme le facteur $(n+1)$ ne s'annule pas sur \mathbb{N} , il est possible de simplifier pour obtenir la récurrence équivalente

$$(n+2)(n+3)a_{n+3} - (n+1)^2a_{n+1} = 0.$$

La vérification que les coefficients de la série ci-dessus vérifient cette identité est facile.

5. Séries algébriques

Théorème 11. *Si la série $Y(X)$ annule un polynôme $P(X, Y)$ de degré d en Y , alors elle est solution d'une équation différentielle linéaire d'ordre au plus d .*

DÉMONSTRATION. La preuve est algorithmique. Quitte à diviser d'abord P par son pgcd avec sa dérivée P_Y par rapport à Y , il est possible de le supposer premier avec P_Y (car la caractéristique est nulle!). En dérivant $P(X, Y) = 0$ et en isolant Y' , il vient

$$Y' = -\frac{P_X}{P_Y}.$$

Par *inversion modulaire* de P_Y (voir le Cours 5), cette identité se réécrit via un calcul de pgcd étendu en

$$Y' = R_1(Y) \bmod P,$$

où R_1 est un polynôme en Y de degré au plus d et à coefficients dans $\mathbb{K}(X)$. Ceci signifie que Y' s'écrit comme combinaison linéaire de $1, Y, Y^2, \dots, Y^{d-1}$ à coefficients dans $\mathbb{K}(X)$. Dériver à nouveau cette équation, puis réécrire Y' et prendre le reste de la division par P mène à nouveau à une telle combinaison linéaire pour Y'' et plus généralement pour les dérivées successives de Y . Les $d+1$ vecteurs $Y, Y', \dots, Y^{(d)}$ sont donc linéairement dépendants et la relation de liaison est l'équation cherchée. \square

Les mêmes arguments que ci-dessus mènent à une autre propriété de clôture des séries D-finies.

Corollaire 2. *Si f est une série D-finie et y une série algébrique sans terme constant, alors $f \circ y$ est D-finie.*

La preuve consiste à observer que les dérivées successives de $f \circ y$ s'expriment comme combinaisons linéaires des $f^{(i)}(y)y^j$ pour un nombre fini de dérivées de f (par D-finitude) et de puissances de y (par la même preuve que pour le théorème 11). Cette preuve fournit encore un algorithme.

La moyenne arithmético-géométrique et les séries hypergéométriques

1. La moyenne arithmético-géométrique

Si a et b sont deux réels tels que $0 \leq b \leq a$, les deux suites définies par

$$a_{n+1} = \frac{a_n + b_n}{2}, \quad b_{n+1} = \sqrt{a_n b_n}, \quad a_0 = a, \quad b_0 = b$$

ont une limite commune dont l'existence se déduit de $b_n \leq b_{n+1} \leq a_{n+1} \leq a_n$ et

$$a_{n+1}^2 - b_{n+1}^2 = \left(\frac{a_n - b_n}{2} \right)^2.$$

Cette limite est notée $M(a, b)$. Cette fonction est clairement homogène : pour $\lambda > 0$, $M(\lambda a, \lambda b) = \lambda M(a, b)$ ce qui permet de concentrer l'étude sur la fonction de une variable $M(1, x)$, dont on admet qu'elle est analytique au voisinage de $x = 1$.

1. Dédire de l'homogénéité et de $M(a_1, b_1) = M(a_0, b_0)$ avec $a_0 = 1 + x$ et $b_0 = 1 - x$ une équation fonctionnelle satisfaite par $M(1, \cdot)$;
2. Utiliser cette équation fonctionnelle pour calculer les 10 premiers coefficients de Taylor de la fonction $A(x) = 1/M(1, \sqrt{1-x})$ (ou $M(1, x) = 1/A(1-x^2)$) à l'origine ;
3. Utiliser ces coefficients pour conjecturer une équation différentielle linéaire satisfaite par $A(x)$;
4. En utilisant la clôture des solutions d'équations différentielles linéaires par substitution algébrique, prouver que la série solution de cette équation différentielle avec les conditions initiales $y(0) = 1$, $y'(0) = 1/4$ satisfait l'équation fonctionnelle satisfaite par A ;
5. La série hypergéométrique est définie comme

$$F(a, b; c; z) := \sum_{n \geq 0} \frac{(a)_n (b)_n}{(c)_n} \frac{z^n}{n!}, \quad \text{où } (x)_n = x(x+1) \cdots (x+n-1).$$

Calculer une récurrence satisfaite par les coefficients de Taylor de $A(x)$ et en déduire l'identité (due à Gauss)

$$M(a, b) = \frac{a}{F\left(\frac{1}{2}, \frac{1}{2}; 1; 1 - \frac{b^2}{a^2}\right)}.$$

2. La relation de Legendre sur les intégrales elliptiques

Les intégrales elliptiques complètes de première et de seconde espèce sont

$$K(k) := \int_0^1 \frac{dt}{\sqrt{(1-t^2)(1-k^2t^2)}}, \quad E(k) := \int_0^1 \frac{\sqrt{1-k^2t^2}}{\sqrt{1-t^2}} dt.$$

6. Calculer une récurrence linéaire satisfaite par les intégrales

$$B_n = \int_0^1 \frac{t^{2n} dt}{\sqrt{1-t^2}};$$

7. En calculant une récurrence sur les coefficients du développement des intégrandes par rapport à k puis sur ceux de l'intégrale terme à terme, en déduire que pour $|k| < 1$,

$$K(k) = \frac{\pi}{2} F\left(\frac{1}{2}, \frac{1}{2}; 1; k^2\right), \quad E(k) = \frac{\pi}{2} F\left(-\frac{1}{2}, \frac{1}{2}; 1; k^2\right).$$

En conclure, après Gauss, que $K(k)$ peut se calculer par une moyenne arithmético-géométrique (π étant donné).

Les questions qui suivent permettent de prouver la *relation de Legendre*, qui s'écrit

$$E(k)K(k') + E(k')K(k) - K(k)K(k') = \frac{\pi}{2}, \quad \text{avec } k' = \sqrt{1-k^2}, \quad 0 \leq k \leq 1.$$

8. En considérant les récurrences satisfaites par leurs coefficients montrer que $F(x) := \frac{2}{\pi} K(\sqrt{x})$ et $G(x) := \frac{2}{\pi} E(\sqrt{x})$ sont liés par

$$G(x) = (1-x)(2xF'(x) + F(x));$$

9. Injecter cette valeur de G dans le membre gauche de la relation de Legendre pour constater que celle-ci se réécrit

$$x(1-x) \begin{vmatrix} F(x) & F(1-x) \\ F'(x) & F'(1-x) \end{vmatrix} = \text{cte};$$

10. Observer que $F(x)$ et $F(1-x)$ sont toutes deux solutions d'une même équation différentielle linéaire d'ordre 2 et en déduire l'existence de cette constante ;

11. Pour obtenir le membre droit de la relation de Legendre en faisant tendre k vers 0, on admettra que

$$K(k') = -\ln(k) + 2\ln 2 + O(k^2 \ln k), \quad k \rightarrow 0.$$

12. Déduire de la relation de Legendre une relation entre $K(1/\sqrt{2})$, $E(1/\sqrt{2})$ et $\pi/2$.

3. L'algorithme de Brent-Salamin pour le calcul de π

En introduisant la suite $T_n = 2^n a_n (E(b_n/a_n) - K(b_n/a_n))$ et en établissant que $T_{n+1} - T_n = 2^n c_n^2 K(b/a)$, on obtient l'identité suivante que l'on admettra :

$$(4) \quad E(k) = \left(1 - \sum_{n=1}^{\infty} 2^n c_k^2\right) K(k),$$

où $c_k := \sqrt{a_k^2 - b_k^2} = c_{k-1}^2 / (4a_k)$.

En mettant ensemble les équations des questions (7), (12) et celle ci-dessus, on obtient finalement

$$\frac{\pi^2}{4M^2(1, 1/\sqrt{2})} \left(1 - 2 \sum_{n=1}^{\infty} 2^k c_k^2 \right) = \frac{\pi}{2},$$

base de l'algorithme de Brent-Salamin pour le calcul de π . La moyenne arithmético-géométrique convergeant quadratiquement (grosso modo, le nombre de décimales correctes double à chaque étape), il s'agit du meilleur algorithme connu du point de vue de la complexité. L'itération est donnée par :

$$\pi = \lim_{n \rightarrow \infty} \pi_n, \quad \pi_n := \frac{2a_{n+1}^2}{1 - \sum_{k=0}^n 2^k c_k^2},$$

où les a_k et b_k sont les suites de la moyenne arithmético-géométrique pour $a_0 = 1$ et $b_0 = 1/\sqrt{2}$. Pour obtenir une bonne complexité, il faut disposer d'une multiplication rapide et calculer inverse et racine carrée par itération de Newton.

13. Calculer π_n pour $n = 1, \dots, 7$ avec 60 décimales de précision ;
14. Pour bénéficier d'arithmétique rapide en Maple, il vaut mieux effectuer les calculs sur des entiers que sur des flottants. Il suffit pour cela de démarrer le calcul avec $a_0 = 10^D$, $b_0 = \sqrt{10^{2D}}/2$ où D est de l'ordre du nombre de décimales cherchées et d'utiliser des entiers tout au long du calcul (via `iquo` et `isqrt`). Écrire cette itération et la tester avec $D = 10^4$, $D = 2 \cdot 10^4$, $D = 4 \cdot 10^4$, et observer l'évolution du temps de calcul avec D .

Sommation hypergéométrique

Résumé

Les suites hypergéométriques sont très courantes dans les applications. Leur algorithmique fait partie des succès du calcul formel. Les deux problèmes principaux sont la sommation indéfinie et la sommation définie. Les algorithmes correspondants sont dus à Gosper et Zeilberger.

Dans tout ce qui suit, \mathbb{K} est un corps de caractéristique nulle.

1. Sommation indéfinie

1.1. Problème de la sommation indéfinie. La sommation indéfinie est l'analogie discret du calcul de primitives.

Définition 1. *Étant donnée une suite $(f_n) \in \mathbb{K}^{\mathbb{N}}$, on appelle $(F_n) \in \mathbb{K}^{\mathbb{N}}$ une somme indéfinie de (f_n) si*

$$\forall n \in \mathbb{N}, \quad F_{n+1} - F_n = f_n.$$

Le lien entre sommation indéfinie et somme est le même qu'entre primitive et intégrale : si $m \leq p \in \mathbb{N}$,

$$\sum_{n=m}^p f_n = \sum_{n=m}^p (F_{n+1} - F_n) = F_{p+1} - F_m.$$

1.2. Sommation indéfinie hypergéométrique.

Définition 2. *Une suite $(u_n) \in \mathbb{K}^{\mathbb{N}}$ est dite hypergéométrique sur \mathbb{K} s'il existe une fraction rationnelle $r(n) = p(n)/q(n) \in \mathbb{K}(n)$ telle que*

$$\forall n \in \mathbb{N}, \quad q(n)u_{n+1} = p(n)u_n.$$

Exemple 1. – Une suite géométrique est hypergéométrique : prendre $r(n) = \alpha$ où α est la raison de la suite. En ce sens, les suites hypergéométriques sont une généralisation des suites géométriques.

– La suite factorielle est hypergéométrique :

$$\forall n \in \mathbb{N}, \quad \frac{(n+1)!}{n!} = n+1.$$

– $k \in \mathbb{N}$ étant fixé, les coefficients binomiaux $\binom{n}{k}$ forment une suite hypergéométrique :

$$\forall n \in \mathbb{N}, \quad \frac{\binom{n+1}{k}}{\binom{n}{k}} = \frac{\frac{(n+1)!}{k!(n+1-k)!}}{\frac{n!}{k!(n-k)!}} = \frac{n+1}{n+1-k}.$$

- $n \in \mathbb{N}$ étant fixé, les coefficients binomiaux $\binom{n}{k}$ forment une suite hypergéométrique :

$$\forall k \in \mathbb{N}, \quad \frac{\binom{n}{k+1}}{\binom{n}{k}} = \frac{\frac{n!}{(k+1)!(n-k-1)!}}{\frac{n!}{k!(n-k)!}} = \frac{n-k}{k+1}.$$

- Le cas général de suites hypergéométriques sur \mathbb{C} s'écrit

$$u_n = CA^n \frac{\prod_{i=1}^p (a_i)(a_i+1) \cdots (a_i+n)}{\prod_{j=1}^q (b_j)(b_j+1) \cdots (b_j+n)},$$

avec $C, A, a_1, \dots, a_p, b_1, \dots, b_q$ dans \mathbb{C} . Il est clair que cette suite est hypergéométrique, et réciproquement, toute fraction rationnelle peut s'écrire $AP(n)/Q(n)$ où p et q sont unitaires et on obtient la formule ci-dessus en nommant a_1, \dots, a_p les racines de P et b_1, \dots, b_q celles de Q .

Problème 1. On se restreint aux suites hypergéométriques.

- **Entrée** : $r(n) \in \mathbb{K}(n)$, et sous-entendue $(u_n) \in \mathbb{K}^{\mathbb{N}}$ telle que $\forall n \in \mathbb{N}$, $u_{n+1}/u_n = r(n)$.
- **Sortie** : une suite hypergéométrique (v_n) telle que $\forall n \in \mathbb{N}$, $v_{n+1} - v_n = u_n$, ou FAIL si une telle suite n'existe pas. Lorsqu'elle existe, on dit que (v_n) est une somme hypergéométrique de (u_n) .

1.3. Algorithme de Gosper.

Lemme 1. Soit $(u_n) \in \mathbb{K}^{\mathbb{N}}$ une suite hypergéométrique. Si (u_n) admet une somme hypergéométrique (v_n) , alors il existe une fraction rationnelle $t(n) \in \mathbb{K}(n)$ telle que $\forall n \in \mathbb{N}$, $v_n = t(n)u_n$.

DÉMONSTRATION. Soit $s(n) \in \mathbb{K}(n)$ telle que $v_{n+1}/v_n = s(n)$. Alors :

$$\forall n \in \mathbb{N}, \quad v_{n+1} - v_n = (s(n) - 1)v_n = u_n$$

D'où le résultat en prenant $t(n) = 1/(s(n) - 1)$. □

Remarque 1. On obtient même l'équation que doit vérifier $t(n)$:

$$\forall n \in \mathbb{N}, \quad t(n+1)r(n)u_n - t(n)u_n = u_n$$

d'où

$$t(n+1)r(n) - t(n) = 1$$

(équation d'inconnue $t(n)$ rationnelle).

Idée 1 (Gosper, 1978). On se débarrasse des différences entières entre racines et pôles en écrivant $r(n)$ sous la forme (appelée forme de Gosper)

$$(1) \quad r(n) = \frac{a(n)}{b(n)} \frac{c(n+1)}{c(n)}, \quad a, b, c \in \mathbb{K}[n]$$

avec $\forall k \in \mathbb{N}$, $\text{pgcd}(a(n), b(n+k)) = 1$, puis on cherche $t(n)$ sous la forme $t(n) = b(n-1)x(n)/c(n)$.

L'équation sur $t(n)$ devient

$$\frac{b(n)}{c(n+1)}x(n+1) - \frac{a(n)}{b(n)} \frac{c(n+1)}{c(n)} - \frac{b(n-1)}{c(n)}x(n) = 1$$

soit finalement

$$(2) \quad a(n)x(n+1) - b(n-1)x(n) = c(n).$$

Théorème 12 (Gosper 1978). *Avec les notations précédentes, si $x(n)$ est une fraction rationnelle solution de (2), alors c'est un polynôme.*

DÉMONSTRATION. On écrit $x(n) = p(n)/q(n)$ avec $p, q \in \mathbb{K}[n]$ premiers entre eux, et on injecte dans l'équation :

$$a(n)p(n+1)q(n) - b(n-1)q(n+1)p(n) = q(n)q(n+1)c(n).$$

On a donc $q(n)|b(n-1)q(n+1)p(n)$ d'où $q(n)|b(n-1)q(n+1)$. De même $q(n+1)|a(n)q(n)$. En itérant, pour tout $K \in \mathbb{N}^*$ on déduit

$$q(n)|b(n-1)b(n)\cdots b(n+K-2)q(n+K), \quad q(n)|a(n-1)a(n-2)\cdots a(n-K)q(n-K).$$

En choisissant K assez grand pour que $\text{pgcd}(q(n), q(n+K)) = 1$, on en déduit

$$q(n)|\text{pgcd}(b(n-1)b(n)\cdots b(n+K-2), a(n-1)a(n-2)\cdots a(n-K)) = 1,$$

où la dernière égalité est conséquence des hypothèses sur a et b . \square

Le principe de l'algorithme de Gosper est donné en Algorithme 1. Il dépend de deux autres algorithmes, pour le calcul de la forme de Gosper et pour la recherche de solutions polynomiales.

Algorithme 1 Algorithme de Gosper

Entrées: $r(n) \in \mathbb{K}(n)$ telle que $\forall n \in \mathbb{N}$, $u_{n+1}/u_n = r(n)$.

Sorties: $f(n)$ telle que la suite hypergéométrique définie par $\forall n \in \mathbb{N}$, $v_n = f(n)u_n$ vérifie $\forall n \in \mathbb{N}$, $v_{n+1} - v_n = u_n$, ou FAIL si (u_n) n'admet pas de somme hypergéométrique.

- 1: Calculer une forme de Gosper de $r(n)$.
 - 2: Trouver une solution polynomiale $x(n) \in \mathbb{K}[n]$ de (2).
 - 3: **si** il n'y a pas de solution **alors**
 - 4: **renvoyer** FAIL
 - 5: **sinon**
 - 6: **renvoyer** $\frac{b(n-1)}{c(n)}x(n)$
 - 7: **fin si**
-

Algorithme 2 Forme de Gosper

Entrées: $r(n) = P(n)/Q(n)$, $\text{pgcd}(P, Q) = 1$

Sorties: $a, b, c \in \mathbb{K}[n]$ obéissant à (1) et tels que $\forall k \in \mathbb{N}$, $\text{pgcd}(a(n), b(n+k)) = 1$.

- 1: $R(k) := \text{Res}_n(P(n), Q(n+k))$
 - 2: Calculer $0 < h_1 < h_2 < \cdots < h_N$ les racines entières de R .
 - 3: $a(n) := P(n); b(n) := Q(n); c(n) := 1$
 - 4: **pour** i allant de 1 à N **faire**
 - 5: $g_i := \text{pgcd}(a(n), b(n+h_i))$
 - 6: $a(n) := a(n)/g_i(n)$
 - 7: $b(n) := b(n)/g_i(n)$
 - 8: $c(n) := c(n)g_i(n-1)g_i(n-2)\cdots g_i(n-h_i)$
 - 9: **fin pour**
 - 10: **renvoyer** a, b, c .
-

Le calcul de la forme de Gosper est donné en Algorithme 2. Dans un premier temps, il calcule l'ensemble des différences entières possibles entre les racines du numérateur et du dénominateur. Dans un second temps (la boucle), ces différences sont accumulées dans le polynôme c . La correction de l'algorithme provient d'un invariant simple à observer par récurrence : à chaque itération, (1) est satisfaite, et après l'étape i , $\text{pgcd}(a(n), b(n + h_i)) = 1$. Ces deux propriétés restent donc vraies à la fin de l'algorithme.

La recherche de solutions polynomiales de l'équation (2) procède aussi en deux temps.

D'abord, on va chercher une borne sur le degré de x , ensuite on peut résoudre par coefficients indéterminés (le système est même linéaire). La discussion sur le degré distingue deux cas :

1. Cas facile : $\deg a \neq \deg b$, ou bien $\deg a = \deg b$ et $\text{lc}(a) \neq \text{lc}(b)$ ¹. Alors le degré du membre gauche de (2) est donné par celui d'un de ses deux termes et on en déduit $\deg x \leq \deg c - \max(\deg a, \deg b)$.
2. Cas restant : on note $a(n) = \lambda n^d + \alpha n^{d-1} + \dots$ $b(n-1) = \lambda n^d + \beta n^{d-1} + \dots$ $x(n) = \gamma n^D + \dots$ où "... " désigne des éléments de degré inférieur à celui des premiers termes, $\lambda \neq 0, \gamma \neq 0$ (soit $d = \deg a = \deg b$ et $D = \deg x$). Alors on peut réécrire l'équation :

$$\underbrace{a(n)}_{\lambda n^d + \dots} \underbrace{(x(n+1) - x(n))}_{\gamma D n^{D-1} + \dots} + \underbrace{(a(n) - b(n-1))}_{(\alpha - \beta)n^{d-1}} \underbrace{x(n)}_{\gamma n^D + \dots} = c(n)$$

qui donne

$$(\lambda \gamma D + (\alpha - \beta)\gamma)n^{d+D-1} + \dots = c(n)$$

Alors, soit $D \leq \deg c - d + 1$, soit $\lambda \gamma D + (\alpha - \beta)\gamma = 0$, d'où $D = (\beta - \alpha)/\lambda$ (car $\gamma \neq 0$). On obtient donc dans ce cas $D \leq \max(\deg c - d + 1, \frac{\beta - \alpha}{\lambda})$.

Une fois le degré de x borné, on peut résoudre l'équation par coefficients indéterminés.

2. Sommation définie

Dans toute cette partie on suppose de plus que \mathbb{K} est un corps topologique.

2.1. Problème de la sommation définie.

Définition 3. On dit qu'une suite $(F_{n,k}) \in \mathbb{K}^{\mathbb{N} \times \mathbb{N}}$ est hypergéométrique si $F_{n+1,k}/F_{n,k}$ et $F_{n,k+1}/F_{n,k}$ sont des fractions rationnelles dans $\mathbb{K}(n, k)$.

Problème 2. Etant donné une suite hypergéométrique $(F_{n,k}) \in \mathbb{K}^{\mathbb{N} \times \mathbb{N}}$, on cherche une récurrence linéaire à coefficients polynomiaux pour la suite (u_n) définie par

$$\forall n \in \mathbb{N}, \quad u_n = \sum_{k \in \mathbb{N}} F_{n,k}$$

Exemple 2. Des identités typiques de ce qui est calculable dans ce contexte sont :

$$\sum_{k=0}^n \binom{n}{k} = 2^n, \quad \sum_{k=0}^n \binom{n}{k}^2 = \binom{2n}{n}.$$

1. Si $P \in \mathbb{K}[n]$, on note $\text{lc}(P)$ son coefficient dominant (*leading coefficient*).

2.2. Algorithme de Zeilberger.

Idée 2. (Zeilberger) On cherche des polynômes $t_0(n), \dots, t_p(n) \in \mathbb{K}[n]$ et $G_{n,k}$ hypergéométrique tels que

$$(3) \quad t_0(n)F_{n,k} + t_1(n)F_{n+1,k} + \dots + t_p(n)F_{n+p,k} = G_{n,k+1} - G_{n,k}.$$

Cela permet de trouver une solution du problème en sommant sur k : si $\lim_{k \rightarrow \pm\infty} G_{n,k} = 0$, alors

$$t_0(n)u_n + t_1(n)u_{n+1} + \dots + t_p(n)u_{n+p} = 0.$$

L'algorithme de Zeilberger part de l'observation que $G_{n,k}$ est une somme indéfinie en k d'une suite hypergéométrique. En effet, avec $g_{n,k}$ le membre gauche de (3), on a

$$\frac{g_{n,k+1}}{g_{n,k}} = \frac{\sum_i t_i(n)F_{n+i,k+1}}{\sum_i t_i(n)F_{n+i,k}}$$

qui se réécrit

$$\frac{g_{n,k+1}}{g_{n,k}} = \frac{\sum_{i=0}^p t_i(n) \frac{F_{n+i,k+1}}{F_{n,k+1}} F_{n,k+1}}{\sum_{i=0}^p t_i(n) \frac{F_{n+i,k}}{F_{n,k}} F_{n,k}} \in \mathbb{K}(n, k).$$

L'idée est alors d'utiliser l'algorithme de Gosper dans le corps $\mathbb{K}(n)$, avec des t_i indéterminés.

1. Posons $P_n(k) = \sum_{i=0}^p t_i(n)F_{n+i,k}/F_{n,k}$, $Q_n(k) = F_{n,k+1}/F_{n,k}$ et $R_n(k) = g_{n,k+1}/g_{n,k}$. Alors P_n dépend linéairement des t_i , Q_n n'en dépend pas, et on a

$$R_n(k) = \frac{P_n(k+1)}{P_n(k)} Q_n(k)$$

Ainsi, en mettant Q_n sous forme de Gosper

$$Q_n(k) = \frac{A_n(k)}{B_n(k)} \frac{c_n(k+1)}{c_n(k)}$$

et en posant $C_n(k) = P_n(k)c_n(k)$, on obtient une forme de Gosper pour R_n

$$R_n(k) = \frac{A_n(k)}{B_n(k)} \frac{C_n(k+1)}{C_n(k)}$$

où A_n et B_n ne dépendent pas des t_i , et C_n en dépend linéairement.

2. L'équation qu'il faut alors résoudre est

$$A_n(k)X(k+1) - B_n(k-1)X(k) = C_n(k)$$

Dans cette équation, seul C_n dépend des t_i , et de façon linéaire. De plus, dans l'algorithme de Gosper, les bornes sur le degré du polynôme inconnu dépendent uniquement de A_n et de B_n , qui ne dépendent pas des t_i , et du degré en k de C_n qui peut être borné indépendamment des t_i . Cette équation est donc un système linéaire fini, d'inconnues à la fois les coefficients de X et les t_i , qu'on peut résoudre.

3. Il suffit alors d'essayer des ordres p successivement dans l'espoir de trouver une récurrence.

Remarque 2. Un théorème de Wilf & Zeilberger assure l'existence d'une telle récurrence (et donc la terminaison de l'algorithme) pour une sous-classe des suites hypergéométriques appelées "proprement" hypergéométriques, c'est-à-dire celles qui peuvent s'écrire sous la forme

$$P(n, k)A^k \frac{\prod_{i=1}^p (a_i n + b_i k + c_i)!}{\prod_{j=1}^q (u_j n + v_j k + w_j)!},$$

où P est un polynôme, les a_i, b_i, u_j, v_j sont des entiers, et p et q sont des entiers positifs ou nuls.

Séries pour la fonction Zêta de Riemann aux entiers positifs

La fonction ζ de Riemann est définie pour $\Re s > 1$ par

$$(Z) \quad \zeta(s) = \sum_{n=1}^{\infty} \frac{1}{n^s}.$$

Cette série converge, mais sa vitesse de convergence n'est pas très élevée. Il existe des formules plus rapides, dont beaucoup ont été prouvées dans les dix dernières années via l'algorithme de Zeilberger.

1. Une première série pour $\zeta(3)$

Andrei Andreevich Markov a prouvé en 1890 l'identité

$$(Z_3) \quad \zeta(3) = \frac{5}{2} \sum_{n=1}^{\infty} \frac{(-1)^{n-1}}{\binom{2n}{n} n^3}.$$

1. Évaluer numériquement les vingt premiers sommants, et la somme. En déduire la précision obtenue. Comparer au résultat de l'évaluation des mille premiers sommants de (Z) en $s = 3$.

Pour prouver (Z_3) , l'algorithme de Zeilberger part de

$$(A) \quad F_{n,k} = \frac{(-1)^k k!^2 (n-k-1)!}{(n+k+1)!(k+1)}.$$

2. Calculer une suite $G_{n,k}$ telle que

$$F_{n+1,k} - F_{n,k} = G_{n,k+1} - G_{n,k}.$$

(On pourra utiliser l'implantation de l'algorithme de Zeilberger disponible en Maple dans le package `SumTools`).

3. (Cette question n'utilise pas de calcul formel, et pourra être admise dans un premier temps). En sommant cette identité d'abord de $k = 0$ à $n - 1$, puis de $n = 0$ à l'infini, montrer que si les sommes concernées convergent, alors

$$(S_1) \quad \sum_{n=0}^{\infty} G_{n,0} = \sum_{n \geq 0} (G_{n,n} + F_{n+1,n}).$$

4. En déduire l'équation (Z_3) en faisant attention aux formes indéterminées.
5. (À la fin, s'il reste du temps). Vérifier les convergences requises sur l'exemple.

2. Une famille de séries de plus en plus rapidement convergentes

6. Calculer le comportement asymptotique du n^{e} sommant de (Z_3) .

Une généralisation de (Z_3) proposée par Amdeberhan en 1996 part de

$$F_{n,k} = \frac{(-1)^k k!^2 (sn - k - 1)!}{(sn + k + 1)! (k + 1)}.$$

Le cas particulier $s = 1$ redonne (A) .

7. Appliquer la même méthode que ci-dessus pour $s = 2$. En déduire une représentation sommatoire de $\zeta(3)$. Calculer le comportement asymptotique du n^{e} sommant.

8. Recommencer avec $s = 3$.

3. D'autres valeurs de $\zeta(2n + 3)$

De nombreuses identités plus ou moins récentes existent également sur les valeurs $\zeta(2n + 3)$ ². Un article de 2008 de Kh. et T. Hessami Pilehrood montre comment obtenir certaines de ces identités par l'algorithme de Zeilberger.

9. Appliquer l'algorithme de Zeilberger à

$$F_{n,k} = \frac{(-1)^n (1+a)_n (1-a)_n}{\Gamma(1+a)\Gamma(1-a)} \frac{k!}{(2n+k+1)!((n+k+1)^2 - a^2)},$$

où $(x)_n = x(x+1)\cdots(x+n-1)$ est le symbole de Pochhammer.

10. En sommant d'abord sur $n \geq 0$, puis sur $k \geq 0$, on obtient comme en question 3

$$(S_2) \quad \sum_{k=0}^{\infty} F_{0,k} = \sum_{n=0}^{\infty} G_{n,0}.$$

Vérifier que la formule obtenue est équivalente à

$$\sum_{k=1}^{\infty} \frac{1}{k(k^2 - a^2)} = \sum_{n=0}^{\infty} \frac{(-1)^n (1+a)_n (1-a)_n (5(n+1)^2 - a^2)}{(2n+2)!(2n+2)((n+1)^2 - a^2)}.$$

11. Développer en série par rapport à a et obtenir par extraction des coefficients de a non seulement l'identité (Z_3) , mais une identité pour $\zeta(5)$. (On pourrait bien sûr continuer et obtenir des formules pour tous les $\zeta(2n + 3)$).

12. La généralisation suivante étend la convergence du cas $s = 2$ de la question 7 à tous les $\zeta(2n + 3)$. Mener le calcul en partant de

$$F_{n,k} = \frac{(-1)^n k! (n-1)! (2n)! (1+a)_k (1-a)_k (1+a)_n (1-a)_n (1+a)_{2n} (1-a)_{2n}}{(2n+k+1)! (3n-1)! (1+a)_{2n+k+1} (1-a)_{2n+k+1}}.$$

2. Les valeurs de ζ aux entiers pairs sont connues depuis Euler : elle s'écrivent

$$\zeta(2n) = (-1)^{n+1} \frac{B_{2n}}{2(2n)!} (2\pi)^{2n}, \quad n = 1, 2, 3, \dots$$

où les constantes $B_{2n}/(2n)!$ sont les coefficients de la série $z/(\exp(z) - 1)$, et les B_n sont appelés *nombre de Bernoulli*.

Résultants : propriétés et calcul euclidien

Résumé

On introduit ici la notion de résultant, utile notamment à calculer des intersections d'ensembles algébriques en éliminant des variables dans les systèmes polynomiaux. Dans un premier temps, on expose les propriétés générales de cet objet, agrémentées de plusieurs applications, puis on présente un algorithme en permettant le calcul.

Dans toute la suite, \mathbb{K} désignera un corps commutatif; on pourrait se placer dans un anneau commutatif quelconque, mais cela nécessiterait plus de précautions. Au besoin, on verra un anneau intègre comme un sous-anneau de son corps des fractions. On notera $\text{lc}(f)$ le coefficient dominant d'un polynôme f (*leading coefficient*).

1. Définition et premières propriétés

Définition 1. Soient $f = a_m X^m + \dots + a_0$ et $g = b_n X^n + \dots + b_0$ deux polynômes à coefficients dans \mathbb{K} , de degrés respectifs m et n . On suppose qu'au moins un des deux polynômes n'est pas constant. On définit alors la matrice de Sylvester¹ de f et g comme

$$\text{Syl}(f, g) = \begin{pmatrix} a_m & \dots & a_0 & & & 0 \\ & \ddots & & & & \\ & & a_m & \dots & a_0 & \\ 0 & & b_n & \dots & b_0 & 0 \\ & \ddots & & & & \\ & & \ddots & & & \\ 0 & & & b_n & \dots & b_0 \end{pmatrix}$$

où les n premières lignes sont occupées par les coefficients de f et les m dernières par ceux de g . La matrice $\text{Syl}(f, g)$ est alors carrée et de taille $m + n$.

Le résultant de f et g est par définition $\text{Res}(f, g) = \det \text{Syl}(f, g)$. En cas d'ambiguïté sur l'indéterminée par rapport à laquelle est calculé le résultant, on sera parfois amené à la préciser, en notant par exemple $\text{Res}_X(f, g)$. On remarque que, si g est un polynôme constant, alors $\text{Res}(f, g) = g^{\deg f}$. Enfin, si f n'est pas constant, on définit son discriminant comme $\text{disc}(f) = (-1)^{\frac{m(m-1)}{2}} a_m^{-1} \text{Res}(f, f')$.

Exemple 1. On a immédiatement $\text{Res}(aX + b, cX + d) = ad - bc$. En outre, $\text{disc}(aX^2 + bX + c) = b^2 - 4ac$ et $\text{disc}(X^3 - pX - q) = 4p^3 - 27q^2$; on retrouve donc les notions de discriminant d'équations polynomiales du second et du troisième

1. Notion introduite en 1839 par le mathématicien britannique James Joseph Sylvester. Le lecteur curieux (et courageux) pourra consulter en ligne l'article fondateur [2].

degrés. On verra que les résultants et discriminants peuvent s'interpréter en termes de compatibilité d'équations polynomiales, ou d'existence de solutions à de telles équations.

Le résultant et le discriminant sont disponibles en Maple sous les appellations `resultant` et `discrim`.

Remarque 1. La matrice de Sylvester de f est g est en fait la transposée de la matrice dans des bases convenables de l'application linéaire de $\mathbb{K}_{n-1}[X] \times \mathbb{K}_{m-1}[X]$ dans $\mathbb{K}_{m+n-1}[X]$ définie par $(u, v) \mapsto uf + vg$.

Remarque 2. Si f et g ont une racine commune $z \in \overline{\mathbb{K}}$, alors ${}^t(z^{m+n-1}, \dots, 1)$ est un vecteur non nul du noyau de $\text{Syl}(f, g)$, et donc $\text{Res}(f, g) = 0$. En particulier, si $\text{disc}(f) \neq 0$, alors f est séparable, et on verra au point (7) du théorème ci-dessous que la réciproque est vraie.

Le théorème suivant énumère quelques propriétés fondamentales du résultant.

Théorème 13. Soient f et g deux polynômes de degrés respectifs m et n , à coefficients dans \mathbb{K} .

1. Le résultant est quasi-symétrique : $\text{Res}(g, f) = (-1)^{mn} \text{Res}(f, g)$;
2. Il est homogène : si $\lambda, \mu \in \mathbb{K}$, alors $\text{Res}(\lambda f, \mu g) = \lambda^n \mu^m \text{Res}(f, g)$;
3. Il vérifie la formule de Poisson : si f et g se décomposent dans $\overline{\mathbb{K}}$ sous la forme $f = a_m(X - x_1) \cdots (X - x_m)$ et $g = b_n(X - y_1) \cdots (X - y_n)$,

$$\text{Res}(f, g) = a_m^n b_n^m \prod_{i,j} (x_i - y_j) = a_m^n \prod_{i=1}^m g(x_i) = (-1)^{mn} b_n^m \prod_{j=1}^n f(y_j);$$

4. Le résultant est multiplicatif, au sens où, si h est un troisième polynôme, $\text{Res}(fg, h) = \text{Res}(f, h) \text{Res}(g, h)$ et $\text{Res}(f, gh) = \text{Res}(f, g) \text{Res}(f, h)$;
5. Il se transporte par spécialisation : pour tout morphisme de corps $\varphi: \mathbb{K} \rightarrow \mathbb{L}$,

$$\varphi(\text{Res}(f, g)) = \text{Res}(\varphi(f), \varphi(g))$$

en notant encore $\varphi: \mathbb{K}[X] \rightarrow \mathbb{L}[X]$ le morphisme d'algèbres induit par φ ;

6. Il vérifie la propriété d'élimination : il existe deux polynômes u et v de $\mathbb{K}[X]$ avec $(u, v) \neq (0, 0)$, $\deg u < n$, $\deg v < m$ et

$$\text{Res}(f, g) = uf + vg$$

donc en particulier $\text{Res}(f, g) \in (f, g) \cap \mathbb{K}$;

7. Le résultant est cohérent, au sens où on a équivalence entre

- (a) $\text{Res}(f, g) = 0$,
- (b) $f \wedge g \neq 1$,
- (c) f et g ont une racine commune dans $\overline{\mathbb{K}}$;

8. Enfin, il vérifie l'inégalité de Bézout-Hadamard : si f et g sont cette fois des éléments de $\mathbb{K}[X, Y]$,

$$\deg_Y \text{Res}_X(f, g) \leq \max(\deg_X f \cdot \deg_Y g + \deg_Y f \cdot \deg_X g, \deg f \cdot \deg g).$$

DÉMONSTRATION. Les assertions (1), (2) et (5) découlent directement de la définition du résultant et des propriétés de base du déterminant. On remarquera que (2) utilise l'intégrité de \mathbb{K} et (5) l'injectivité du morphisme de corps φ (dans un anneau quelconque, φ aurait pu annuler les coefficients dominants).

Preuve de (3) : on va raisonner avec des racines formelles X_i et Y_j , et donc avec des polynômes à coefficients dans $\mathbb{K}(X_1, \dots, X_m, Y_1, \dots, Y_n)$, ce qui neutralisera les éventuels problèmes de division par zéro. Par définition de la matrice de Sylvester, et puisque les X_i (resp. Y_j) sont racines de f (resp. g), on a

$$\begin{aligned} & \text{Syl}(f, g) \begin{pmatrix} Y_1^{n+m-1} & \cdots & Y_n^{n+m-1} & X_1^{n+m-1} & \cdots & X_m^{n+m-1} \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ 1 & \cdots & 1 & 1 & \cdots & 1 \end{pmatrix} = \\ & = \begin{pmatrix} Y_1^{n-1}f(Y_1) & \cdots & Y_n^{n-1}f(Y_n) & X_1^{n-1}f(X_1) & \cdots & X_m^{n-1}f(X_m) \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ g(Y_1) & \cdots & g(Y_n) & g(X_1) & \cdots & g(X_m) \end{pmatrix} \\ & = \left(\begin{array}{ccc|ccc} Y_1^{n-1}f(Y_1) & \cdots & Y_n^{n-1}f(Y_n) & & & \\ \vdots & \ddots & \vdots & & & \\ f(Y_1) & \cdots & f(Y_n) & & & \\ \hline & & & X_1^{m-1}g(X_1) & \cdots & X_m^{m-1}g(X_m) \\ & & 0 & \vdots & \ddots & \vdots \\ & & & g(X_1) & \cdots & g(X_m) \end{array} \right) \end{aligned}$$

par conséquent, en prenant le déterminant, en développant par multilinéarité le membre de droite et en utilisant deux fois l'identité classique sur les déterminants de Vandermonde, on obtient

$$\begin{aligned} \text{Res}(f, g) \prod_{j>i} (Y_j - Y_i) \prod_{j,i} (Y_j - X_i) \prod_{j>i} (X_j - X_i) &= \\ &= \prod_{j=1}^n f(Y_j) \prod_{i=1}^m g(X_i) \prod_{j>i} (Y_j - Y_i) \prod_{j>i} (X_j - X_i) \end{aligned}$$

d'où

$$\text{Res}(f, g) \prod_{j,i} (Y_j - X_i) = \prod_{j=1}^n f(Y_j) \prod_{i=1}^m g(X_i)$$

et le résultat.

Preuve de (4) : quitte à se placer dans la clôture algébrique de \mathbb{K} , on peut supposer scindés tous les polynômes considérés, et le résultat provient alors de la formule de Poisson.

Preuve de (6) : supposons $\text{Res}(f, g) \neq 0$. En ajoutant à la dernière colonne de la matrice de Sylvester les autres colonnes multipliées par une puissance adéquate

de X , on obtient

$$\text{Res}(f, g) = \begin{vmatrix} a_m & \dots & a_0 & 0 & X^{n-1}f \\ & \ddots & & & \vdots \\ & & & & \vdots \\ 0 & & a_m & \dots & a_0 & Xf \\ b_n & \dots & b_0 & 0 & X^{m-1}g \\ & \ddots & & & \vdots \\ 0 & & b_n & \dots & b_1 & Xg \\ & & & & & g \end{vmatrix}$$

ce qui, en développant le déterminant par rapport à la dernière colonne, fournit l'existence de u et v et la majoration de leur degré annoncée. Par conséquent $\text{Res}(f, g) = uf + vg$, et u et v ne peuvent être simultanément nuls, puisque par hypothèse $\text{Res}(f, g) \neq 0$.

Enfin, si $\text{Res}(f, g) = 0$, la remarque 1 implique l'existence d'un couple non nul (u, v) avec $uf + vg = 0 = \text{Res}(f, g)$, les degrés de u et v satisfaisant les majorations voulues.

Preuve de (7) : procédons par triple implication.

(a) \Rightarrow (b) Si $\text{Res}(f, g) = 0$, le point précédent ou la remarque 1 donnent l'existence d'un couple non nul (u, v) de polynômes avec $uf + vg = 0$, $\deg u < n$ et $\deg v < m$. Si f et g étaient premiers entre eux, on devrait avoir $f \mid v$ et $g \mid u$, soit après examen des degrés $u = v = 0$ ce qui est absurde. Donc $f \wedge g \neq 1$.

(b) \Rightarrow (c) Une racine de $f \wedge g$ dans $\overline{\mathbb{K}}$ fournit une racine commune à f et g .

(c) \Rightarrow (a) Ce fait a déjà été vu dans la remarque 2.

Preuve de (8) : on écrit explicitement les coefficients s_{ij} de la matrice de Sylvester de f et g . On a

$$s_{ij} = \begin{cases} a_{m-j+i} & \text{si } i \in \{1, \dots, n\} \text{ et } j \in \{i, \dots, i+m\} \\ b_{i-j} & \text{si } i \in \{n+1, \dots, n+m\} \text{ et } j \in \{i-n, \dots, i-n+m\} \\ 0 & \text{sinon.} \end{cases}$$

Par ailleurs, $\text{Res}_X(f, g) = \sum_{\sigma \in \mathfrak{S}_{m+n}} \prod_{i=1}^{m+n} s_{i\sigma(i)}$ et

$$\deg_Y \prod_{i=1}^{m+n} s_{i\sigma(i)} = \sum_{i=1}^n \deg_Y a_{m-\sigma(i)+i} + \sum_{i=n+1}^{n+m} \deg_Y b_{i-\sigma(i)}.$$

Par conséquent, comme $\deg_Y a_{m-\sigma(i)+i} \leq \deg_Y f$ et $\deg_Y b_{i-\sigma(i)} \leq \deg_Y g$,

$$\deg_Y \text{Res}_X(f, g) \leq n \deg_Y f + m \deg_Y g = \deg_X f \cdot \deg_Y g + \deg_Y f \cdot \deg_X g.$$

Enfin, comme $\deg_Y a_i \leq \deg f - i$ et $\deg_Y b_i \leq \deg g - i$,

$$\begin{aligned}
\deg_Y \prod_{i=1}^{m+n} s_{i\sigma(i)} &\leq \sum_{i=1}^n (\deg f - m + \sigma(i) - i) + \sum_{i=n+1}^{n+m} (\deg g - i + \sigma(i)) \\
&= n \deg f - nm + \sum_{i=1}^n (\sigma(i) - i) + m \deg g + \sum_{i=n+1}^{n+m} (\sigma(i) - i) \\
&= n \deg f + m \deg g - nm + \sum_{i=1}^{n+m} \sigma(i) - \sum_{i=1}^{n+m} i \\
&= n \deg f + m \deg g - nm \\
&= \deg f \deg g - (\deg f - m)(\deg g - n)
\end{aligned}$$

or $\deg f \geq \deg_X f = m$ et $\deg g \geq n$, donc $\deg_Y \prod_{i=1}^{m+n} s_{i\sigma(i)} \leq \deg f \deg g$, ce qui clôt la démonstration. \square

2. Applications

2.1. Une forme faible du théorème de Bézout.

Théorème 14. *Soient f et g deux polynômes homogènes premiers entre eux de $\mathbb{K}[X, Y, Z]$, de degrés respectifs d_1 et d_2 . Alors l'intersection des courbes de $\mathbb{P}^2(\mathbb{K})$ définies par f et g comporte au plus $d_1 d_2$ points.*

DÉMONSTRATION. L'idée de la preuve est d'injecter les zéros projectifs communs de f et g dans l'ensemble des zéros sur \mathbb{K} d'un polynôme non nul de degré au plus $d_1 d_2$, que l'on construit comme un résultant. Remarquons d'abord que l'on peut remplacer \mathbb{K} par une extension infinie sans changer la majoration sur le cardinal de l'intersection, et qu'il suffit de montrer que si l'on a plus de deux points d'intersection, alors on en a moins de $d_1 d_2$.

Soient donc p_1, \dots, p_k des points d'intersection distincts des deux courbes, avec k supérieur à 2. Il est classique que, \mathbb{K} étant supposé infini, \mathbb{K}^3 ne peut être réunion finie de sous-espaces stricts : il existe donc $p_0 \in \mathbb{P}^2(\mathbb{K})$ hors des $\binom{k}{2}$ droites joignant deux à deux les p_i . Choisissons alors un système de coordonnées projectives tel que p_0 soit le point à l'infini de coordonnées projectives $[0, 0, 1]$, et dans lequel p_i a pour coordonnées $[x_i, y_i, z_i]$.

On « projette » maintenant chaque p_i sur $[x_i, y_i] \in \mathbb{P}^1(\mathbb{K})$. Cette projection est bien définie et injective du fait que, comme p_0, p_i et p_j ne sont pas alignés pour $i \neq j$, la matrice

$$\begin{pmatrix} 0 & x_i & x_j \\ 0 & y_i & y_j \\ 1 & z_i & z_j \end{pmatrix}$$

est inversible. D'autre part, ces projections sont racines de $r = \text{Res}_Z(f, g)$, où l'on appelle encore f et g les polynômes obtenus après changement de coordonnées.

Ce polynôme est homogène et de degré inférieur à $d_1 d_2$ par l'inégalité de Bézout–Hadamard ; par le théorème 13, et parce que le pgcd est invariant par extension de corps et changement de coordonnées homogènes, il est de plus non nul. Le polynôme r a donc au plus $d_1 d_2$ zéros projectifs dont font partie les projections des p_i , ce qui fournit le résultat annoncé. \square

2.2. Manipulation de nombres et d'entiers algébriques. Le résultant permet de calculer des polynômes annulant la somme, le produit ou les inverses de nombres algébriques :

Proposition 1. *Soient f et g deux polynômes de degrés respectifs m et n . Alors on peut calculer des polynômes annulant les sommes et les produits des racines de f et g (dans $\overline{\mathbb{K}}$), ainsi que leurs inverses si ces racines sont non nulles, par les résultants*

$$\text{Res}_Y(f(X - Y), g(Y)), \quad \text{Res}_Y(Y^m f\left(\frac{X}{Y}\right), g(Y)), \quad \text{Res}_Y(XY - 1, f(Y)).$$

DÉMONSTRATION. Ce résultat découle immédiatement de la partie (7) du théorème 13. Par exemple, en notant $R(X) = \text{Res}_Y(f(X - Y), g(Y))$, pour tout $x \in \overline{\mathbb{K}}$, $R(x) = 0$ si et seulement s'il existe $y \in \overline{\mathbb{K}}$ tel que $f(x - y) = g(y) = 0$, c'est à dire si x est somme d'une racine de f et d'une racine de g . \square

On obtient par conséquent une preuve constructive du fait que l'ensemble des nombres algébriques d'une extension forme un corps, et que l'ensemble des entiers algébriques d'une extension de \mathbb{Q} forme un anneau, car le résultant de deux polynômes unitaires à coefficients entiers l'est encore.

2.3. Implication de courbes unicursales. Supposons donnée une courbe plane paramétrée par des fractions rationnelles, disons de la forme $\{x = p_1(t)/q_1(t), y = p_2(t)/q_2(t)\}$. Le théorème de Lüroth implique (voir par exemple [1, p. 94]) que cette courbe est algébrique et irréductible. On peut en fait calculer une équation implicite polynomiale dont les points de la courbe seront solutions grâce au résultant

$$\text{Res}_t(p_1(t) - xq_1(t), p_2(t) - yq_2(t)).$$

La justification de cet énoncé se fait de la même façon que dans la preuve de la proposition 1. Par exemple, le cercle unité est paramétré rationnellement (en caractéristique différente de 2) par

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

ce qui donne après calcul l'équation implicite

$$x^2 + y^2 = 1.$$

2.4. Certification d'identités algébriques. Le résultant permet de certifier dans certains cas qu'un polynôme annule bien une série algébrique donnée :

Proposition 2. *Soit $F \in \mathbb{K}[[X]]$ une série formelle algébrique de degré inférieur à d . Supposons connu un polynôme $f \in \mathbb{K}[X, Y]$ tel que $f(X, F) = O(X^\sigma)$ avec $\sigma > d \deg f$. Alors $f(X, F) = 0$.*

DÉMONSTRATION. Soit $g \in \mathbb{K}[X, Y]$ un polynôme irréductible de degré inférieur à d tel que $g(X, F) = 0$. Le polynôme $\text{Res}_Y(f(X, Y), g(X, Y)) \in \mathbb{K}[X]$ est de degré inférieur à $d \deg f < \sigma$; comme il est de la forme $uf + vg$ avec u et v deux polynômes, avec $\deg u < \deg g$, et qu'il ne change pas quand on remplace la variable Y par la série F , c'est un $O(X^\sigma)$. Par conséquent $uf + vg = 0$, donc g divise uf et, étant irréductible, il divise u ou f . Mais $\deg u < \deg g$, donc finalement $g \mid f$ et $f(X, F)$ est la série nulle. \square

3. Calcul

La proposition suivante, facile à prouver en utilisant le théorème 13, donne un moyen rapide de calculer le résultant de deux polynômes.

Proposition 3. *Soient f et g dans $\mathbb{K}[X]$ de degrés respectifs m et n , avec g non constant, et soit $f = gq + r$ la division euclidienne de f par g . Alors*

$$\text{Res}(f, g) = (-1)^{mn} \text{lc}(g)^{m - \deg r} \text{Res}(g, r).$$

DÉMONSTRATION. En appliquant deux fois la formule de Poisson,

$$\begin{aligned} \text{Res}(g, f) &= \text{lc}(g)^m \prod_{g(\alpha)=0} f(\alpha) \\ &= \text{lc}(g)^m \prod_{g(\alpha)=0} r(\alpha) \\ &= \text{lc}(g)^{m - \deg r} \text{Res}(g, r) \end{aligned}$$

d'où le résultat par quasi-symétrie du résultant (dans le cas où $r = 0$, on a bien $\text{Res}(f, g) = 0$). \square

Grâce à cette proposition, on peut écrire un algorithme itératif semblable à l'algorithme d'Euclide, qui permet de calculer en $O(d^2)$ le résultant de deux polynômes de degré majoré par d . C'est mieux que l'algorithme naïf en $O(d^3)$ consistant à calculer un déterminant par pivot de Gauss ; on exploite en fait la forme particulière des matrices de Sylvester.

Algorithme 1 Calcul euclidien du résultant

Entrées: deux polynômes f et g , dont l'un au moins est non constant.

Sorties: le résultant de f et g .

```

1:  $h \leftarrow f$ ;
2:  $k \leftarrow g$ ;
3:  $r \leftarrow 1$ ;
4: tant que  $\deg k > 0$  faire
5:    $s \leftarrow h \bmod k$ ;
6:    $r \leftarrow (-1)^{\deg h \deg k} \text{lc}(k)^{\deg h - \deg s} r$ ;
7:    $h \leftarrow k$ ;
8:    $k \leftarrow s$ ;
9: fin tant que
10: renvoyer  $r \cdot k^{\deg h}$ .

```

La justification de la correction de cet algorithme est immédiate en utilisant la proposition 3, dont la boucle **tant que** est la traduction littérale. La dernière action utilise la remarque page 85 sur la valeur du résultant lorsqu'un des polynômes est constant, et a pour but de contourner le problème d'annulation éventuelle du degré de s . La complexité est essentiellement la même que celle de l'algorithme d'Euclide usuel.

Remarque 3. De la même façon que cet algorithme est une adaptation de l'algorithme d'Euclide, une adaptation de l'algorithme d'Euclide étendu fournirait des polynômes u et v tels que $\text{Res}(f, g) = uf + vg$.

Bibliographie

- [1] Samuel (Pierre). – *Géométrie projective*. – PUF, Paris, 1986, 176p.
- [2] Sylvester (James Joseph). – On rational derivation from equations of coexistence, that is to say, a new and extended theory of elimination, 1839-40. In *The Collected mathematical papers of James Joseph Sylvester*, pp. 40–53. – Baker, H. F., New York, 1839.

Utilisation de résultants

1. Manipulation de nombres algébriques

1. Conjecturer puis prouver une expression algébrique de

$$\frac{\sin \frac{2\pi}{7}}{\sin^2 \frac{3\pi}{7}} - \frac{\sin \frac{\pi}{7}}{\sin^2 \frac{2\pi}{7}} + \frac{\sin \frac{3\pi}{7}}{\sin^2 \frac{\pi}{7}}.$$

2. Prouver l'identité suivante due à Ramanujan :

$$\sqrt[3]{\cos a} + \sqrt[3]{\cos 2a} + \sqrt[3]{\cos 4a} = \sqrt[3]{\frac{5 - 3\sqrt[3]{7}}{2}}, \quad \text{où } a = 2\pi/7.$$

3. Calculer

$$\sum_{P(\alpha)=0} F(\alpha), \quad \text{où } F(\alpha) = \frac{\alpha^{10}}{\alpha^2 + 1} \quad \text{et } P(\alpha) = \alpha^4 + p\alpha + q.$$

Deux approches seront employées : d'abord via le calcul du polynôme

$$\prod_{P(\alpha)=0} (T - F(\alpha));$$

ensuite l'utilisation de la série génératrice des sommes de Newton

$$\frac{XP'(X)}{P(X)} = \sum_{\substack{P(\alpha)=0 \\ i \geq 0}} \alpha^i X^{-i}.$$

2. Polynômes de Tchebychev entiers

Soit $\mathbb{Z}_k[X]$ l'ensemble des polynômes à coefficients entiers et de degrés au plus k . Il existe (au moins) un polynôme $P_k \in \mathbb{Z}_k[X]$ tel que

$$\mu_k := \max_{x \in [0,1]} |P_k(x)|$$

soit minimal. De tels polynômes sont appelés polynômes de Tchebychev entiers dans l'intervalle $[0, 1]$. Il est possible de calculer un certain nombre de facteurs de tels polynômes grâce à des observations simples sur les résultants. Cet exercice décrit une partie de ce calcul pour $k = 2m = 34$. Par symétrie, il n'est pas difficile de voir que $P_k(X) = Q_m(T)$ où $T = X(1 - X)$, et $Q_m(T) \in \mathbb{Z}_m[T]$, ce qui réduit de moitié le degré des polynômes à chercher. L'intervalle d'étude change un peu puisque

$$\mu_k = \max_{x \in [0,1]} |P_k(x)| = \max_{t \in [0, \frac{1}{4}]} |Q_m(t)|.$$

Si Q_m se factorise en $Q_m = AB$ avec A et B dans $\mathbb{Z}[T]$ et que c_m est une borne connue sur $|Q_m|$, alors pour tout $t \in [0, 1/4]$,

$$|A(t)| \cdot |B(t)| \leq c_m.$$

Si B est un facteur de Q_m , cette inégalité entraîne alors une inégalité sur le facteur A qui peut permettre de prouver sa nullité en des points bien choisis.

4. On admet que $\mu_{34} < 0.33 \times 10^{-12}$. Montrer que T est un facteur de Q_{17} .
5. Montrer que $4T - 1$ et $5T - 1$ sont aussi des facteurs de Q_{17} .
6. Montrer enfin que $29T^2 - 11T + 1$ est un facteur de Q_{17} . (C'est ici qu'il faut utiliser les résultants de manière non triviale.)

Markov a donné une inégalité sur les dérivées d'un polynôme : si P est un polynôme de degré n à coefficients réels, alors

$$\max_{a \leq x \leq b} |Q^{(r)}(x)| \leq \frac{2^r}{(b-a)^r} \frac{n^2(n^2-1^2) \cdots (n^2-(r-1)^2)}{(2r-1)!!} \max_{a \leq x \leq b} |Q(x)|,$$

où $(2i+1)!! = 1 \cdot 3 \cdot 5 \cdots (2i+1)$.

7. Montrer que $T^4 | Q_{17}$.
8. Continuer à utiliser cette inégalité pour augmenter les multiplicités des facteurs $4T - 1$, $5T - 1$, et T de Q_{17} qui peuvent être trouvés par cette méthode.

Bases de Gröbner

Résumé

Les bases de Gröbner sont un outil très important du calcul formel. Elles permettent de nombreux calculs avec des idéaux d'anneaux de polynômes, ce qui en fait une structure de données utile pour manipuler les solutions de systèmes polynomiaux.

La division Euclidienne, l'algorithme d'Euclide et l'algorithme d'Euclide étendu rendent effectifs de nombreux calculs dans $\mathbb{K}[x]$ (\mathbb{K} est un corps). En particulier, ces algorithmes fournissent

- un test de divisibilité dans $\mathbb{K}[x]$;
- un test d'appartenance à l'idéal $(P) \subset \mathbb{K}[x]$, où $P \in \mathbb{K}[x]$;
- un calcul de forme normale dans $\mathbb{K}[x]/(P)$;
- un calcul d'élimination (les résultants).

Les bases de Gröbner permettent une généralisation de ces opérations à l'anneau $\mathbb{A} = \mathbb{K}[x_1, \dots, x_n]$ des polynômes à n variables et à coefficients dans \mathbb{K} .

On utilisera la notation multi-exposant : si $\alpha = (\alpha_1, \dots, \alpha_n) \in \mathbb{N}^n$, on notera $x^\alpha = x_1^{\alpha_1} x_2^{\alpha_2} \dots x_n^{\alpha_n}$.

1. Définitions

1.1. Ordres monomiaux.

Définition 1. On appelle

- monôme : un élément de \mathbb{A} de la forme x^α où $\alpha \in \mathbb{N}^n$.
- terme : un élément de \mathbb{A} de la forme λm où $\lambda \in \mathbb{K}$ et m est un monôme.
- ordre monomial : un ordre total sur les monômes qui est compatible avec le produit (i.e. $m_1 \prec m_2 \Rightarrow mm_1 \prec mm_2$) et tel que toute suite décroissante de monômes est stationnaire.

Une conséquence simple de cette définition est que si \prec est un ordre monomial, alors 1 est le plus petit élément. En effet, si $m \prec 1$ pour un certain monôme m , alors en multipliant par m , $m^2 \prec m \prec 1$. On construit de cette manière une suite infinie strictement décroissante.

D'autre part, si $n = 1$, il n'y a qu'un ordre monomial possible, à savoir l'ordre donné par le degré.

Exemple 1. L'ordre lexicographique est un ordre monomial. Il s'agit de l'ordre défini par $x^\alpha \prec x^\beta$ si et seulement si le premier coefficient non nul de $\alpha - \beta$ est négatif. En Maple, cet ordre est noté $\text{plex}(x_1, \dots, x_n)$. Par exemple, pour $\text{plex}(x, y, z)$:

$$1 \prec z \prec z^2 \prec \dots \prec y \prec yz \prec \dots \prec y^2 \prec \dots \prec x \prec \dots$$

Exemple 2. L'ordre du degré lexicographique inverse est également un ordre monomial. Il s'agit de l'ordre défini par $x^\alpha \prec x^\beta$ si et seulement si $\sum \alpha_i < \sum \beta_i$ ou $\sum \alpha_i = \sum \beta_i$ et le *dernier* élément non nul de $\alpha - \beta$ est positif. En Maple, cet ordre est noté $\text{tdeg}(x_1, \dots, x_n)$. Par exemple, pour $\text{tdeg}(x, y, z)$:

$$1 \prec z \prec y \prec x \prec z^2 \prec \dots \prec x^2 \prec z^3 \prec \dots \prec y^3 \prec \dots \prec x^2y \prec x^3.$$

Définition 2. Un ordre monomial sur \mathbb{A} étant fixé, soit $f \in \mathbb{A}$ un polynôme. On appelle

- monôme de tête de f : le plus grand monôme de f ;
- terme de tête de f : le terme correspondant au monôme de tête. On le note $\text{LT}(f)$ (pour leading term).

La compatibilité de l'ordre monomial avec le produit entraîne la relation $\text{LT}(fg) = \text{LT}(f)\text{LT}(g)$ pour tous $f, g \in \mathbb{A}$.

1.2. Bases de Gröbner.

Définition 3. Un ordre monomial sur \mathbb{A} étant fixé, un sous-ensemble fini $G = \{g_1, \dots, g_k\}$ d'un idéal $I \subset \mathbb{A}$ est une base de Gröbner de I si $\langle \text{LT}(G) \rangle = \langle \text{LT}(I) \rangle$. (Ici, $\langle A \rangle$ désigne l'idéal engendré par la partie A).

Il n'y a pas unicité des bases de Gröbner. Par exemple, si G est une base de Gröbner d'un idéal I et si $g \in I$ alors $G \cup \{g\}$ est encore une base de Gröbner de I .

Bien que ce ne soit pas évident d'après la définition, on verra plus loin que si G est une base de Gröbner d'un idéal I , alors G engendre I .

Exemple 3. Si $n = 1$, $\mathbb{A} = \mathbb{K}[x]$ est un anneau principal et si $I \subset \mathbb{K}[x]$ est un idéal, il existe $g \in I$ tel que $I = \langle g \rangle$. On a alors $\langle \text{LT}(g) \rangle = \langle \text{LT}(I) \rangle$. Inversement, si $\langle \text{LT}(G) \rangle = \langle \text{LT}(I) \rangle$ avec $G \subset I = \langle g \rangle$ alors il existe une constante $c \in \mathbb{K}$ telle que $cg \in G$. Ainsi, une base de Gröbner d'un idéal I contient nécessairement un polynôme qui engendre I . Par exemple, si $f_1, \dots, f_n \in \mathbb{K}[x]$, alors une base de Gröbner de l'idéal engendré par les f_i contient un *pgcd* des f_i .

Exemple 4. Si $A = (a_{ij})$ est une matrice en forme échelon dans $\mathbb{K}^{m \times n}$, alors l'idéal

$$\left\langle \sum_{j=1}^n a_{ij}x_j, 1 \leq i \leq m \right\rangle \subset \mathbb{A}$$

admet l'ensemble $\{\sum_{j=1}^n a_{ij}x_j\}$ comme base de Gröbner pour l'ordre lexicographique. Ainsi, les bases de Gröbner généralisent à la fois le *pgcd* et la réduction de Gauss.

Exemple 5. Considérons le système

$$f_1 = x^2 + y^2 - 4, \quad f_2 = xy - 1$$

correspondant aux points d'intersections d'un cercle et d'une hyperbole :

Une base de Gröbner pour $\text{plex}(x, y)$ est donnée par le système

$$y^4 - 4y^2 + 1, \quad x + y^3 - 4y.$$

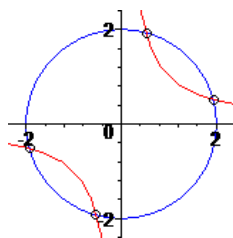
Le premier polynôme admet pour racines les ordonnées des points d'intersections.

Le second permet de calculer les valeurs des abscisses correspondantes.

Pour l'ordre $\text{tdeg}(x, y)$, le système suivant est une base de Gröbner :

$$x^2 + y^2 - 4, \quad xy - 1, \quad y^3 + x - 4y.$$

En particulier, on observe que le nombre de polynômes dépend de l'ordre monomial.



2. Applications

2.1. Division et forme normale.

Définition 4. On dit qu'un polynôme $f \in \mathbb{A}$ est réduit par rapport à une partie $G \subset \mathbb{A}$ si aucun des monômes de f n'est divisible par le monôme de tête d'un élément de G .

On dit qu'une base de Gröbner $G = \{g_1, \dots, g_k\}$ est réduite si pour tout $i \in \{1, \dots, k\}$, g_i est réduit par rapport à $G \setminus \{g_i\}$.

Théorème 15 (Division). Soit G une base de Gröbner d'un idéal $I \subset \mathbb{A}$ et $F \in \mathbb{A}$. Il existe un unique couple $(B, R) \in \mathbb{A}^2$ tel que $F = B + R$, $B \in I$ et R est réduit par rapport à G . Dans ce cas, on note $R = \overline{F}^G$ et on dit que R est le reste de la division de F par G . De plus B appartient à l'idéal engendré par G .

DÉMONSTRATION. *Existence.* L'existence est donnée par l'algorithme de division ci-dessous :

Algorithme 1 Algorithme de Division

Entrées: $F, G = \{g_1, \dots, g_k\}$ et l'ordre monomial correspondant

Sorties: R et a_1, \dots, a_k tels que $F = a_1g_1 + \dots + a_kg_k + R$

- 1: Initialisation : $R = a_1 = \dots = a_k = 0$; $f = F$;
 - 2: **tant que** $f \neq 0$ **faire**
 - 3: $S := \{i \mid \text{LT}(g_i) \mid \text{LT}(f)\}$;
 - 4: **si** $S = \emptyset$ **alors**
 - 5: $r := r + \text{LT}(f)$; $f := f - \text{LT}(f)$;
 - 6: **sinon**
 - 7: $i := \min S$; $a_i := a_i + \text{LT}(f)/\text{LT}(g_i)$; $f := f - g_i \text{LT}(f)/\text{LT}(g_i)$;
 - 8: **fin si**
 - 9: **fin tant que**
 - 10: **renvoyer** (r, a_1, \dots, a_k) .
-

À chaque étape, la relation

$$(1) \quad F = f + r + a_1g_1 + \dots + a_kg_k$$

est maintenue ; par construction seuls des monômes réduits sont ajoutés à r ; enfin, la terminaison est assurée par la décroissance du terme de tête de f à chaque passage dans la boucle.

Unicité. Ecrivons $F = B_1 + R_1 = B_2 + R_2$. On en déduit $R_1 - R_2 = B_2 - B_1 \in I$. Donc $\text{LT}(R_1 - R_2) \in \langle \text{LT}(I) \rangle = \langle \text{LT}(G) \rangle$. Mais $\text{LT}(R_1 - R_2)$ est réduit, donc il n'est divisible par le monôme de tête d'un élément de G que s'il est nul. \square

Corollaire 1. *Sous les mêmes hypothèses, $F \in I$ si et seulement si $\overline{F}^G = 0$ ce qui fournit un test d'appartenance à un idéal dès que l'on possède une base de Gröbner.*

Corollaire 2. *Une base de Gröbner d'un idéal I engendre I .*

En effet il est clair que $\langle G \rangle \subset I$. Inversement, si $F \in I$, alors $F = F + 0$ est l'unique décomposition. D'après le théorème précédent, on a alors $F \in \langle G \rangle$.

Remarque 1. Lorsque $n = 1$, cette réduction correspond exactement à la division euclidienne : si G est une base de Gröbner d'un idéal I , G contient le pgcd g des éléments de G (et si G est réduite alors G ne contient qu'un élément). Si $F \in \mathbb{K}[x]$ alors \overline{F}^G est le reste de la division euclidienne de F par g .

Remarque 2. Dans la preuve du théorème précédent, nous n'avons pas utilisé l'hypothèse que G était une base de Gröbner pour montrer l'existence de la division. Ainsi, si G est une partie quelconque de \mathbb{A} et $F \in \mathbb{A}$, on peut définir le reste de la division de F par G . On notera toujours \overline{F}^G ce reste.

En revanche pour l'unicité nous avons utilisé l'hypothèse. On peut même montrer la réciproque : si $G \subset I$ est une partie finie telle que pour tout $F \in \mathbb{A}$ il existe un unique R réduit par rapport à G tel que $F - R \in I$, alors G est une base de Gröbner. En effet par unicité si $F \in I$ alors le reste de la division de F par G est nul et donc $F = \sum h_i g_i$. Ainsi le terme de tête de F est divisible par le terme de tête d'un élément de G . On a donc montré que $\langle \text{LT}(I) \rangle \subset \langle \text{LT}(G) \rangle$, l'inclusion inverse découle de $G \subset I$.

Remarque 3. L'algorithme de division permet aussi de réduire les bases de Gröbner. Si G n'est pas réduite, alors on réduit chaque g_i par $G \setminus \{g_i\}$, on le supprime si le reste est nul et on le remplace par son reste sinon. Le résultat engendre le même idéal, et ne modifie pas l'ensemble des termes de tête de G .

Remarque 4. Deux bases de Gröbner réduites pour le même ordre monomial sont identiques à des facteurs constants près. En effet, soient G, G' deux bases de Gröbner réduites. Si $g_1 \in G$ alors $\text{LT}(g_1)$ est divisible par le terme de tête d'un élément de G' , disons g'_1 . À son tour, le terme dominant de g'_1 est divisible par le terme dominant d'un élément $g_2 \in G$. Mais alors $\text{LT}(g_2)$ divise $\text{LT}(g_1)$ et comme G_1 est réduite $g_1 = g_2$. Ainsi $\text{LT}(g_1) = c \text{LT}(g'_1)$ pour un $c \in \mathbb{K}$. Posons alors $f_1 = g_1 - c g'_1 \in I$. Si $f_1 \neq 0$, son monôme de tête apparaît alors dans g_1 ou g'_1 , disons g_1 . Il n'est pas divisible par $\text{LT}(g_1)$ puisque $\text{LT}(f_1) \prec \text{LT}(g_1)$, ni divisible par $\text{LT}(g)$ pour $g \in G \setminus \{g_1\}$ puisque G est réduite. Ceci contredit $f_1 \in I$. Ainsi $g_1 = c g'_1$. En raisonnant de même pour tous les éléments de G , puis par symétrie, on obtient la conclusion. On obtient même l'unicité en forçant les termes de tête à être unitaires.

Remarque 5. Cette observation fournit un test d'égalité entre idéaux : si I_1 et I_2 sont deux idéaux possédant chacune une base de Gröbner alors $I_1 = I_2$ si et seulement si les bases de Gröbner réduites à coefficient de tête unitaires sont égales.

Remarque 6. Nous n'avons toujours pas démontré l'existence de bases de Gröbner.

2.2. Elimination.

Théorème 16 (Elimination). *Soit G une base de Gröbner de $I \subset \mathbb{K}[x_1, \dots, x_n]$ pour l'ordre lexicographique. Alors $G \cap \mathbb{K}[x_q, \dots, x_n]$ est une base de Gröbner de $I \cap \mathbb{K}[x_q, \dots, x_n]$ pour l'ordre lexicographique.*

Remarque 7. Le théorème est valable pour d'autres ordres monomiaux appelés ordres d'élimination, qui séparent les groupes de variables (x_1, \dots, x_{q-1}) et (x_q, \dots, x_n) lexicographiquement, mais traitent comme l'ordre du degré les variables à l'intérieur de chaque groupe. En Maple, ces ordres sont notés lexdeg.

DÉMONSTRATION. Notons $\mathbb{A}_q = \mathbb{K}[x_q, \dots, x_n]$, $G_q = G \cap \mathbb{A}_q$ et $I_q = I \cap \mathbb{A}_q$. Comme $G_q \subset I \cap \mathbb{A}_q$, $\langle G_q \rangle \subset I_q$ ($\langle G_q \rangle$ désigne ici l'idéal de \mathbb{A}_q engendré par G_q).

Réciproquement, si $F \in I_q \subset I$ alors $\text{LT}(F) \in \mathbb{A}_q$. En appliquant l'algorithme de division à F et G , si $g \in G$ est tel que $\text{LT}(g) \mid \text{LT}(F)$ alors $\text{LT}(g) \in \mathbb{A}_q$ et par définition de l'ordre lexicographique, g lui-même appartient alors à \mathbb{A}_q . Donc $g \in G_q$, et l'opération de soustraction maintient f dans \mathbb{A}_q . Ainsi à chaque étape de l'algorithme, tous les polynômes de l'écriture (1) sont dans \mathbb{A}_q et on obtient donc $F \in \langle G_q \rangle$. \square

L'élimination a de nombreuses applications. En voici quelques unes.

Résultant. Soient $f, g \in \mathbb{A} := \mathbb{K}[X_1, \dots, X_n, Y]$ deux polynômes. On muni \mathbb{A} de l'ordre lexicographique. Soit I l'idéal engendré par f et g et G la base de Gröbner réduite de I . Alors $G \cap \mathbb{K}[X_1, \dots, X_n]$ ne contient qu'un élément : le résultant de f et g par rapport à Y .

Implication. Soit un système polynomial

$$\begin{cases} x_1 = f_1(U_1, \dots, U_k) \\ \vdots \\ x_n = f_n(U_1, \dots, U_k) \end{cases}$$

et soit $I \subset \mathbb{K}[U_1, \dots, U_k, x_1, \dots, x_n]$ l'idéal engendré par ce système. L'élimination des U_i dans la base de Gröbner de I donne les équations implicites de la variété définie par le système. Si les f_i sont des fractions rationnelles, $f_i = p_i/q_i$, alors on travaille avec l'idéal

$$\langle q_1 x_1 - p_1, \dots, q_n x_n - p_n, 1 - t q_1 \cdots q_n \rangle \subset \mathbb{K}[t, U_1, \dots, U_k, x_1, \dots, x_n].$$

Relations de dépendance. Soient $f_1, \dots, f_m \in \mathbb{K}[x_1, \dots, x_n]$ des polynômes et $g \in \langle f_1, \dots, f_m \rangle$. L'élimination de t dans l'idéal $I = \langle y_1 - t f_1, \dots, y_m - t f_m, y - t g \rangle \subset \mathbb{K}[t, x_1, \dots, x_n, y_1, \dots, y_m, y]$ permet de calculer une relation de dépendance, c'est-à-dire des $h_i \in \mathbb{K}[x_1, \dots, x_n]$ tels que $g = \sum h_i f_i$.

Bases de Gröbner et coloriage de graphes



5			8		4
	9			5	7
4	7	1			
			3		4
			6		7
9	8			3	6
	9		8		
	4		7	5	
		3	4	1	6
				9	8

Un graphe est un nombre fini \mathcal{S} de sommets et \mathcal{A} d'arêtes, qui sont des paires de sommets. Deux sommets sont dits voisins s'ils sont reliés par une arête. Le *coloriage* d'un graphe avec m couleurs consiste à associer une couleur à chaque sommet du graphe de sorte que deux sommets voisins ne soient pas de la même couleur.

Le calcul peut être effectué par base de Gröbner. Pour forcer une variable X à ne pouvoir prendre que des valeurs parmi un ensemble fixé $\mathcal{C} := \{c_1, \dots, c_m\}$, il suffit de mettre dans un idéal le polynôme

$$P(X) = (X - c_1) \cdots (X - c_m).$$

Pour que deux variables X et Y ne puissent prendre que des valeurs différentes dans \mathcal{C} , on mettra

$$P(X), \quad P(Y), \quad Q(X, Y) := \frac{P(X) - P(Y)}{X - Y},$$

le dernier étant bien un polynôme. Ainsi, un problème de coloriage se code dans l'idéal engendré par

$$\{P(s) \mid s \in \mathcal{S}\} \cup \{Q(a) \mid a \in \mathcal{A}\}.$$

1. Écrire une procédure prenant en entrée une variable X et un entier m et renvoyant le polynôme $P(X)$. On pourra choisir $c_i = i$, $i = 1, \dots, m$.
2. Écrire une procédure prenant en arguments X, Y et m et renvoyant $Q(X, Y)$.
3. Saisir les arêtes du graphe de voisinage des pays d'Amérique du Sud (ou récupérer les quelques lignes de Maple sur la page du cours);
4. Prouver par un calcul de base de Gröbner que la carte d'Amérique du Sud n'est pas coloriable avec 3 couleurs, mais qu'elle l'est avec 4. Calculer une solution possible.

Pour le Sudoku, les couleurs sont les chiffres de 1 à 9, les sommets du graphe sont les 81 cases d'une grille 9×9 , et sont voisins deux sommets qui sont soit sur la même ligne, soit sur la même colonne, soit dans la même sous-grille de taille 3×3 .

5. Procéder comme pour le coloriage de carte. (La grille peut être récupérée en Maple sur la page du cours).
6. Cette grille n'a qu'une solution. Vérifier qu'il est encore possible de résoudre en enlevant l'information sur la case en haut à gauche, et compter le nombre de solutions correspondantes.

Bases de Gröbner II : Calcul et Géométrie

Résumé

L'une des applications principales des bases de Gröbner est en relation avec la géométrie, et l'appartenance au radical d'un idéal. Quant au calcul de ces bases, il est permis par un algorithme simple de Buchberger, dont la correction et la terminaison nécessitent un peu de travail.

1. Radicaux et Nullstellensatz

Dans tout ce qui suit, \mathbb{A} est un anneau commutatif unitaire et \mathbb{K} est un corps.

Définition 1. Soit \mathcal{I} un idéal de \mathbb{A} . Son radical :

$$\sqrt{\mathcal{I}} := \{f \in \mathbb{A} \mid \exists p \in \mathbb{N}, f^p \in \mathcal{I}\}$$

est un idéal de \mathbb{A} .

Cette définition vise à éliminer les multiplicités dans les polynômes sur lesquels on travaille. Par exemple, si $\mathbb{A} = \mathbb{K}[X]$ et $\mathcal{I} = \langle X^2 \rangle$, alors $\sqrt{\mathcal{I}} = \langle X \rangle$.

On se place maintenant définitivement dans le cas $\mathbb{A} = \mathbb{K}[X_1, \dots, X_n]$, cadre des bases de Gröbner. On notera \mathbf{X} pour X_1, \dots, X_n .

Proposition 1 (Astuce de Rabinowitsch, 1929). Soient f, f_1, \dots, f_r des éléments de $\mathbb{K}[\mathbf{X}]$. On définit aussi :

$$\mathcal{I} := \langle f_1, \dots, f_r \rangle \subset \mathbb{K}[\mathbf{X}]$$

$$\tilde{\mathcal{I}} := \langle f_1, \dots, f_r, 1 - tf \rangle \subset \mathbb{K}[\mathbf{X}, t]$$

Alors :

$$f \in \sqrt{\mathcal{I}} \Leftrightarrow \tilde{\mathcal{I}} = \langle 1 \rangle.$$

Ce résultat donne un algorithme pour le test d'appartenance au radical : le calcul d'une base de Gröbner de $\tilde{\mathcal{I}}$.

DÉMONSTRATION. \Rightarrow Supposons que $f^p \in \mathcal{I}$. Alors $f^p \in \tilde{\mathcal{I}}$ donc $t^p f^p \in \tilde{\mathcal{I}}$. Or $1 - t^p f^p \in \tilde{\mathcal{I}}$ donc $1 \in \tilde{\mathcal{I}}$.

\Leftarrow Si $1 \in \tilde{\mathcal{I}}$, alors il s'écrit

$$1 = g_1(\mathbf{X}, t)f_1(\mathbf{X}) + \dots + g_r(\mathbf{X}, t)f_r(\mathbf{X}) + g(\mathbf{X}, t)(1 - tf(\mathbf{X})).$$

En injectant $t = \frac{1}{f(\mathbf{X})}$, et en réduisant au même dénominateur, on obtient explicitement la décomposition de f^m en termes des f_i , où m est le maximum des degrés des g_i en t , ce qui montre $f \in \sqrt{\mathcal{I}}$. \square

On arrive au théorème fondamental de cette partie.

Théorème 17 (Nullstellensatz de Hilbert). *On suppose que \mathbb{K} est algébriquement clos. Soient f, f_1, \dots, f_r des éléments de $\mathbb{K}[\mathbf{X}]$. Alors $f \in \sqrt{\langle f_1, \dots, f_r \rangle}$ si et seulement si f s'annule sur le lieu des zéros communs des f_i dans \mathbb{K}^n .*

Ainsi, la géométrie de la variété définie par les f_i est codée dans le radical de l'idéal qu'ils engendrent.

DÉMONSTRATION. $\boxed{\Rightarrow}$ Si $f^p = \sum g_i f_i$, et si $\mathbf{x} \in \mathbb{K}^n$ est un zéro commun aux f_i , alors $f^p(\mathbf{x})$ donc $f(\mathbf{x}) = 0$.

$\boxed{\Leftarrow}$ Soit f s'annule en tous les zéros communs des f_i . D'après l'astuce de Rabinowitsch, il suffit de montrer que 1 est dans l'idéal engendré par $f_1, \dots, f_r, 1 - tf$. Tout d'abord, on observe que ces polynômes n'ont pas de zéros communs. En effet, s'il en existait un, disons $\mathbf{a} = (a_1, \dots, a_n)$, alors par hypothèse $f(\mathbf{a}) = 0$, donc $(1 - tf)(\mathbf{a}) = 1$, ce qui est contradictoire. La conclusion découle alors de la forme faible du Nullstellensatz ci-dessous. \square

Proposition 2 (Nullstellensatz faible). *Si \mathbb{K} est algébriquement clos, et \mathcal{I} est un idéal strict de $\mathbb{K}[\mathbf{X}]$, alors $\exists \mathbf{a} = (a_1, \dots, a_n) \in \mathbb{K}^n$ tel que*

$$\forall f \in \mathcal{I}, \quad f(\mathbf{a}) = 0.$$

La preuve procède par récurrence, et pour $n \geq 2$ utilise un argument de projection pour se ramener à une variable de moins. Le lemme suivant permet d'assurer que cette projection se passe bien. (Notons que si \mathbb{K} est algébriquement clos, il est infini.)

Lemme 1 (Lemme de normalisation de Noether). *Supposons $n \geq 2$, et \mathbb{K} infini. Soit $f \in \mathbb{K}[\mathbf{X}]$ de degré $d > 0$. Alors il existe $(\lambda_1, \dots, \lambda_{n-1}) \in \mathbb{K}^{n-1}$ tels que le coefficient de X_n^d dans*

$$f(X_1 + \lambda_1 X_n, \dots, X_{n-1} + \lambda_{n-1} X_n, X_n)$$

soit non nul.

DÉMONSTRATION. Posons

$$f(\mathbf{X}) = \sum_{i_1 + \dots + i_n \leq d} a_{i_1, \dots, i_n} X_1^{i_1} \dots X_n^{i_n}$$

Puisque f est de degré d , le polynôme

$$g(\mathbf{X}) := \sum_{i_1 + \dots + i_n = d} a_{i_1, \dots, i_n} X_1^{i_1} \dots X_n^{i_n}$$

est non nul.

Or le coefficient de X_n^d dans $f(X_1 + \lambda_1 X_n, \dots, X_{n-1} + \lambda_{n-1} X_n, X_n)$ vaut précisément

$$\sum_{i_1 + \dots + i_n = d} a_{i_1, \dots, i_n} \lambda_1^{i_1} \dots \lambda_{n-1}^{i_{n-1}}$$

Comme \mathbb{K} est infini, on est assuré de l'existence de scalaires $\lambda_1, \dots, \lambda_{n-1}$ qui lui donnent une valeur non nulle (encore par récurrence sur le nombre de variables : pour une variable, le nombre de racines est borné par le degré, ensuite on regarde le coefficient de tête, on prend un point où il ne s'annule pas et on conclut sur le polynôme en une variable ainsi obtenu). \square

DÉMONSTRATION (DU NULLSTELLENSATZ FAIBLE). On procède par récurrence sur n .

Le cas $n = 1$ est assuré par le fait que \mathbb{K} est algébriquement clos.

Si $n \geq 2$, on va projeter pour se ramener à une variable de moins. Soit $g \in \mathcal{I}$. D'après le lemme de normalisation, on peut supposer quitte à renormaliser g qu'il existe $\lambda_1, \dots, \lambda_{n-1}$ tels que $g(X_1 + \lambda_1 X_n, \dots, X_{n-1} + \lambda_{n-1} X_n, X_n) = X_n^d + r(X_1, \dots, X_n)$, r étant de degré au plus $n - 1$ en X_n .

Alors en posant $\mathcal{J} := \{f(X_1 + \lambda_1 X_n, \dots, X_{n-1} + \lambda_{n-1} X_n, X_n), f \in \mathcal{I}\}$, on obtient un autre idéal strict de $\mathbb{K}[\mathbf{X}]$. Donc, quitte à remplacer \mathcal{I} par \mathcal{J} , on suppose que $g(\mathbf{X}) = X_n^d + g_{d-1} X_n^{d-1} + \dots + g_0$, $g_i \in \mathbb{K}[X_1, \dots, X_{n-1}]$.

On pose alors $\mathcal{I}' = \mathcal{I} \cap \mathbb{K}[X_1, \dots, X_{n-1}]$. C'est un idéal strict de $\mathbb{K}[X_1, \dots, X_{n-1}]$. Par hypothèse de récurrence, il existe donc $\mathbf{a} = a_1, \dots, a_{n-1}$ qui annule tous les polynômes de \mathcal{I}' .

On pose alors $\mathcal{J} = \{f(\mathbf{a}, X_n), f \in \mathcal{I}\}$. C'est un idéal de $\mathbb{K}[X_n]$. S'il est strict alors il est engendré par un polynôme non constant en X_n , dont n'importe quelle racine a_n donne la réponse (\mathbf{a}, a_n) à la question. Si on suppose le contraire, alors $\exists f \in \mathcal{I}$ tel que $f(\mathbf{a}, X_n) = 1$. Ce polynôme peut s'écrire

$$f(\mathbf{X}) = f_0 + f_1 X_n + \dots + f_k X_n^k,$$

où les f_i sont dans $\mathbb{K}[X_1, \dots, X_{n-1}]$. L'hypothèse implique que $f_0(\mathbf{a}) = 1$ et $f_1(\mathbf{a}) = \dots = f_k(\mathbf{a}) = 0$.

Le résultant R de f et g par rapport à la variable X_n est à la fois un élément de $\mathbb{K}[X_1, \dots, X_{n-1}]$ et de $\langle f, g \rangle \subset \mathcal{I}$. Donc $R \in \mathcal{I}'$, ce qui implique $R(\mathbf{a}) = 0$. Pourtant, en évaluant la matrice de Sylvester de (f, g) en \mathbf{a} , on obtient des 1 sur la diagonale et des 0 au-dessus, ce qui entraîne $R(\mathbf{a}) = 1$, une contradiction. Donc \mathcal{J} est strict et la preuve est terminée. \square

2. Calcul effectif des bases de Gröbner

Nous allons maintenant voir comment mettre en place le calcul des bases de Gröbner, et enfin prouver leur existence.

2.1. S-polynômes. On se place dans $\mathbb{K}[X_1, \dots, X_n]$, et on fixe un ordre monomial.

Définition 2. Soient f et g deux polynômes. On appelle S-polynôme de f et g et on note $S(f, g)$ le polynôme suivant :

$$S(f, g) := (\text{LM}(f) \vee \text{LM}(g)) \left(\frac{f}{\text{LT}(f)} - \frac{g}{\text{LT}(g)} \right),$$

où l'on note $f \vee g$ le ppcm de f et g .

Par construction, $S(f, g)$ est un polynôme, et il appartient à $\langle f, g \rangle$. La notation S vient du mot *syzygie*, qui vient lui-même du grec ancien *suzugia* qui signifie « sous le même joug ».

La proposition suivante donne la clé de la construction de bases de Gröbner.

Proposition 3. L'ensemble $G = \{g_1, \dots, g_m\}$ est une base de Gröbner de $\langle G \rangle$ si et seulement si

$$\forall 1 \leq i < j \leq m, \quad \overline{S(g_i, g_j)}^G = 0.$$

DÉMONSTRATION. Le sens direct est clair.

Soit $f \in \mathcal{I} := \langle G \rangle$. Il s'agit de montrer que $\text{LT}(f) \in \langle \text{LT}(G) \rangle$. Parmi toutes les décompositions

$$f = \sum h_i g_i,$$

on en choisit une qui minimise la quantité

$$\delta := \max_i \text{LM}(h_i g_i),$$

ce qui est possible puisqu'il n'y a pas de chaîne infinie décroissante. Il suffit de montrer que $\text{LM}(f) = \delta$. Soit $S := \{i \mid \text{LM}(h_i g_i) = \delta\}$ et quitte à renuméroter les h_i et les g_i , on peut supposer $S = \{1, \dots, k\}$ pour un certain k . La décomposition de f se récrit

$$\begin{aligned} f &= \sum_{i \in S} h_i g_i + \sum_{i \notin S} h_i g_i \\ &= \sum_{i \in S} \text{LT}(h_i) g_i + \underbrace{\sum_{i \in S} (h_i - \text{LT}(h_i)) g_i + \sum_{i \notin S} h_i g_i}_{\text{chaque terme a un LM} < \delta} \end{aligned}$$

On note $\text{LT}(h_i) = c_i \mathbf{X}^{\alpha_i} = c_i X_1^{\alpha_{i1}} \cdots X_n^{\alpha_{in}}$ et on observe que $S(\mathbf{X}^{\alpha_i} g_i, \mathbf{X}^{\alpha_j} g_j)$ est un multiple de $S(g_i, g_j)$:

$$S(\mathbf{X}^{\alpha_i} g_i, \mathbf{X}^{\alpha_j} g_j) = \delta \left(\frac{\mathbf{X}^{\alpha_i} g_i}{\text{LC}(g_i) \delta} - \frac{\mathbf{X}^{\alpha_j} g_j}{\text{LC}(g_j) \delta} \right) = \frac{\mathbf{X}^{\alpha_i} g_i}{\text{LC}(g_i)} - \frac{\mathbf{X}^{\alpha_j} g_j}{\text{LC}(g_j)}.$$

Maintenant,

$$\begin{aligned} \sum_{i=1}^k c_i \mathbf{X}^{\alpha_i} g_i &= c_1 \text{LC}(g_1) S(\mathbf{X}^{\alpha_1} g_1, \mathbf{X}^{\alpha_2} g_2) \\ &\quad + (c_1 \text{LC}(g_1) + c_2 \text{LC}(g_2)) S(\mathbf{X}^{\alpha_3} g_3, \mathbf{X}^{\alpha_3} g_3) + \cdots \\ &\quad + (c_1 \text{LC}(g_1) + \cdots + c_{k-1} \text{LC}(g_{k-1})) S(\mathbf{X}^{\alpha_{k-1}} g_{k-1}, \mathbf{X}^{\alpha_k} g_k) \\ &\quad + (c_1 \text{LC}(g_1) + \cdots + c_k \text{LC}(g_k)) \frac{\mathbf{X}^{\alpha_k} g_k}{\text{LC}(g_k)}. \end{aligned}$$

Tous les termes de la somme sauf le dernier ont, par construction des S-polynômes, un monôme dominant strictement inférieur à δ et par hypothèse peuvent être récrits par l'algorithme de division comme combinaison des g_i , dont le terme de tête n'atteint pas δ . Par minimalité de δ , le dernier terme doit donc avoir pour monôme δ et on a récrit f comme une somme avec un seul monôme égal à δ et tous les autres inférieurs, donc $\text{LM}(f) = \delta$. \square

2.2. L'algorithme de Buchberger. L'algorithme de Buchberger est donné en figure. Il se base sur des calcul de S-polynômes que l'on réduit puis que l'on ajoute à la base que l'on a déjà.

PREUVE DE L'ALGORITHME. À chaque étape de l'algorithme, l'idéal engendré par G est $\langle f_1, \dots, f_m \rangle$. L'inclusion provient de l'initialisation de G et ensuite G ne s'accroît que de S-polynômes d'éléments de G . Enfin, lorsque l'algorithme termine, tous les S-polynômes de G sont bien réduits à 0 par G , ce qui prouve qu'il s'agit d'une base de Gröbner. Le seul point délicat à prouver est donc la terminaison.

Algorithme 1 Algorithme de Buchberger**Entrées:** $f_1, \dots, f_m \in \mathbb{K}[\mathbf{X}]$, muni d'un ordre monomial.**Sorties:** Une base de Gröbner de l'idéal engendré par les f_i ,
pour l'ordre monomial donné. $G := \{f_1, \dots, f_m\}$ $S := \{S(f_i, f_j), i < j\}$ **tant que** $S \neq \emptyset$ **faire** Choisir un $p \in S$ $S := S \setminus \{p\}$ $g := \bar{p}^G$ **si** $g \neq 0$ **alors** $S := S \cup \{S(g, h), h \in G\}$ $G := G \cup \{g\}$ **fin si****fin tant que****renvoyer** G

À chaque étape, soit le cardinal de S décroît, $\langle \text{LT}(G) \rangle$ croît. Il suffit de montrer que la deuxième possibilité ne peut se produire qu'à un nombre fini d'étapes. L'union de tous ces idéaux $\langle \text{LT}(G) \rangle$ est un idéal. Le résultat est alors une conséquence du lemme de Dickson ci-dessous. \square

Lemme 2 (Lemme de Dickson). *Soit A un ensemble de multi-indices en n variables. Alors tout idéal monomial $\mathcal{I} = \langle \mathbf{X}^\alpha, \alpha \in A \rangle$ admet une base monomiale finie.*

DÉMONSTRATION. On procède par récurrence sur le nombre de variables. Pour $n = 1$, on a $\mathcal{I} = \langle X^\beta \rangle$ où $\beta = \min A$.

Supposons $n > 1$. On note $\mathbf{X} = X_1, \dots, X_{n-1}$ et $Y = X_n$. Posons

$$\mathcal{J} := \{\mathbf{X}^\alpha \mid \exists m, \mathbf{X}^\alpha Y^m \in \mathcal{I}\}$$

Où α est un multi-indice en $n - 1$ variables.

L'idéal $\langle \mathcal{J} \rangle$ est un idéal monomial de $\mathbb{K}[\mathbf{X}]$. Il admet donc une base finie notée $\mathbf{X}^{\alpha_1}, \dots, \mathbf{X}^{\alpha_s}$. Posons alors :

$$m_i := \min \{m \in \mathbb{N} \mid \mathbf{X}^{\alpha_i} Y^m \in \mathcal{I}\}, \quad m := \max_i m_i.$$

Ensuite, pour $k = 0, \dots, m - 1$, on considère les « tranches »

$$\mathcal{J}_k := \{\mathbf{X}^\alpha \mid \mathbf{X}^\alpha Y^k \in \mathcal{I}\}.$$

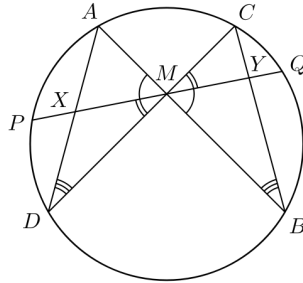
Chaque $\langle \mathcal{J}_k \rangle$ est un idéal monomial de $\mathbb{K}[\mathbf{X}]$, et admet par hypothèse de récurrence une base finie $\mathbf{X}^{\alpha_1^{(k)}}, \dots, \mathbf{X}^{\alpha_{s_k}^{(k)}}$.

Une base finie de \mathcal{I} est donc finalement donnée par

$$\{\mathbf{X}^{\alpha_j^{(k)}} Y^k \mid 0 \leq k < m, 1 \leq j \leq s_k\} \cup \{\mathbf{X}^{\alpha_1} Y^m, \dots, \mathbf{X}^{\alpha_s} Y^m\}.$$

 \square

Bases de Gröbner pour la géométrie



Le théorème du papillon est un exercice classique de géométrie Euclidienne. Soit M le milieu d'une corde PQ d'un cercle et soient AB et CD deux autres cordes passant par M ; AD et BC coupent PQ en X et Y respectivement. Il s'agit de montrer que M est aussi le milieu de XY .

Le but de l'exercice est d'utiliser les bases de Gröbner pour parvenir à la preuve sans trop de considérations géométriques. Sans perte de généralité, on pourra choisir de centrer le cercle en 0 , de lui donner un rayon 1 , de donner à M une abscisse nulle et d'imposer à PQ d'être horizontale. Il s'agit donc de construire un idéal de $\mathbb{K}_{\text{pol}} := \mathbf{Q}[x_A, y_A, x_B, y_B, x_C, y_C, x_D, y_D, x_P, y_P, x_Q, y_Q, x_X, y_X, x_Y, y_Y, x_M, y_M]$ codant la géométrie du problème, puis de tester si le polynôme $x_X + x_Y$ y appartient (ou en toute rigueur s'il appartient à son radical).

1. Écrire une procédure prenant en argument un point et renvoyant un polynôme exprimant que ce point est sur le cercle, une autre prenant trois points et exprimant qu'ils sont alignés;
2. Former un système mettant en équations le problème à l'aide de ces deux procédures et calculer une base de Gröbner G_1 de ce système pour un ordre du degré lexicographique inverse;
3. Constaté que $x_X + x_Y$ n'appartient pas à l'idéal engendré par G_1 , ni même à son radical, et donc que le théorème n'est pas vrai en toute généralité.

Comme souvent en géométrie, l'existence de cas dégénérés rend la propriété fautive ou mal posée. Il est cependant possible de tester la validité *générique* de la propriété en plaçant les paramètres du problème dans le corps de base. Ici, il s'agit donc de calculer la base dans l'anneau

$$\mathbb{K}_{\text{rat}} = \mathbf{Q}(y_M, y_A, y_C)[x_A, x_B, y_B, x_C, x_D, y_D, x_P, y_P, x_Q, y_Q, x_X, y_X, x_Y, y_Y, x_M]$$

plutôt que dans \mathbb{K}_{pol} .

4. Calculer une base de Gröbner G_2 de ce système pour un ordre du degré lexicographique inverse dans \mathbb{K}_{rat} et vérifier l'appartenance de la condition $x_X + x_Y$ à l'idéal engendré par G_2 dans \mathbb{K}_{rat} .

On peut ensuite identifier puis traiter séparément les cas de dégénérescence.

5. Factoriser les polynômes de G_1 de petit degré et en déduire des cas de dégénérescence à éviter. Poursuivre le calcul jusqu'à avoir identifié tous les cas dégénérés dans \mathbb{K}_{pol} .

Quatrième partie

Compléments

Systèmes linéaires et algorithme de Gauss-Jordan

Introduction

On fixe une bonne fois pour toutes un corps commutatif \mathbb{K} et deux entiers naturels non nuls m et n .

Le but de ce chapitre est l'étude d'un algorithme, dû à Carl Friedrich Gauss et amélioré par Wilhelm Jordan¹, permettant, étant donnée une matrice $A \in \mathcal{M}_{m,n}(\mathbb{K})$, de calculer efficacement diverses fonctions de A telles que son déterminant, son inverse, son rang, son noyau, son image, ... Cet algorithme nous permettra également de résoudre les systèmes linéaires de forme $Ax = b$, où $A \in \mathcal{GL}_n(\mathbb{K})$ est une matrice carrée inversible, et b , un vecteur de \mathbb{K}^n .

1. Formes réduites

Commençons par un peu de terminologie.

Soit $A \in \mathcal{M}_{m,n}(\mathbb{K})$. On dit que A est *échelonnée* si elle vérifie les trois conditions suivantes :

1. Toutes les lignes nulles de A sont situées en bas de A ,
2. Chaque ligne commence par strictement plus de zéros que la ligne située au-dessus d'elle,
3. Le premier élément non nul de toute ligne non nulle est égal à 1.

On dit que A est *réduite* si A est échelonnée et si de plus

4. Le premier élément non nul de toute ligne non nulle est le seul élément non nul de toute sa colonne.

Par exemple, la matrice

$$\begin{pmatrix} 0 & 1 & 4 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$

est *réduite*, tandis que la matrice

$$\begin{pmatrix} 1 & 3 & 1 & 0 \\ 0 & 1 & 0 & 2 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

1. respectivement 1777-1855 et 1842-1899. Il semble toutefois que cette méthode fût connue des Chinois dès le premier siècle de notre ère...

4. Exploitation de l'algorithme de Gauss-Jordan

Nous disposons donc d'une méthode efficace pour mettre une matrice sous une forme dont, nous l'avons vu, il est facile d'extraire des renseignements. Toutefois, afin d'obtenir ces renseignements non pas sur une forme simple de la matrice de départ, mais sur cette matrice elle-même, il est parfois nécessaire de « remonter » les opérations élémentaires ayant mené à cette forme simple.

Nous donnons ici des détails précis quant à la marche à suivre pour obtenir tel ou tel renseignement sur une matrice $A \in \mathcal{M}_{m,n}(\mathbb{K})$, en précisant à chaque fois (GJ) ou (G) afin d'indiquer si ladite méthode requiert l'emploi de l'algorithme de Gauss-Jordan, ou si l'algorithme de Gauss suffit.

Rang (G). Le rang de A , qui est invariant au fil de l'algorithme, puisque les **matrices d'opérations élémentaires** sont inversibles, est tout bonnement le nombre de lignes non nulles de toute **forme échelonnée** de A .

Déterminant (G). Le calcul du déterminant (dans le cas $n = m$ bien entendu) est à peine plus compliqué : Tout d'abord, le déterminant de A est bien sûr nul si A n'est pas inversible, autrement dit si les **formes échelonnées** de A contiennent des lignes nulles. Ensuite, si A est inversible, alors comme toutes les **formes échelonnées** de A sont triangulaires supérieures, de diagonales entièrement composées de 1, donc, puisque les déterminants des **transvections**, des **dilatations** de rapport λ , et des **échanges de lignes** sont respectivement 1, λ et -1 , le déterminant de A est le produit des pivots (c'est-à-dire des rapports des dilatations) réalisés par l'algorithme de Gauss, multiplié par un signe $(-1)^e$, où e est le nombre d'**échanges de lignes** utilisés par l'algorithme.

Systèmes linéaires (GJ). Toujours dans le cas $m = n$, A inversible, il est facile de résoudre le système linéaire $Ax = b$: En effet, lorsque A est inversible, les conditions 2, 3 et 4 montrent que la forme réduite de A n'est autre que la matrice-identité I_n . Ainsi, le produit de **matrices élémentaires** par lesquelles A est multipliée à gauche lors de l'algorithme de Gauss-Jordan est égal à l'inverse de A . On en déduit que si on applique Gauss-Jordan non pas à A , mais à la **matrice augmentée**

$$\left(\begin{array}{c|c} A & b \end{array} \right) \in \mathcal{M}_{n,n+1}(\mathbb{K}),$$

on obtient, en sus de la matrice identité, le vecteur inconnu x accolé à sa droite⁵ :

$$\left(\begin{array}{c|c} I_n & x \end{array} \right) \in \mathcal{M}_{n,n+1}(\mathbb{K}).$$

Il est facile d'imaginer une généralisation de ce principe permettant de résoudre *d'un seul coup* plusieurs systèmes linéaires, dès lors qu'ils ont la même matrice A (mais pas les mêmes b) : par exemple, pour résoudre

$$Ax_1 = b_1, \dots, Ax_r = b_r,$$

5. Si se restreint à l'algorithme de Gauss, on n'obtient pas I_n mais une matrice triangulaire supérieure, ce qui permet malgré tout de résoudre le système en cascade ; mais bien entendu, cette technique favorise la propagation des erreurs d'arrondi.

il suffit d'appliquer Gauss-Jordan à la matrice augmentée

$$\left(\begin{array}{c|ccc} A & b_1 & \cdots & b_r \end{array} \right) \in \mathcal{M}_{n,n+r}(\mathbb{K}),$$

ce qui donne

$$\left(\begin{array}{c|ccc} I_n & x_1 & \cdots & x_r \end{array} \right) \in \mathcal{M}_{n,n+r}(\mathbb{K}).$$

Inverse (GJ). En particulier, si on souhaite non pas résoudre des systèmes linéaires $Ax_i = b_i$, $i = 1, \dots, r$, mais inverser la matrice A , il suffit de prendre $r = n$, $b_i = e_i$, où $(e_i)_{1 \leq i \leq n}$ désigne la base canonique de \mathbb{K}^n . On applique donc Gauss-Jordan à la matrice augmentée

$$(A \mid I_n) = \left(\begin{array}{c|ccc} A & 1 & \cdots & 0 \\ & \vdots & \ddots & \vdots \\ & 0 & \cdots & 1 \end{array} \right) \in \mathcal{M}_{n,2n}(\mathbb{K}),$$

et le résultat est

$$(I_n \mid A^{-1}) = \left(\begin{array}{c|ccc} 1 & \cdots & 0 & \\ \vdots & \ddots & \vdots & \\ 0 & \cdots & 1 & \end{array} \middle| \begin{array}{c} A^{-1} \\ \\ \end{array} \right) \in \mathcal{M}_{n,2n}(\mathbb{K}).$$

Noyau (GJ). Soit $u \in \mathcal{L}(\mathbb{K}^n, \mathbb{K}^m)$ l'application linéaire dont la matrice dans les bases canoniques $(e_i)_{1 \leq i \leq n}$ de \mathbb{K}^n et $(f_j)_{1 \leq j \leq m}$ de \mathbb{K}^m est notre matrice $A \in \mathcal{M}_{m,n}(\mathbb{K})$. Nous savons que l'algorithme de Gauss-Jordan transforme la matrice A en une de ses **formes réduites**, qui est la matrice de u écrite dans une autre base de l'espace d'arrivée \mathbb{K}^m . Quitte à permuter ses colonnes en la multipliant à *droite* par une matrice de permutation $P \in \mathfrak{S}_n$, ce qui revient à réécrire une nouvelle fois la matrice de u après avoir permuté les vecteurs de la base $(e_i)_{1 \leq i \leq n}$ de l'espace de départ \mathbb{K}^n , on peut supposer d'après les conditions 1, 2, 3 et 4 que cette forme réduite est de la forme

$$\left(\begin{array}{c|ccc} I_r & A' & & \\ \hline 0 & \cdots & 0 & \end{array} \right) \in \mathcal{M}_{m,n}(\mathbb{K}),$$

où r désigne le rang de A . D'après le théorème du rang, le noyau de u , $\text{Ker } u$, est de dimension $n - r$; or on constate que les colonnes de la matrice

$$\left(\begin{array}{c} A' \\ -I_{n-r} \end{array} \right) = \left(\begin{array}{ccc} A' & & \\ -1 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & -1 \end{array} \right) \in \mathcal{M}_{n,n-r}(\mathbb{K})$$

sont les coordonnées dans la base permutée de $n - r$ vecteurs de l'espace de départ \mathbb{K}^n , linéairement indépendants à cause du bloc scalaire, et dont les images par u sont nulles ; par conséquent, ils forment donc une base du noyau $\text{Ker } u = \text{Ker } A$. Pour exprimer ces vecteurs dans la base canonique de \mathbb{K}^n , il suffit de permuter leurs coordonnées selon la permutation inverse P^{-1} .

En conclusion, en accolant un bloc scalaire $-I_{n-r}$ sous le bloc $A' \in \mathcal{M}_{r, n-r}(\mathbb{K})$ fourni par Gauss-Jordan, puis en permutant le cas échéant les lignes de la matrice obtenue suivant la permutation inverse P^{-1} , on obtient une base du noyau de A .

Remarquons au passage que cette méthode permet de calculer le sous-espace propre associé à une valeur propre donnée λ d'une matrice carrée $A \in \mathcal{M}_n(\mathbb{K})$: il suffit d'appliquer Gauss-Jordan à $A - \lambda I_n \in \mathcal{M}_n(\mathbb{K})$.

Image (GJ). Soit à nouveau $u \in \mathcal{L}(\mathbb{K}^n, \mathbb{K}^m)$ l'application linéaire dont la matrice dans les bases canoniques $(e_i)_{1 \leq i \leq n}$ de \mathbb{K}^n et $(f_j)_{1 \leq j \leq m}$ de \mathbb{K}^m est A . Les opérations élémentaires effectuées par l'algorithme de Gauss-Jordan correspondent à des changements de base dans l'espace d'arrivée \mathbb{K}^m , tels que la matrice de u dans la nouvelle base finalement obtenue $(f'_j)_{1 \leq j \leq m}$ soit une forme réduite de A . Quitte à permuter les vecteurs de la base $(e_i)_{1 \leq i \leq n}$ de l'espace de départ, ce qui ne change pas l'image, on peut supposer que cette forme réduite s'écrit

$$\left(\begin{array}{c|c} I_r & A' \\ \hline 0 & 0 \end{array} \right) = \left(\begin{array}{ccc|ccc} 1 & \cdots & 0 & & & \\ \vdots & \ddots & \vdots & & & \\ 0 & \cdots & 1 & & & \\ \hline 0 & \cdots & 0 & 0 & \cdots & 0 \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ 0 & \cdots & 0 & 0 & \cdots & 0 \end{array} \right) \in \mathcal{M}_{m,n}(\mathbb{K}),$$

où r désigne le rang de A . Il est alors clair que l'image $\text{Im } A$ de A , qui est celle par définition celle de u , admet pour base $(f'_j)_{1 \leq j \leq r}$; ainsi, tout le problème est d'exprimer les f'_j , $1 \leq j \leq r$, dans l'ancienne base $(f_j)_{1 \leq j \leq m}$. Pour ce faire, il suffit d'appliquer, dans l'ordre inverse, les opérations élémentaires inverses à celles ayant mené à la forme réduite de A à la matrice

$$\left(\begin{array}{c|c} I_r & \\ \hline 0 & 0 \end{array} \right) = \left(\begin{array}{ccc|ccc} 1 & \cdots & 0 & & & \\ \vdots & \ddots & \vdots & & & \\ 0 & \cdots & 1 & & & \\ \hline 0 & \cdots & 0 & 0 & \cdots & 0 \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ 0 & \cdots & 0 & 0 & \cdots & 0 \end{array} \right) \in \mathcal{M}_{m,n}(\mathbb{K}),$$

Les r colonnes de la matrice $m \times r$ obtenue forment alors une base de $\text{Im } A$.

5. L'algorithme de Wiedemann

L'algorithme de Gauss-Jordan est donc un méthode efficace pour inverser une matrice ou résoudre un système linéaire. On peut toutefois légitimement espérer mieux lorsque la matrice sur laquelle on travaille présente une forme très particulière, et notamment si elle est creuse, c'est-à-dire si presque tous ses coefficients sont nuls. Nous allons voir que dans ce cas, il existe effectivement une méthode plus

rapide pour inverser la matrice (et donc, si c'est ce qu'on veut, résoudre le système $Ax = b$) : l'algorithme de Wiedemann.

Étant donnée une matrice carrée $A \in \mathcal{M}_n(\mathbb{K})$, cet algorithme en calcule le *polynôme minimal*

$$\mu_A = X^d + m_{d-1}X^{d-1} + \cdots + m_1X + m_0 \in \mathbb{K}[X].$$

La connaissance de ce polynôme est en effet très précieuse : en effet, A est inversible si et seulement si le coefficient constant $m_0 \in \mathbb{K}$ n'est pas nul (sinon μ_A ne serait pas minimal), et alors dans ce cas

$$A^{-1} = \frac{-1}{m_0}(A^{d-1} + m_{d-1}A^{d-2} + \cdots + m_1)$$

peut s'évaluer par la méthode de Horner en d multiplications par A . Or, d'une part, si s désigne le nombre d'éléments non nuls de A , chacune de ses multiplications peut se faire en $\mathcal{O}(s)$; d'autre part, d'après le théorème de Cayley-Hamilton, $d \leq n$, si bien que le calcul de A^{-1} se fait ainsi en $\mathcal{O}(n^2 + ns)$, le n^2 provenant des additions de matrices et des multiplications par les scalaires m_i . Cet algorithme est donc très intéressant dès que A est suffisamment creuse pour que $s = \mathcal{O}(n)$.

L'idée pour trouver μ_A est la suivante : Étant donnés deux vecteurs-colonnes $u, v \in \mathbb{K}^n$, la suite numérique $(a_n = {}^t u A^n v)_{n \in \mathbb{N}} \in \mathbb{K}^{\mathbb{N}}$ vérifie une relation de récurrence linéaire à coefficients constants puisque, en convenant que $m_d = 1$,

$$\forall n \in \mathbb{N}, \quad 0 = {}^t u A^n \mu_A(A) v = {}^t u \sum_{i=0}^d m_i A^{n+i} v = \sum_{i=0}^d m_i a_{n+i},$$

et donc le polynôme caractéristique de cette récurrence divise toujours μ_A , et en fait, il se trouve même que, génériquement, ces deux polynômes coïncident ! Ainsi, le calcul de μ_A se ramène (à un nombre presque sûrement fini d'échecs près) à celui du polynôme minimal de la récurrence satisfaite par $(a_n)_{n \in \mathbb{N}}$, donc, grâce à l'algorithme de Berlekamp-Massey, au calcul de l'approximant de Padé (n, n) de la série formelle associée

$$\sum_{n=0}^{+\infty} a_n X^n \in \mathbb{K}[[X]],$$

ce que l'on sait faire en $\mathcal{O}(n^2)$ par l'algorithme d'Euclide étendu.

Au final, on réalise ainsi le calcul de A^{-1} en $\mathcal{O}(n^2 + ns)$.

Systèmes linéaires sur des anneaux euclidiens

Résumé

La méthode du pivot de Gauss permet de résoudre des systèmes linéaires à coefficients dans un corps (par exemple \mathbb{R}) en échelonnant une matrice. Le but de ce chapitre est de donner des résultats théoriques et des algorithmes lorsque les coefficients sont pris dans un anneau principal ou euclidien (par exemple \mathbb{Z}) : algorithme de Bareiss, formes normales de Hermite et de Smith.

1. Algorithme de Bareiss

On cherche à appliquer l'algorithme du pivot de Gauss à une matrice à coefficients dans \mathbb{Z} par exemple. Si A est une telle matrice, $A = (a_{i,j})_{1 \leq i,j \leq n}$, et on suppose que A est générique de sorte qu'elle est inversible et qu'il n'est pas nécessaire d'échanger des lignes dans l'algorithme du pivot de Gauss. Alors, si les $a_{i,j}$ ont au plus L chiffres, en notant $A^{(k)}$ la matrice obtenue après k itérations de l'algorithme de Gauss (sur \mathbf{Q}), on a :

$$a_{i,j}^{(k+1)} = \frac{a_{i,j}^{(k)} a_{k,k}^{(k)} - a_{k,j}^{(k)} a_{i,k}^{(k)}}{a_{k,k}^{(k)}}$$

et donc par une récurrence immédiate, les coefficients de $A^{(n)}$ sont des rationnels dont les numérateurs et dénominateurs ont au plus $2^n L$ chiffres, d'où une complexité en $O(n^3 M_B(2^n L))$ où $M_B(a)$ est la complexité de la multiplication binaire à a chiffres.

Dans la suite, on montrera qu'en modifiant très légèrement l'algorithme du pivot de Gauss, on peut tout calculer dans \mathbb{Z} , avec une complexité moindre.

Lemme 1 (Identité de Sylvester-Bareiss). Soit $A = \begin{pmatrix} M & z & t \\ x & a & b \\ y & c & d \end{pmatrix}$ une matrice à

coefficients dans un anneau commutatif, où M est une matrice carrée, z et t des vecteurs colonnes, x et y des vecteurs lignes et a, b, c, d des scalaires. Alors :

$$\det A \det M = \begin{vmatrix} \det \begin{pmatrix} M & z \\ x & a \end{pmatrix} & \det \begin{pmatrix} M & t \\ x & b \end{pmatrix} \\ \det \begin{pmatrix} M & z \\ y & c \end{pmatrix} & \det \begin{pmatrix} M & t \\ y & d \end{pmatrix} \end{vmatrix}$$

DÉMONSTRATION. Il suffit de prouver l'égalité dans l'anneau de polynômes sur \mathbb{Z} dont les variables sont les coefficients de A , puis d'évaluer dans l'anneau qui nous intéresse. La matrice M , vue comme matrice à coefficients dans $\mathbf{Q}[a_{i,j}]$ est inversible (son déterminant est non nul, car sinon toute matrice aurait un déterminant nul).

Par conséquent,

$$\begin{aligned} \begin{vmatrix} M & z \\ x & a \end{vmatrix} &= \det \left(\begin{pmatrix} M & 0 \\ x & 1 \end{pmatrix} \begin{pmatrix} I_{n-1} & M^{-1}z \\ 0 & a - xM^{-1}z \end{pmatrix} \right) \\ &= \det M(a - xM^{-1}z) \end{aligned}$$

De la même façon, puisque

$$A = \begin{pmatrix} M & 0 & 0 \\ x & 1 & 0 \\ y & 0 & 1 \end{pmatrix} \begin{pmatrix} I_{n-2} & M^{-1} \begin{pmatrix} z & t \end{pmatrix} \\ 0 & \begin{pmatrix} a & b \\ c & d \end{pmatrix} - \begin{pmatrix} x \\ y \end{pmatrix} M^{-1} \begin{pmatrix} z & t \end{pmatrix} \end{pmatrix}$$

On a :

$$\begin{aligned} \det A \det M &= (\det M)^2 \begin{vmatrix} a - xM^{-1}z & b - xM^{-1}t \\ c - yM^{-1}z & d - yM^{-1}t \end{vmatrix} \\ &= \begin{vmatrix} \det M(a - xM^{-1}z) & \det M(b - xM^{-1}t) \\ \det M(c - yM^{-1}z) & \det M(d - yM^{-1}t) \end{vmatrix} \\ &= \begin{vmatrix} \det \begin{pmatrix} M & z \\ x & a \end{pmatrix} & \det \begin{pmatrix} M & t \\ x & b \end{pmatrix} \\ \det \begin{pmatrix} M & z \\ y & c \end{pmatrix} & \det \begin{pmatrix} M & t \\ y & d \end{pmatrix} \end{vmatrix} \end{aligned}$$

□

Corollaire 1. Si $A = (a_{i,j})_{1 \leq i,j \leq n}$, soient, pour $1 \leq k \leq n$:

$$a_{i,j}^{[k]} = \begin{vmatrix} a_{1,1} & \dots & a_{1,k} & a_{1,j} \\ \vdots & & \vdots & \vdots \\ a_{k,1} & \dots & a_{k,k} & a_{k,j} \\ a_{i,1} & \dots & a_{i,k} & a_{i,j} \end{vmatrix}.$$

Alors $a_{i,j}^{[k]} a_{k-1,k-1}^{[k-2]} = a_{k,k}^{[k-1]} a_{i,j}^{[k-1]} - a_{i,k}^{[k-1]} a_{k,j}^{[k-1]}$ pour $2 \leq k \leq n$ et pour tous i, j .

DÉMONSTRATION. On applique le lemme à :

$$\begin{aligned} M &= \begin{pmatrix} a_{1,1} & \dots & a_{1,k-1} \\ \vdots & & \vdots \\ a_{k-1,1} & \dots & a_{k-1,k-1} \end{pmatrix}, \\ z &= \begin{pmatrix} a_{1,k} \\ \vdots \\ a_{k-1,k} \end{pmatrix}, \\ t &= \begin{pmatrix} a_{1,j} \\ \vdots \\ a_{k-1,j} \end{pmatrix}, \end{aligned}$$

$$x = (a_{k,1} \quad \dots \quad a_{k,k-1}),$$

$$y = (a_{i,1} \quad \dots \quad a_{i,k-1}),$$

et $a = a_{k,k}$, $b = a_{k,j}$, $c = a_{i,k}$, $d = a_{i,j}$.

□

Remarque 1. Noter la grande similitude avec la récurrence donnant les coefficients de la k -ième itérée dans l'algorithme de Gauss, à un petit décalage d'indice près, ce qui permet d'avoir des coefficients entiers.

Algorithme 1 Algorithme de Bareiss

Entrées: Une matrice $M \in \mathcal{M}_n(R)$ sur un anneau intègre R , dont les mineurs principaux sont non nuls.

Sorties: Une matrice $U \in \mathcal{M}_n(R)$ triangulaire supérieure telle qu'il existe $L \in$

$$\text{GL}_n(\text{Frac } R) \text{ avec } M = LU \text{ et } u_{k,k} = \det \begin{pmatrix} a_{1,1} & \dots & a_{1,k} \\ \vdots & & \vdots \\ a_{k,1} & \dots & a_{k,k} \end{pmatrix}.$$

$c := 1$

pour k de 1 à $n-1$ **faire**

pour i de $k+1$ à n **faire**

pour j de $k+1$ à n **faire**

$a[i,j] := (a[i,j]*a[k,k] - a[k,j]*a[i,k])/c$

fin pour

$a[i,k] := 0$

fin pour

$c := a[k,k]$

fin pour

Remarque 2. La condition sur les mineurs permet d'assurer qu'il n'y aura pas à échanger de lignes pour avoir un pivot non nul (sinon, il faudrait tester si $c = 0$).

DÉMONSTRATION. Grâce au corollaire, on a par récurrence sur k que la matrice calculée après la k -ième itération vaut :

$$M^{[k]} = \begin{pmatrix} a_{1,1} & a_{1,2} & \dots & \dots & \dots & \dots & a_{1,n} \\ 0 & a_{2,2}^{[1]} & \dots & \dots & \dots & \dots & a_{2,n}^{[1]} \\ \vdots & 0 & \ddots & & & & \\ \vdots & \vdots & & a_{k,k}^{[k-1]} & \dots & \dots & a_{k,n}^{[k-1]} \\ \vdots & \vdots & & 0 & a_{k+1,k+1}^{[k]} & \dots & a_{k+1,n}^{[k]} \\ \vdots & \vdots & & \vdots & \vdots & & \vdots \\ 0 & 0 & & 0 & a_{n,k+1}^{[k]} & \dots & a_{n,n}^{[k]} \end{pmatrix}$$

Si la matrice M a des coefficients à n chiffres, tous les coefficients calculés sont bornés par $n!(2^L)^n = O(n^n 2^{Ln})$ (d'après l'expression sous forme de déterminant donnée dans le corollaire), donc l'algorithme est en $O(n^3 M_B(n \log n + Ln))$, où $M_B(a)$ est la complexité de la multiplication binaire à a chiffres. \square

2. Formes normales de Hermite et de Smith

Définition 1. H est une forme de Hermite de $A \in \mathcal{M}_{m,n}(\mathbb{Z})$ si :

1. Il existe $U \in \text{GL}_n(\mathbb{Z})$ telle que $AU = H$.

2. H a une forme échelonnée par colonne, c'est-à-dire qu'il existe $r \geq 1$ tel que $\text{col}_1(H), \dots, \text{col}_r(H) \neq 0$, $\text{col}_s(H) = 0$ pour tout $s > r$, et il existe $1 \leq c(1) < c(2) < \dots < c(r) \leq n$ tels que $H_{c(i),i}$ soit le premier élément non nul de la colonne i .

Théorème 18 (Hermite). *Tout matrice sur un anneau principal admet une forme de Hermite.*

Remarque 3. Dans le cas de l'anneau \mathbb{Z} , si on impose $H_{c(i),i} > 0$ et $0 \leq H_{c(i),j} < H_{c(i),i}$ pour tout $j < i$, on a même unicité.

DÉMONSTRATION. Le cas crucial est celui de la dimension 2. Si $\begin{pmatrix} a & b \\ v & a_0 \end{pmatrix}$ est une matrice ligne, alors en notant $d = \text{pgcd}(a, b)$, on peut écrire $a = da_0$, $b = db_0$ et par le théorème de Bézout, $a_0u + b_0v = 1$. Par conséquent, $\begin{pmatrix} u & -b_0 \\ v & a_0 \end{pmatrix} \in \text{GL}_2(R)$ (et même $\text{SL}_2(R)$) et $\begin{pmatrix} a & b \\ v & a_0 \end{pmatrix} \begin{pmatrix} u & -b_0 \\ v & a_0 \end{pmatrix} = \begin{pmatrix} d & 0 \\ v & a_0 \end{pmatrix}$

En dimension n , si $(a_{1,1} \ \dots \ a_{1,n})$ est la première ligne de A , en la multipliant A à droite par une matrice de la forme $\begin{pmatrix} I_{n-2} & 0 & 0 \\ 0 & x & y \\ 0 & z & t \end{pmatrix}$ avec $(a_{1,n-1} \ a_{1,n}) \begin{pmatrix} x & y \\ z & t \end{pmatrix} = (\text{pgcd}(a_{1,n-1}, a_{1,n}) \ 0)$, on obtient une matrice dont la première ligne est

$$(a_{1,1} \ \dots \ a_{1,n-2} \ \text{pgcd}(a_{1,n-1}, a_{1,n}) \ 0)$$

Par récurrence, en multipliant A à droite par $n-1$ matrices dans $\text{SL}_n(R)$, on obtient une matrice dont la première ligne est $(\text{pgcd}(a_{1,1}, \dots, a_{1,n}) \ 0 \ \dots \ 0)$.

Toujours par récurrence, on applique ce procédé à la deuxième ligne (en ignorant le premier élément de la deuxième ligne, pour ne pas modifier la première ligne), etc.

On obtient ainsi une forme échelonnée. \square

Théorème 19. *Si R est euclidien, $A \in \mathcal{M}_{m,n}(R)$ peut être mise sous la forme de Hermite en utilisant uniquement des transformations élémentaires sur les colonnes.*

Remarque 4. Les transformations élémentaires sont les opérations sur les colonnes de la forme $c_i \leftrightarrow c_j$ ($i \neq j$), $c_i \leftarrow c_i + \alpha c_j$ ($i \neq j$ et $\alpha \in R$), et $c_i \leftarrow -c_i$.

DÉMONSTRATION. Il s'agit simplement d'adapter la preuve du cas principal en utilisant l'algorithme d'Euclide pour le calcul de coefficients de Bézout : partant du vecteur $\begin{pmatrix} a & b \end{pmatrix}$, si par exemple $v(b) \leq v(a)$ (où v est le stathme euclidien), la division euclidienne de a par b s'écrit $a = bq + r$ avec $v(r) < v(b)$ ou $r = 0$, donc en faisant l'opération élémentaire $c_1 \leftarrow c_1 - qc_2$, on obtient le vecteur $\begin{pmatrix} r & b \end{pmatrix}$. La terminaison de l'algorithme d'Euclide montre qu'on arrive finalement au vecteur $(\text{pgcd}(a, b) \ 0)$.

Le même argument que dans le cas principal permet de passer de la dimension 2 à la dimension n . \square

En faisant des opérations à la fois sur les lignes et les colonnes (donc en multipliant notre matrice à droite et à gauche par des matrices inversibles), on obtient une forme normale plus simple, dite forme de Smith :

Localisation de racines de polynômes

Résumé

Dans le chapitre qui suit, le problème étudié est celui de la localisation des racines d'un polynôme, qu'il soit réel ou complexe. Selon les deux cas possibles, les méthodes ne seront pas les mêmes. Dans le cas complexe (plus général en un sens), on donnera d'abord un algorithme performant pour calculer la valeur d'une racine, puis nous présenterons quelques résultats théoriques de localisation. En revanche, lorsque le polynôme sera supposé réel, l'accent sera porté sur le nombre de racines du polynôme dans un intervalle prescrit, problème que circonscrit très convenablement le théorème de Sturm.

1. La méthode de Newton

On se donne un polynôme $f \in \mathbb{C}[X]$, et on définit, pour x n'annulant pas f' , le nombre $N_f(x) = x - \frac{f(x)}{f'(x)}$. Modulo une hypothèse de stabilité, le résultat suivant, connu sous le nom de *méthode de Newton*, montre qu'il est facile d'approcher une racine simple de f grâce à N_f . Plus précisément :

Théorème 22. *Si ζ est une racine simple de f , et $r > 0$ est tel que*

$$r \cdot \sup_{|x-\zeta| \leq r} |f''(x)| \leq \frac{1}{2}|f'(\zeta)|,$$

alors pour tout x_0 tel que $|x_0 - \zeta| \leq r$, la suite $(x_k)_{k \geq 0}$ définie par l'itération $x_{k+1} = N_f(x_k)$ converge vers ζ avec vitesse quadratique :

$$|x_k - \zeta| \leq \left(\frac{1}{2}\right)^{2^k - 1} |x_0 - \zeta|.$$

DÉMONSTRATION. Il y a deux étapes :

1. Commençons par montrer que f' ne s'annule pas sur $D(\zeta, r)$, ce qui justifiera l'existence de la suite $(x_k)_{k \in \mathbb{N}}$. Pour cela, il suffit d'écrire, pour $x \in D(\zeta, r)$:

$$|f'(x) - f'(\zeta)| = \left| \int_{\zeta}^x f''(t) dt \right| \leq \frac{|x - \zeta|}{2r} |f'(\zeta)| \leq \frac{|f'(\zeta)|}{2}$$

et donc $|f'(x)| \geq \frac{|f'(\zeta)|}{2} > 0$ ce qui règle le premier point.

2. Quant à la convergence, on va procéder par récurrence sur k , l'initialisation étant bien sûr vérifiée. Alors, il s'agit seulement de bien exploiter la formule de Taylor (celle avec reste intégral). En effet, pour tout $x \in D(\zeta, r)$ on a

$$0 = f(\zeta) = f(x + (\zeta - x)) = f(x) + (\zeta - x)f'(x) + \int_x^\zeta (\zeta - t)f''(t)dt.$$

Cela se réécrit encore :

$$x - \frac{f(x)}{f'(x)} - \zeta = \frac{1}{f'(x)} \int_x^\zeta (\zeta - t)f''(t)dt$$

et donc

$$|N_f(x) - \zeta| \leq \frac{|f''(\zeta)|}{|f'(x)|} \frac{1}{2r} \frac{|\zeta - x|^2}{2} \leq \frac{1}{2r} |\zeta - x|^2.$$

Alors, on termine en écrivant :

$$|x_{k+1} - \zeta| \leq \frac{1}{2r} \left(\frac{1}{2}\right)^{2^{k+1}-2} \times 2|x_0 - \zeta|^2 \leq \left(\frac{1}{2}\right)^{2^{k+1}-1} |x_0 - \zeta|.$$

□

Cette méthode est donc performante (elle converge vite, et demande finalement assez peu de calculs), d'autant plus qu'elle est largement généralisable à des classes de fonctions bien plus larges que les polynômes. Cependant, un des inconvénients est qu'il faut trouver une borne *a priori* qui garantisse la validité des calculs. Cela requiert donc en particulier une localisation plus ou moins bonne de la racine étudiée ; c'est l'objet de la section qui suit.

2. Distance aux racines

2.1. Borne de Cauchy. Soit $P = X^n + a_{n-1}X^{n-1} + \dots + a_0 \in \mathbb{C}[X]$. Le résultat que l'on commence par présenter montre que pour majorer le module des racines de P , il suffit de se ramener à un polynôme réel :

Théorème 23. *Si au moins un des a_i est non-nul, alors toutes les racines de P sont majorées en module par l'unique racine strictement positive de*

$$Q(x) = x^n - \sum_{i=0}^{n-1} |a_i| x^i.$$

DÉMONSTRATION. Tout d'abord, l'écriture $\frac{Q(x)}{x^n} = 1 - f(x)$ où $f : \mathbb{R}_+^* \rightarrow \mathbb{R}$ est une fonction continue positive décroissant strictement de $+\infty$ à 0, montre que Q admet en effet une seule racine strictement positive $r > 0$.

L'inégalité triangulaire entraîne $|P(z) - z^n| \leq \sum_{i=0}^{n-1} |a_i| \cdot |z|^i$ pour tout $z \in \mathbb{C}$. Il s'ensuit que pour toute racine $\zeta \in \mathbb{C}^*$ de P on a $\sum_{i=0}^{n-1} |a_i| \cdot |\zeta|^i \geq |\zeta|^n$ et donc $Q(|\zeta|) \leq 0$, ou encore $f(|\zeta|) \geq 1 = f(r)$. Puisque f est strictement décroissante, cela permet de conclure que $|\zeta| \leq r$. □

Dans le cas où 0 n'est pas racine de P , on peut appliquer le résultat précédent au polynôme réciproque $X^n P(1/X)$ et obtenir ainsi une borne inférieure sur le module des racines de P .

Corollaire 1. *Si $P(0) \neq 0$, le module de la racine de P la plus proche de 0 est minoré par l'unique racine positive du polynôme $|a_0| - \sum_{i=1}^{n-1} |a_i| x^i - x^n$.*

Corollaire 2 (Borne de Cauchy). *Toutes les racines du polynôme $P(X) = X^n + a_{n-1}X^{n-1} + \dots + a_0 \in \mathbb{C}[X]$ sont contenues dans le disque centré à l'origine et de rayon $1 + \max_{1 \leq i \leq n} (|a_i|)$.*

DÉMONSTRATION. Soit A le maximum des modules des coefficients a_i . Pour $s > 1 + A$ on a les majorations successives

$$|a_0| + \dots + |a_{n-1}|s^{n-1} < 1 + A + sA + \dots + s^{n-1}A = 1 + A \cdot \frac{s^n - 1}{s - 1} < s^n.$$

Cela entraîne que l'unique racine strictement positive $r > 0$ du polynôme $X^n - |a_{n-1}|X^{n-1} - \dots - |a_0|$ est bornée par $1 + A$. Le Théorème 23 permet alors de conclure. \square

Corollaire 3 (Borne de Eneström et Kakeya). *Si tous les coefficients du polynôme $R(X) = a_nX^n + a_{n-1}X^{n-1} + \dots + a_0 \in \mathbb{R}[X]$ sont strictement positifs, alors les racines de R sont toutes contenues dans la couronne $\alpha \leq |z| \leq \beta$, où*

$$\alpha := \min_{1 \leq k \leq n} \left(\frac{a_{k-1}}{a_k} \right), \quad \beta := \max_{1 \leq k \leq n} \left(\frac{a_{k-1}}{a_k} \right).$$

DÉMONSTRATION. Le polynôme $P(X) = (X - \beta)/a_n \cdot R(X)$ s'écrit $P(X) = X^{n+1} - c_nX^n - \dots - c_0$, où $c_k = (\beta a_k - a_{k-1})/a_n$ pour $k \geq 1$ et $c_0 = \beta a_0/a_n$. L'hypothèse entraîne que tous les c_k sont positifs et $c_0 > 0$. Le réel β étant l'unique racine réelle de P , le Théorème 23 montre alors que toutes les autres racines de P – c'est-à-dire les racines de R – ont un module au plus égal à β . La borne inférieure s'obtient en considérant le polynôme réciproque de P . \square

2.2. Bornes a posteriori. Le résultat que nous allons présenter maintenant est tout à fait non banal en première approche, et pourtant, sa preuve est plutôt brève.

Théorème 24. *Soit $P \in \mathbb{C}[X]$ un polynôme unitaire de degré n , et $z_0 \in \mathbb{C}$ un nombre complexe quelconque. Alors :*

1. *Le disque fermé $\overline{D}(z_0, |P(z_0)|^{1/n})$ contient au moins une racine de P .*
2. *Si de plus $P'(z_0) \neq 0$, alors le disque $\overline{D}\left(z_0, n \left| \frac{P(z_0)}{P'(z_0)} \right| \right)$ aussi.*

Remarque 1. Si $P = X^n$, alors la première borne est atteinte pour tout z_0 .

DÉMONSTRATION. On peut tout d'abord se ramener au cas où $z_0 = 0$, par exemple en considérant $P_{z_0}(X) = P(X - z_0)$. Adoptons quelques notations : soient ζ_i les racines de P , et $\rho = \min_i |\zeta_i|$. On peut tout de suite écarter le cas où $\rho = 0$ dans lequel les deux assertions sont bien vérifiées. Ainsi, le polynôme réciproque de P , égal à $X^n P(\frac{1}{X}) = a_0X^n + a_1X^{n-1} + \dots + 1$ a pour racines les $\frac{1}{\zeta_i}$, et donc grâce aux relations entre coefficients et racines, on peut écrire :

$$\left| \frac{a_k}{a_0} \right| = \left| \sum_{1 \leq i_1 < \dots < i_k \leq n} \frac{1}{\zeta_{i_1} \dots \zeta_{i_k}} \right| \leq C_n^k \rho^{-k}.$$

Dans l'hypothèse où $a_k \neq 0$, on a donc $\rho^k \leq C_n^k \frac{|a_0|}{|a_k|}$. D'où les deux cas :

- Si $k = n$, alors $\rho^n \leq |a_0| = |P(0)|$.
- Si $k = 1$ et $P'(0) = a_1$ est non nul, alors $\rho \leq n \frac{|P(0)|}{|P'(0)|}$.

La preuve est donc maintenant complète. \square

3. Comptage de racines

3.1. Indice de Cauchy. Soit f une fraction rationnelle dans $\mathbb{R}(X)$.

Définition 1. L'indice de Cauchy de f en t , noté I_{t-}^{t+} vaut :

- 0 si $f(t_-) = f(t_+)$,
- 1 si $f(t_-) = -\infty$ et $f(t_+) = +\infty$,
- -1 si $f(t_-) = +\infty$ et $f(t_+) = -\infty$.

L'indice de Cauchy de f sur l'intervalle $[a, b]$ vaut

$$I_a^b := \sum_{t \in [a, b]} I_{t-}^{t+}$$

Proposition 1. Si $P \in \mathbb{R}[X]$ est tel que $P(a)P(b) \neq 0$, alors le nombre de racines de P dans l'intervalle $[a, b]$ vaut $I_a^b \left(\frac{P'}{P} \right)$.

DÉMONSTRATION. La preuve repose sur l'égalité $\frac{P'(x)}{P(x)} = \sum_{P(x_i)=0} \frac{m_i}{x-x_i}$, où m_i est la multiplicité de la racine $x_i \in [a, b]$ de P . Alors, comme $m_i > 0$, en chaque x_i , on est dans le second point de la définition, et la conclusion s'ensuit. \square

3.2. Changements de signe. Si $(a_0, \dots, a_n) \in \mathbb{R}^{n+1}$, on note $v(a_0, \dots, a_n)$ son nombre de changements de signe, c'est à dire le nombre d'indices i tels qu'il existe $k > 0$ vérifiant $a_i a_{i+k} < 0$ et $a_{i+j} = 0$ pour tout $j \in \{1, \dots, k-1\}$.

On se convaincra aisément que cela correspond bien à la notion intuitive de nombre de changements de signe (pour une famille finie de réels) qu'on peut avoir.

3.3. Suites de Sturm.

Définition 2. Une suite finie (P_0, \dots, P_n) de polynômes réels est une suite de Sturm pour l'intervalle $[a, b]$ si :

1. $P_0(a)P_0(b) \neq 0$,
2. $\forall x \in [a, b], P_n(x) \neq 0$,
3. Si, pour $x \in [a, b]$, $P_k(x) = 0$, alors soit $P_{k-1}(x)P_{k+1}(x) < 0$ (si $k > 0$), soit $P_1(x) \neq 0$ (si $k = 0$).

Le théorème suivant, démontré par Charles Sturm en 1829, s'énonce très clairement grâce aux notations introduites :

Théorème 25. Si (P_0, \dots, P_n) est une suite de Sturm pour $[a, b]$, alors $I_a^b \left(\frac{P_1}{P_0} \right) = v((P_i(a))_i) - v((P_i(b))_i)$.

DÉMONSTRATION. Tout d'abord, remarquons que $v((P_k(t))_k)$ ne peut changer qu'en un t racine d'un P_k .

D'autre part, si $P_k(t) = 0$, et $k > 0$, alors $P_{k-1}(t)P_{k+1}(t) < 0$, donc $v((P_k(t))_k)$ ne peut changer qu'en une racine de P_0 .

Reste à étudier le cas de P_0 . On se place donc autour d'une racine t de P_0 . Si le signe ϵ_- de $P_0(t_-)$ est le même que ϵ_+ celui de $P_0(t_+)$, alors $v((P_k(x))_k)$ n'en change pas en $x = t$, et on a aussi $I_{t-}^{t+} \left(\frac{P_1}{P_0} \right) = 0$ grâce à la propriété 3 vérifiée par une suite de Sturm.

Si $(\epsilon_-, \epsilon_+) = (-, +)$ et $P_1(t) > 0$, alors $I_{t-}^{t+} \left(\frac{P_1}{P_0} \right) = 1$.

Si $(\epsilon_-, \epsilon_+) = (-, +)$ et $P_1(t) < 0$, alors $I_{t_-}^+ \left(\frac{P_1}{P_0} \right) = -1$.

On obtient de même les résultats attendus dans les cas où $(\epsilon_-, \epsilon_+) = (+, -)$, ce qui conclut quant à la démonstration du théorème de Sturm : en effet, on a vérifié qu'en chaque point où pouvait changer la valeur de $v((P_k(t))_k)$, celle de $I_{t_-}^+ \left(\frac{P_1}{P_0} \right)$ changeait dans les mêmes quantités, et donc le résultat global s'obtient en passant à la somme des indices de Cauchy en les racines de P_0 . \square

Voyons maintenant quelle application concrète on peut faire de ce théorème en ce qui concerne le nombre de racines d'un polynôme.

3.4. Racines d'un polynôme dans un intervalle. Rappelons d'abord ce que l'on entend par algorithme d'Euclide signé. Étant donnés deux polynômes P_0 et P_1 , on peut appliquer l'algorithme d'Euclide habituel, mais en introduisant un signe dans les restes comme suit :

$$\begin{aligned} P_0 &= Q_1 P_1 - P_2 \\ P_1 &= Q_2 P_2 - P_3 \\ &\vdots \\ P_{m-2} &= Q_{m-1} P_{m-1} - P_m \end{aligned}$$

avec $P_m = \text{pgcd}(P_0, P_1)$.

L'algorithme décrit ci-dessus fournissant la suite $(P_i)_{2 \leq i \leq m}$ à partir de la donnée (P_0, P_1) s'appelle *l'algorithme d'Euclide signé*. Voici un résultat, de vérification très élémentaire, nous donnant un algorithme (s'appuyant de manière essentielle sur celui qu'on vient de décrire) pour construire des suites de Sturm :

Théorème 26. *Soient P_0 un polynôme réel, $a < b$ deux réels en lesquels P_0 ne s'annule pas, et $P_1 := P_0'$. Alors les polynômes (P_0, P_1, \dots, P_m) construits par l'algorithme d'Euclide signé sont tels que la suite $(\frac{P_0}{P_m}, \frac{P_1}{P_m}, \dots, 1)$ est une suite de Sturm pour l'intervalle $[a, b]$.*

En considérant conjointement la proposition 1 ainsi que les théorèmes 25 et 26, on obtient le

Corollaire 4. *Les suites de Sturm permettent de calculer le nombres de racines distinctes dans un intervalle.*

Insistons bien sur le fait qu'avec ce résultat, on ne prend pas en compte la multiplicité des racines.

Voyons tout de suite un exemple pour comprendre la puissance de ces résultats :

Exemple 1. On prend $P = 2x^3 - 7x^2 + 3x - 2$, alors on dispose de la suite

$$\begin{aligned} P_0 &= 2x^3 - 7x^2 + 3x - 2 \\ P_1 &= 6x^2 - 14x + 3 \\ P_2 &= 62x + 15 \\ P_3 &= -1. \end{aligned}$$

On choisit $a = 0$ et $b = +\infty$ (ce n'est pas gênant, il suffit en fait de prendre b plus grand que tous les zéros des polynômes intervenant dans la suite). Alors, v vaut 2 en 0 et 1 en l'infini. On en déduit que P ne possède qu'une seule racine réelle positive.

Il ne faudrait cependant pas croire que la théorie s'arrête là : il y a beaucoup d'autres configurations (c'est-à-dire différentes de la recherche de racines de polynômes réels sur un intervalle) où on peut appliquer les résultats de ce chapitre. Nous en proposons une pour élargir un peu notre horizon, c'est l'objet de la partie qui suit.

3.5. Nombre de racines dans un demi-plan. On se donne $P \in \mathbb{C}[X]$ unitaire, et on décompose P en $P = R + iS$ avec $R, S \in \mathbb{R}[X]$. On peut alors montrer qu'il existe un lien entre le nombre de racines réelles et celui de racines dans le demi-plan supérieur. L'énoncé suivant rend ce lien explicite :

Théorème 27. Soient k le nombre de racines réelles de P , m le nombre de racines de P dans $\mathcal{H} = \{z \in \mathbb{C}; \Im(z) > 0\}$, et n le degré de P . Alors,

$$m = \frac{1}{2} \left(n - k - I_{-\infty}^{+\infty} \left(\frac{S}{R} \right) \right).$$

3.6. Théorème de Sylvester-Hermite.

Théorème 28. Soit $P \in \mathbb{R}[X]$ de degré n ayant les racines ζ_1, \dots, ζ_n supposées deux à deux distinctes. Pour $k \geq 0$ on définit les sommes de Newton $N_k = \zeta_1^k + \dots + \zeta_n^k$. Alors, le nombre de racines réelles de P est égal à la signature de la forme quadratique associée à la matrice symétrique réelle

$$M = \begin{bmatrix} N_0 & N_1 & \cdots & N_{n-1} \\ N_1 & N_2 & \cdots & N_n \\ \vdots & \vdots & \dots & \vdots \\ N_{n-1} & N_n & \cdots & N_{2n-2} \end{bmatrix}$$

DÉMONSTRATION. Soit $F(x_1, \dots, x_n)$ la forme quadratique $y_1^2 + \dots + y_n^2$, où $y_r = x_1 + \zeta_r x_2 + \dots + \zeta_r^{n-1} x_n$ pour $1 \leq r \leq n$. Les coefficients de F étant symétriques en les racines ζ_i , ils sont réels. La forme F peut donc se représenter comme

$$(1) \quad h_1^2 + \dots + h_p^2 - h_{p+1}^2 - \dots - h_n^2,$$

où les h_i sont des formes linéaires en les ζ_i , à coefficients réels. La matrice de F est exactement la matrice M .

Aux racines réelles ζ_j de P correspondent des formes réelles y_j , aux racines purement imaginaires conjuguées $\zeta_k, \bar{\zeta}_k$ correspondent des formes complexes conjuguées :

$$y_k^2 + \bar{y}_k^2 = (\lambda_k + i\mu_k)^2 + (\lambda_k - i\mu_k)^2 = 2\lambda_k^2 - 2\mu_k^2.$$

Donc, le nombre $n - p$ de carrés portant un signe négatif dans (1) est égal au nombre de différents couples de racines purement imaginaires conjuguées de P . Il s'ensuit que le nombre de racines réelles de P vaut $n - 2(n - p)$, ce qui est égal à la signature de (1). \square