Résolution des systèmes polynomiaux

Éric Schost, GAGE, École polytechnique
91128 Palaiseau
Eric.Schost@polytechnique.fr

Ce texte accompagne un cours donné en avril 2003 à l'*École Jeunes Chercheurs en Algorithmique et Calcul Formel*, organisée par le GDR ALP à l'Université de Marne-la-Vallée. Il aborde quelques aspects algorithmiques liés à la résolution des systèmes de polynômes.

Des exemples de ce type de problèmes nous proviennent autant de questions mathématiques ou informatiques que des sciences de l'ingénieur (robotique, traitement du signal ou de l'image, ...). De par cette diversité, il n'existe pas *une*, mais *des* façons de résoudre des systèmes, selon que l'on choisira de calculer des approximations numériques des racines (quand le contexte le permet), telle ou telle représentation « exacte », ...

On aborde ici le problème du côté du calcul formel, en décrivant quelques méthodes symboliques de résolution des systèmes. Une large part des idées qui sont présentées ne sont pas nouvelles, et remontent à l'aube du vingtième siècle (sous le nom de théorie de l'élimination). Après une éclipse au profit de la géométrie algébrique « abstraite », ces méthodes constructives ont connu un considérable regain d'intérêt avec l'arrivée des ordinateurs ; le domaine est actuellement très actif, de sorte que de nombreuses références mentionnées ici sont extrêmement récentes.

Plan. Dans ce qui suit, on a fait le choix de ne parler que des systèmes *ayant un nombre fini de solutions* : ce choix permet au document de conserver une taille raisonnable, et couvre par ailleurs un grand nombre d'applications. À titre de motivation, on en décrit quelques-unes dans la section 1, venant du traitement du signal ou de l'image, et de la cryptologie.

Le vocabulaire géométrique nécessaire est donné dans la section 2 : on y introduit les notions de variété, de dimension et de degré, qui sont utilisées constamment dans la suite. Utilisant ce vocabulaire, on décrit dans la section 3 ce que l'on entend fréquemment par « résoudre » un système de polynômes. Ces deux sections présentent peu d'algorithmes : il s'agit avant tout de comprendre ce que l'on veut calculer.

Les trois dernières sections sont davantage orientées vers l'algorithmique, et décrivent tour à tour trois approches sur la question :

– les calculs de base de Gröbner, solution très populaire, qui étend l'idée de division avec reste (euclidienne) des polynômes en une variable aux polynômes en plusieurs variables ;

– le(s) formalisme(s) du résultant, également une généralisation à une situation multivariée d'un objet connu pour les polynômes à une variable, ici le résultant de Sylvester ;

– les méthodes de résolution « géométriques », où l'utilisation formelle de méthodes d'inspiration « numérique » (l'opérateur d'approximation de Newton) évite une explosion combinatoire à laquelle les deux premières méthodes sont confrontées.

Comparer ces familles d'algorithmes est extrêmement délicat, aussi bien du point de vue de la complexité (on ignore encore largement ce qu'il en est pour les deux premières approches) que du point de vue pratique (certains algorithmes sont implantés dans des langages de bas niveau, d'autres non). Aussi les trois dernières sections sont-elles largement indépendantes les unes des autres.

> Des parties du texte utilisent une police plus petite et un format centré. Il s'agit essentiellement de commentaires ou de digressions.

Pour en savoir plus. De nombreux aspects ne sont pas couverts dans ces notes. En voici quelques-uns, pour lesquels on donne des pointeurs.

Liens avec le calcul numérique. Motivés notamment par des problèmes issus d'ingénierie et de la physique, de nombreux algorithmes mêlent désormais calculs symboliques et numériques, afin par exemple de traiter des systèmes à coefficients approchés. Les méthodes de type « résultant » évoquées dans la section 5 sont souvent envisagées sous cet angle ; on renvoie également aux thèses de Ruatta [118] et Trébuchet [128], ainsi qu'aux travaux de Verschelde et ses collaborateurs [130].

Par ailleurs, des premières études comparatives de complexité ont été effectuées entre l'approche formelle et l'approche numérique, utilisant notamment la théorie des zéros approchés de Smale [124]. Pour davantage de détails, voir [26, 27].

L'algorithmique en une ou deux variables. Les algorithmes décrits dans la suite s'attaquent à des situations très générales, où le nombre de variables peut être arbitrairement grand.

L'algorithmique des « systèmes » à une ou deux variables est souvent très spécifique, et de nombreuses questions restent ouvertes à ce sujet. On ignore ainsi quelles sont les bornes optimales de complexité pour le calcul d'un polynôme caractéristique ou minimal modulo un polynôme univarié $F(X)$ [123], le produit diamant de Brawley et Carlitz [19, 17], le calcul du résultant bivarié évoqué dans la section 5, ...

Les représentations « triangulaires ». Dans les sections qui suivent, on se concentre sur la résolution de systèmes par ce que l'on appelle l'introduction d'*éléments primitifs*, qui permettent de se ramener au final à des manipulations de polynômes en *une seule* variable.

Les *représentations triangulaires* [133, 78, 90, 132, 7, 104, 8, 37] offrent une alternative. De manière informelle, ces représentations respectent davantage la structure des problèmes, en évitant l'introduction dudit élément primitif, au prix de la manipulation de polynômes en plusieurs variables ; on renvoie à [75] pour une synthèse détaillée.

Noter que les questions de complexité relatives aux ensembles triangulaires sont encore imparfaitement comprises [53, 120].

1 Quelques exemples d'applications

Les applications suivantes illustrent une partie des besoins en termes de résolution de systèmes polynomiaux ; elles sont issues du traitement du signal et de l'image, et de la cryptologie. On aurait également pu détailler de nombreux exemples issus de robotique [125, 91, 105], voire de la théorie des nombres, tels les systèmes Cyclic, un des *benchmarks* les plus connus pour les systèmes algébriques [16].

Schématiquement, il faut retenir que sur ces exemples, le besoin d'algorithmes rapides se fait sentir pour au moins deux raisons.

- D'une part, il faut pouvoir résoudre le plus rapidement possible des systèmes ayant un nombre « modéré » de solutions, c'est le cas dans l'exemple de traitement d'image ci-dessous, ainsi que pour des problèmes issus de robotique (par « modéré », on entend de quelques dizaines à quelques centaines de solutions). Cette exigence de rapidité est d'autant plus forte que ces systèmes sont susceptibles de venir « en série » : ils décrivent des situations paramétrées, que l'on veut résoudre pour un grand nombre de jeux de paramètres.
- Certaines applications peuvent engendrer des systèmes ayant des nombres extrêmement élevés de solutions. C'est le cas d'un exemple de cryptologie présenté ici : le nombre de solutions peut monter jusqu'à plusieurs centaines de milliers.

1.1 Traitement du signal

De nombreux problèmes issus du traitement du signal demandent de déterminer les coefficients de familles d'ondelettes ou de filtres récursifs, en respectant diverses contraintes (régularité, qualité de reconstruction, ...) ; ces contraintes se traduisent par des relations (souvent polynomiales) sur ces coefficients. L'exemple ci-dessous décrit un tel problème, posé par Mallat [98] ; on renvoie également à [51, 92] pour l'illustration de cette problématique.

Les espaces *fovéaux* ont été définis dans le but d'obtenir des techniques d'approximation efficaces pour les fonctions réelles à singularités isolées. La motivation sous-jacente est le traitement d'images : Le Pennec et Mallat ont montré dans [113] comment utiliser ces techniques pour la compression d'images.

Une construction possible d'espaces fovéaux passe par des dilatations successives de deux ondelettes mères. Les critères de détection et d'approximation des singularités que l'on impose sur ces espaces fovéaux se traduisent alors par des conditions sur ces ondelettes mères. En particulier, pour d arbitraire, on peut imposer une régularité C^d sur ces ondelettes, pour obtenir des approximations de même régularité. Les calculs montrent alors que les ondelettes à support et « oscillation » minimaux sont solutions de systèmes *non algébriques*.

La résolution de tels systèmes passe par le calcul d'approximations polynomiales. Les calculs ont été menés jusqu'à $d = 2$: cela impose de traiter des systèmes de 3 variables, ayant quelques centaines de solutions, chiffres relativement modérés au regard des performances des logiciels actuels. Ces systèmes sont néanmoins instables numériquement, ayant des grappes de racines quasiment confondues. Ceci entraîne une explosion diophantienne, les coefficients d'une « résolution » (en un sens précisé dans la section 3) ayant approximativement 30000 chiffres décimaux. Il a fallu plusieurs jours de calcul (avec l'implantation Kronecker présentée dans la section 6) pour obtenir le résultat (et plusieurs dizaines de mégaoctets pour le stocker).

1.2 Vision artificielle

Les questions de vision artificielle se prêtent bien à un traitement par le calcul formel, du fait que les modèles de caméra les plus souvent utilisés (modèle de projection sténopé, sphérique, ...) induisent des relations linéaires ou quadratiques, entre coordonnées spatiales et coordonnées dans le plan de l'image. On renvoie à [47] pour une présentation détaillée du sujet.

Les premières questions auxquelles on pense sont des questions de *reconstruction* de scène : une caméra prend un certain nombre de vues d'une scène fixe. Supposant que l'on a correctement apparié les différentes images de divers objets, et que les paramètres internes de la caméra (distance focale, ...) sont connus, il est théoriquement possible d'en déduire les positions relatives des différents points de vue.

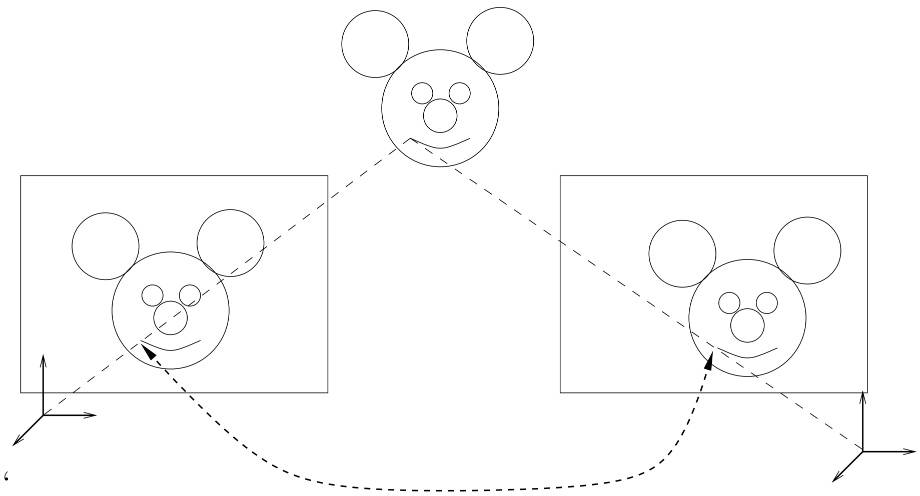

FIG. 1 – Un appariement effectué entre deux vues d'un sympathique personnage.

Supposons par exemple que l'on observe des nuages de points. Un petit calcul montre qu'alors il suffit de prendre deux images et d'y apparier 5 points (la Figure 1 représente un tel appariement) : on dispose alors d'un système de 5 équations de degré 2 en 5 inconnues, qui a (génériquement) 10 solutions. Si par contre on observe des droites, il faut évidemment trois vues ; le problème alors est ramené à étudier un système défini par 327 équations à 27 variables [40].

D'autres questions se posent en amont de la reconstruction, telles les questions de *calibration*, qui consistent à déterminer les paramètres internes de la caméra (distance focale, ...). Ces questions se formalisent également par un système d'équations polynomiales, les *équations de Kruppa* [101], que l'on voit traditionnellement comme un système de 5 quadriques en 5 inconnues, ayant 32 solutions. Ce système est très sensible au bruit des mesures ; l'application des méthodes symboliques pour l'autocalibration en situation réelle reste une question délicate.

1.3 Cryptographie

Les cryptosystèmes utilisant des protocoles de type *logarithme discret* [81] reposent sur des calculs dans un groupe fini, disons G. Leur sécurité demande que le cardinal de G soit suffisamment grand et que ses facteurs premiers ne soient pas trop petits : il faut donc savoir calculer le cardinal du groupe où l'on travaille.

Les courbes elliptiques sur des corps finis occupent une place de choix pour ce type de questions : il s'agit bien de groupes finis, pour lesquels on dispose de lois de groupe explicites. Hormis dans des situations bien particulières (petite caractéristique, ...), le calcul du cardinal se fait en suivant l'algorithme de Schoof et les améliorations dues à Elkies et Atkin [119, 6, 42]. Cet algorithme repose sur l'étude de la ℓ-torsion du groupe G, pour ℓ premier, le plus grand possible. Ici, la ℓ-torsion est décrite par un polynôme en une variable, de degré $\ell^2 - 1$.

Pour diminuer la taille des clés que l'on manipule, il est possible d'utiliser non plus des courbes elliptiques, mais les (Jacobiennes de) courbes hyperelliptiques, en particulier les courbes de genre 2. La problématique reste la même, il faut pouvoir calculer le nombre de points, et les idées de Schoof s'appliquent toujours. Cependant, en genre 2, les points de ℓ-torsion deviennent solutions d'un système de 4 équations en 4 inconnues, et sont au nombre de $\ell^4 - 1$.

Donnons quelques chiffres. On estime que pour être « sûr », le groupe où l'on travaille doit avoir un cardinal de l'ordre de 2^{160}. De là, on déduit que la courbe doit être définie sur un corps à approximativement 10^{25} éléments. Diverses améliorations de l'algorithme de Schoof en genre 2 sont données dans [54, 76] ; elles montrent que dans ces conditions il faut pouvoir déterminer la ℓ-torsion pour $\ell = 17$, voire 19, c'est-à-dire résoudre des systèmes ayant jusqu'à 130000 solutions.

1.4 Cryptanalyse

D'autres familles de cryptosystèmes sont rangées sous le nom générique de *cryposystèmes multivariables*, d'après les travaux initiés par Matsumoto, Imai [100] et popularisés par Patarin, notamment sur la base de son système HFE [112]. On présente ici cas particulier (et simplifié) de HFE, acronyme de *Hidden Field Equations*, dont l'idée consiste à dissimuler un problème univarié sous une forme multivariée.

Soient n un entier et $K = \mathbb{F}_{2^n}$ le corps fini à 2^n éléments. On se donne un polynôme F de $K[X]$, d'une forme très particulière : on choisit F comme combinaison linéaire de monômes de la forme $X^{2^\alpha + 2^\beta}$ ou X^{2^α}.

K est un espace vectoriel de dimension n sur $\mathbb{F}_2$; on peut donc écrire tout x de K comme $x = (x_1, \ldots, x_n)$, avec les x_i dans $\mathbb{F}_2$. L'application $F : K \to K$ peut alors se réécrire coordonnée par coordonnée : il existe des polynômes $F_1, \ldots, F_n$ dans $\mathbb{F}_2[X_1, \ldots, X_n]$ tels que pour tout $x = (x_1, \ldots, x_n)$ dans K, $F(x)$ s'écrive $(F_1(x_1, \ldots, x_n), \ldots, F_n(x_1, \ldots, x_n))$. Le choix des monômes intervenant dans F entraîne que les F_i ont degré 2.

HFE est un cryposystème asymétrique : le polynôme F constitue la clé privée, et la famille $F_1, \ldots, F_n$ la clé publique. Les messages sont les éléments de K, et ce sont leurs valeurs par la fonction F qui circulent en clair, de sorte que l'émetteur n'a qu'à évaluer les polynômes F_i. Le destinataire qui connaît la clé privée doit factoriser un polynôme de $K[X]$ pour retrouver le message : la complexité (non déterministe) de cette opération est polynomiale en n et en le degré de F [131]. Par contre, un truand qui ne connaît pas F doit résoudre le système $F_1, \ldots, F_n$ de n équations à n inconnues sur $\mathbb{F}_2$ pour parvenir au même résultat.

Les bornes données dans la section suivante prédisent qu'un tel système peut avoir jusqu'à 2^n solutions. Un premier défi de Patarin proposait $n = 80$: un système « générique » de cette complexité ne peut pas être résolu. Faugère a néanmoins cassé le défi de Patarin, en quelques jours de calcul, avec son algorithme F5 (voir la section 4). Intuitivement, la structure cachée du problème entraîne que le système algébrique $F_1, \ldots, F_n$ est beaucoup plus « régulier » qu'un système générique, et finit par céder.

2 Variétés, dimension, degré

Après avoir présenté ces quelques exemples, on introduit dans cette section les éléments de vocabulaire les plus utiles pour parler des systèmes polynomiaux. On commence par une rapide digression permettant d'introduire la notion de *clôture algébrique*, puis on définit (brièvement) les *variétés*, leur *dimension* et leur *degré*.

Étant donné un système de polynômes $\mathbf{F}$ à coefficients dans un corps k, la problématique de sa « résolution » couvre les questions suivantes :

1. Le système $\mathbf{F}$ a-t-il des solutions ?
2. Si oui, sont-elles en nombre fini ou infini ?
3. Si elles sont en nombre fini, comment les calculer ?

Comme le suggère la section précédente, les cas souvent rencontrés sont $k = \mathbb{Q}, \mathbb{R}$ ou $\mathbb{C}$ (corps de *caractéristique zéro*), ou $k = \mathbb{F}_p$ ou $\mathbb{F}_{p^n}$ (corps finis à p ou p^n éléments, p étant un nombre premier).

Si l'on s'intéresse uniquement aux solutions dans le corps k lui-même, les questions ci-dessus admettent alors diverses réponses, qui varient considérablement selon la nature de k.

- Si le corps de base est fini, une recherche exhaustive permet de répondre (pas très efficacement) à ces questions.

– Depuis les travaux de Matyasevic, on sait que l'existence de solutions entières à un système d'équations polynomiales est un problème indécidable; on ignore ce qu'il en est pour les solutions dans $\mathbb{Q}$.

– L'existence de solutions réelles peut se tester effectivement; les premiers algorithmes remontent à Tarski et Seidenberg, et des solutions bien meilleures en termes de complexité ont été proposées depuis, voir [23, 66, 71, 115, 72, 11, 9, 10].

Aussi, il faut imposer davantage de restrictions sur l'espace où l'on recherche les solutions si l'on désire suivre une approche unifiée.

Variétés et corps algébriquement clos. Rappelons qu'un corps K est dit *algébriquement clos* si tout polynôme (non nul) de $K[X]$ de degré d a d racines dans K. Tout corps k possède une *clôture algébrique* (unique à isomorphisme près), qui est le plus petit corps algébriquement clos le contenant; on la notera $\overline{k}$. La clôture algébrique de $\mathbb{R}$ est $\mathbb{C}$, la clôture algébrique de $\mathbb{Q}$ se note $\overline{\mathbb{Q}}$, ...

Dans la suite, on va adopter le point de vue de la géométrie algébrique classique, et utiliser le dictionnaire algèbre — géométrie, valable précisément pour les corps algébriquement clos.

Ainsi, sauf mention du contraire, quand on parlera des *solutions* d'un système de polynômes $\mathbf{F} = F_1, \ldots, F_s$ de $k[X_1, \ldots, X_n]$, il s'agira des points $x = (x_1, \ldots, x_n)$ à coordonnées dans une clôture algébrique de k, et solutions des équations $F_1(x) = \cdots = F_s(x) = 0$. On appellera *variété définie par* $\mathbf{F}$ l'ensemble de ses solutions; une telle variété porte le qualificatif d'*algébrique* — si $k = \mathbb{R}$ ou $\mathbb{C}$, ce n'est pas nécessairement une variété différentiable : les singularités sont autorisées.

Dimension. La *dimension* d'une variété algébrique est un premier invariant important. Nous en donnons un définition géométrique, qui étend la définition des espaces affines (il existe des définitions algébriques équivalentes, sur lesquelles on ne reviendra pas, voir [41]).

Soit k un corps quelconque; la dimension d'une variété algébrique $V \subset \overline{k}^n$ est le plus grand entier d tel que l'on puisse projeter V de manière *surjective* sur un sous-espace linéaire de dimension d de $\overline{k}^n$. Si k est le corps $\mathbb{R}$ ou $\mathbb{C}$, et si V est lisse, cette notion coïncide avec la dimension des variétés différentiables.

Une idée intuitive de la notion de dimension d'une variété V est le nombre de degrés de liberté dont on dispose sur V. Revenant à la définition des variétés par les polynômes, on voit chaque nouvelle équation comme une contrainte, qui devrait faire chuter la dimension de 1. Il en découle qu'une variété de dimension i dans un espace de dimension n sera « typiquement » (mais pas toujours, c'est là une question difficile) donnée par un système de $n - i$ équations à n inconnues.

Le cas qui nous intéressera est celui des variétés de *dimension zéro*, dont on montre qu'il s'agit d'ensembles *finis* de points, et qui admettent donc des descriptions finies. Dans la suite, notre problématique sera donc la suivante : comment *calculer une description d'une variété de dimension zéro* (les exemples présentés dans la section précédente rentrent tous dans ce cadre).

Degré et nombre de Bézout. Pour mesurer la complexité d'une variété, on utilise une autre grandeur fondamentale, le *degré*.

Le degré d'une variété de dimension zéro est tout simplement son nombre de points. On peut étendre cette définition de manière naturelle à des variétés de dimension positive [68], de sorte que le degré d'une variété V définie par un seul polynôme de degré d (sans carré) est précisément d (on dit alors que V est une *hypersurface*).

> La référence [68] ci-dessus donne une définition *géométrique* du degré : si V est équidimensionelle c'est le maximum du nombre de points d'intersection avec un sous-espace affine de dimension complémentaire. Il existe également des définitions de degré *algébrique*, qui prennent en compte les multiplicités (notamment par le polynôme de Hilbert, ..., voir [41]).

Les inégalités (ou égalités) de Bézout sont un outil important pour estimer le degré de variétés algébriques. Ces résultats généralisent et tirent leur nom du cas bien connu de l'*inégalité de Bézout* dans le plan : deux courbes de degrés d_1 et d_2 se coupent soit en un nombre infini de points, soit en au plus $d_1 d_2$ points (il est possible d'écrire une véritable égalité, à condition de prendre en compte les multiplicités et les solutions à l'infini).

Ces estimations se prolongent à des variétés quelconques ; on peut ainsi écrire l'inégalité $\deg(V \cap W) \leq (\deg V)(\deg W)$ (où $\deg V$ désigne bien sûr le degré de V), voir [68, 52]. En vertu de la remarque faite plus haut sur le degré d'une hypersurface, si V est une variété de dimension zéro définie par n polynômes de degré au plus d, son degré est au plus d^n. Ce nombre est appelé *nombre de Bézout* du système.

On vérifie rapidement que cette borne supérieure peut être atteinte : considérons la variété définie dans $\mathbb{C}^n$ par les polynômes $X_1^d - 1, \dots, X_n^d - 1$. Pour obtenir une solution, on a d choix possibles pour X_1, d choix possibles pour X_2, ..., de sorte qu'il y a bien d^n solutions en tout.

Le nombre de Bézout contrôle donc de manière fondamentale la complexité des calculs liés aux systèmes polynomiaux : les bornes supérieures de complexité de tous les algorithmes présentés dans la suite seront *au mieux* polynomiales en d^n.

La croissance de ce nombre est extrêmement rapide : un système de 10 équations de degré 2 peut avoir jusqu'à 1024 solutions ; un système de 10 équations de degré 3 peut avoir plus de 59000 solutions : la simple question de manipuler les solutions devient délicate.

> Il existe divers raffinements de la borne de Bézout, pour obtenir des bornes supérieures *a priori* sur le nombres de solutions d'un système polynomial. Citons notamment les estimations utilisant le volume mixte [15, 85, 79], associés aux systèmes dits *creux* : la géométrie du *support* des polynômes que l'on considère détermine le nombre de solutions qu'ils peuvent avoir.

3 Résolution des systèmes : points de vue algébrique et géométrique

Savoir quels objets calculer et leur donner un sens sont des questions primordiales pour résoudre des systèmes polynomiaux. De par la nature des objets considérés, ces questions appellent des réponses de nature à la fois géométrique et algébrique.

Dans cette section, on illustre ce double point de vue en décrivant une représentation très largement utilisée des solutions d'un système polynomial. Étant donnée une variété V de dimension zéro, cette représentation est la donnée de :

- une forme linéaire $U = \sum_{i=1}^{n} u_i X_i$;
- un *polynôme éliminant* P, dont les racines sont les valeurs que prend U sur V ;
- des *paramétrisations*, c'est-à-dire des polynômes ou des fractions rationnelles $G_1, \ldots, G_n$ telles que les relations $X_i = G_i(U)$ sont vérifiées sur V.

Sous cette forme, il devient en effet facile de compter les solutions, de trouver les composantes irréductibles de V (s'il est possible de factoriser des polynômes univariés sur le corps de base), d'approcher numériquement les racines si on travaille sur $\mathbb{R}$ ou $\mathbb{C}$, de tester si un polynôme s'annule ou non sur V, etc... Les méthodes présentées dans les trois dernières sections permettent toutes de calculer ce type de sortie.

En termes de complexité, la discussion de la section précédente montre que le degré de V peut être égal à d^n, si V est définie par un système de n équations de degré d. Ainsi, le polynôme P et les G_i sont des polynômes en une variable, de degré potentiellement d^n.

Plan de la section. On décrit pour commencer un exemple simple, l'intersection de deux courbes dans le plan ; c'est l'occasion d'introduire un point de vue géométrique sur les questions de résolution des systèmes. Dans un second temps, on reprend cette présentation, en adoptant cette fois le langage algébrique, puis l'on montre comment mener les calculs dans un langage de calcul formel, magma [2]. La dernière partie aborde un sujet plus pointu, la forme de Chow du système ; on peut se passer de l'aborder en première lecture.

La plupart des résultats présentés dans cette section sont valables sur un corps quelconque. Certains demandent des hypothèses plus fortes : on supposera que l'on travaille sur un corps de caractéristique zéro, un corps fini, ou plus généralement un corps *parfait*.

> Rappelons que k est parfait si tout polynôme irréductible de $k[X]$ a toutes ses racines simples, dans une clôture algébrique ; les corps finis et les corps de caractéristique zéro en sont des exemples. À défaut de perfection, on serait amené à placer des hypothèses de séparabilité, voire introduire des notions telles le degré de séparabilité ...

Historique. On trouve trace de ce type de représentation dès les travaux de Kronecker en 1882 [83], puis chez Macaulay en 1916 [96], van der Waerden [129], Narasimham [110], et certainement bien d'autres. Dans le milieu du calcul formel, ces techniques apparaissent dans les travaux de Chistov et Grigoriev [30], Kobayashi, Moritsugu et Hogan [74], le *Shape Lemma* de Gianni et Mora [56], les travaux de Canny [23], Renegar [115], Lakshman et Lazard [86], Giusti et Heintz [60], Alonso, Becker, Roy et Wörmann [5] ...

Notons en particulier que les notions de résolution géométrique, développée par Giusti, Heintz, Pardo et leurs collaborateurs [111, 62, 61, 59, 63] et de représentation univariée rationnelle (RUR) introduite par Rouillier [116, 117] relèvent de ce type de formalisme. Citons enfin les travaux de Kobayashi, Fujise et Furukawa [80], Canny, Kaltofen et Lakshman

[25], Canny et Emiris [24], Mourrain et Pan [107], parmi d'autres références, qui envisagent ce type de représentation en revenant à la définition donnée par Macaulay [96] du u-résultant.

3.1 Le point de vue de la géométrie : un exemple

Soient $F_1 = X_1^2 - 1, F_2 = X_1^2 + 2X_2^2 - 2$ des polynômes de $\mathbb{Q}[X_1, X_2]$. Les solutions du système $F_1 = F_2 = 0$ sont les points d'intersection de deux courbes planes, et l'on voit rapidement que ce sont les points de coordonnées

$$\left(-1, -\frac{1}{\sqrt{2}}\right), \left(-1, \frac{1}{\sqrt{2}}\right), \left(1, -\frac{1}{\sqrt{2}}\right), \left(1, \frac{1}{\sqrt{2}}\right).$$

La variété V définie par ce système est représentée sur la Figure 2.

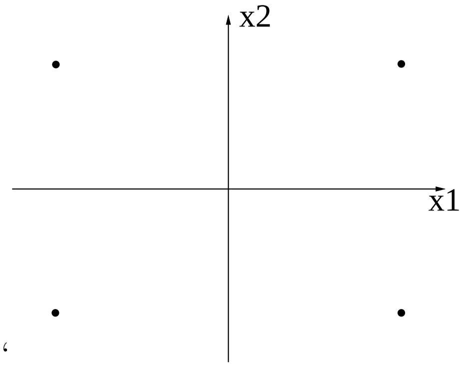

FIG. 2 – La variété V.

Algébriquement, la « résolution » d'un système polynomial se ramène souvent à une question d'*élimination* : on cherche à éliminer le plus possible de variables du système d'entrée. Comme on l'a suggéré dans le préambule, si l'on réussit à éliminer toutes les variables sauf une, on est ramené à manipuler des polynômes à une variable, pour lesquels compter, isoler, ..., les solutions est bien plus facile.

Le pendant géométrique des calculs d'élimination est l'idée de *projection*. Regardons pour commencer les projections de V sur les axes de coordonnées ; il s'agit respectivement des ensembles $\{-1, 1\}$ et $\{-1/\sqrt{2}, 1/\sqrt{2}\}$. Dans les deux cas, les projections ne sont pas injectives (puisqu'on trouve à chaque fois deux points dans les préimages), on ne peut donc pas les inverser. Graphiquement, la situation est décrite dans la Figure 3.

Au lieu de projeter sur les axes de coordonnées, considérons la projection sur une droite oblique, la droite D d'équation $X_1 - X_2 = 0$ (voir la Figure 4). La projection se fait parallèlement aux droites d'équations $X_1 + X_2 =$ constante ; on munit donc naturellement l'espace d'arrivée D de la coordonnée $U = X_1 + X_2$.

Pour P dans V, les valeurs $U(P)$ sont

$$-1 - \frac{1}{\sqrt{2}}, \quad -1 + \frac{1}{\sqrt{2}}, \quad 1 - \frac{1}{\sqrt{2}}, \quad 1 + \frac{1}{\sqrt{2}} \ ;$$

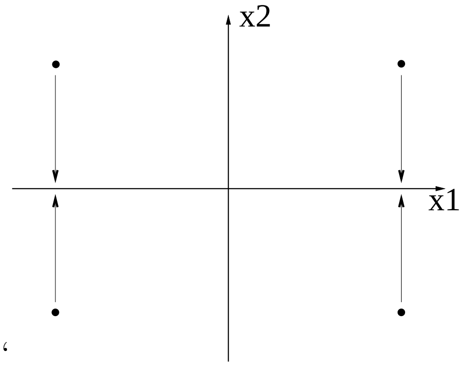

FIG. 3 – Projection de V sur l'axe X_1.

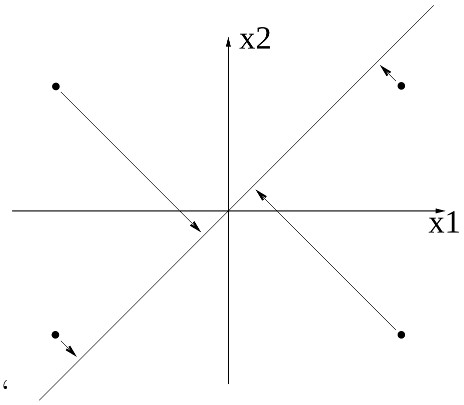

FIG. 4 – Projection de V sur la droite D

ce sont les racines du polynôme $U^4-3U^2+1/4$. La projection est injective sur V, on peut donc l'inverser au-dessus de son image, ce qui n'est rien d'autre qu'un problème d'*interpolation*.

Prenons par exemple la coordonnée X_1 : il existe un unique polynôme G_1 de degré 3 (ou moins) tel que

$$G_1\left(-1-\frac{1}{\sqrt{2}}\right)=-1,\quad G_1\left(-1+\frac{1}{\sqrt{2}}\right)=-1,\quad G_1\left(1-\frac{1}{\sqrt{2}}\right)=1,\quad G_1\left(1+\frac{1}{\sqrt{2}}\right)=1$$

Explicitement, $G_1=-U^3+7/2U$; on a alors $G_1(U(P))=X_1(P)$ pour tout P dans V. De même, on trouve que $G_2=U^3-5/2U$ vérifie $G_2(U(P))=X_2(P)$ pour P dans V.

En résumé, on a produit une description des points de V de la forme :

$$U^4-3U^2+\frac{1}{4}=0,\quad \left\{\begin{array}{lcl} X_1 & = & -U^3+\frac{7}{2}U \\ X_2 & = & U^3-\frac{5}{2}U \end{array}\right. \qquad \text{avec } U=X_1+X_2.$$

3.2 Traduction algébrique

On va reprendre et légèrement généraliser ce qui précède. On utilise désormais un langage algébrique, ce qui va nous permettre de faire des calculs.

Idéal et algèbre quotient. Soient $\mathbf{F} = F_1, \ldots, F_s$ des polynômes de $k[X_1, \ldots, X_n]$. Si x est solution du système $\mathbf{F}$, il est également solution de toutes les équations qui s'en déduisent, c'est-à-dire de tous les polynômes de la forme $\sum_{i=1}^{s} G_i F_i$, avec $G_i \in k[X_1, \ldots, X_n]$ pour $i \leq s$.

On formalise cette propriété *via* la notion d'idéal : Un *idéal* I dans un anneau R est une sous-partie (non vide) de R, stable par addition, et telle que ab appartient à I pour tous a dans R et b dans I. L'ensemble des polynômes de la forme $\sum_{i=1}^{s} G_i F_i$ est précisément un idéal de $k[X_1, \ldots, X_n]$, on dit que c'est l'*idéal engendré* par $F_1, \ldots, F_s$. Les principales propriétés de la variété définie par le système $\mathbf{F}$ se lisent sur cet idéal.

> Par exemple, le système $\mathbf{F}$ n'a aucune solution (dans une clôture algébrique de k) si et seulement si l'idéal engendré par $F_1, \ldots, F_s$ dans $k[X_1, \ldots, X_n]$ contient la constante 1. C'est le Nullstellensatz (Théorème des zéros) de Hilbert dans sa forme dite « faible ».

On définit une relation $\simeq$ sur $k[X_1, \ldots, X_n]$ par $G \simeq H \iff G - H \in I$. La relation $\simeq$ est une *relation d'équivalence* (symétrique, refléxive, transitive), on peut donc parler de ses *classes d'équivalence*, dont on note l'ensemble par $A = k[X_1, \ldots, X_n]/I$. On dit que A est le *quotient* de $k[X_1, \ldots, X_n]$ par I. Le fait que I est un idéal entraîne que A est une k-*algèbre* : on peut additionner et multiplier les classes d'équivalence entre elles, et les multiplier par des constantes.

En termes géométriques, les éléments de A définissent des *fonctions polynomiales* sur V. En effet, un élément g de A définit un polynôme de $k[X_1, \ldots, X_n]$ « à un élément de I près » ; par définition, les éléments de I sont nuls sur V, donc g définit bien la restriction à V d'une fonction polynomiale (à valeurs dans $\overline{k}$).

La notion de *radical* d'un idéal permet de rattacher l'algèbre à la géométrie : si I est un idéal dans un anneau R, son radical, noté par $\sqrt{I}$, est l'ensemble $\{r \mid \exists n \in \mathbb{N}, r^n \in I\}$. On admet que $\sqrt{I}$ est également un idéal de R, et I sera dit *radical* s'il est son propre radical.

Géométriquement, l'idéal engendré par $F_1, \ldots, F_s$ est radical si toutes les solutions du système sont « simples », c'est-à-dire sans multiplicité (pour une définition « intuitive » de la multiplicité : par exemple, si l'on perturbe lègèrement les coefficients du système, une solution multiple explose en plusieurs points distincts ; leur nombre donne la multiplicité). En passant d'un idéal à son radical, on garde les mêmes solutions, mais on « gomme » leurs multiplicités.

> Le Nullstellensatz « fort » affirme que l'ensemble des polynômes qui s'annulent sur la variété définie par $F_1, \ldots, F_s$ est le radical de l'ideal engendré par $F_1, \ldots, F_s$.

Toutes ces notions mènent au résultat fondamental suivant : *A est un k-espace vectoriel de dimension finie si et seulement si V est de dimension zéro. En outre, si l'idéal I est radical, le degré de V est égal à la dimension de A comme k-espace vectoriel.*

> C'est la dernière partie qui demande que le corps de base soit parfait ; voici un exemple typique de difficulté dans le cas non parfait. Si p est un nombre premier, le corps $\mathbb{F}_p(T)$ n'est pas parfait (c'est le prototype de corps non parfait) : le polynôme $X^p - T$ est irréductible, mais n'a qu'une seule racine de multiplicité p. Ainsi l'idéal engendré par $X^p - T$ dans $\mathbb{F}_p(T)[X]$ est-il radical, mais le quotient de $\mathbb{F}_p(T)[X]$ par cet idéal a dimension p, alors qu'il n'y a qu'une seule solution.

Matrices de multiplication et polynômes minimaux. Un premier lien entre l'algèbre et les questions d'élimination provient des notions de polynôme minimal et polynôme caractéristique d'un élément de A. Ces polynômes sont définis à partir des *matrices de multiplication* dans A.

L'algèbre A est en effet munie d'applications linéaires distinguées, les *endomorphismes de multiplication* : pour F dans A, on note M_F l'application k-linéaire

$$\begin{array}{rcl} A & \to & A \\ G & \mapsto & FG. \end{array}$$

Remarquer que pour tous F, G, on a l'égalité $M_F M_G = M_{FG}$.

Soit F dans A. Puisque M_F est une application linéaire, elle a un *polynôme minimal* $m_F \in k[T]$, de sorte que $m_F(M_F) = 0$; en utilisant l'égalité mentionnée ci-dessus, on en déduit que $m_F(F) = 0$ dans A, donc $m_F(F)$ appartient à I. Ainsi $m_F(F)$ s'annule sur toutes les solutions de $F_1 = \cdots = F_s = 0$. Plus précisément, on montre que les racines de m_F sont exactement les valeurs que prend F sur l'ensemble des solutions du système. La même conclusion reste évidemment valable si l'on considère le polynôme caractéristique de M_F, seules les multiplicités changent.

En d'autres termes, le calcul du polynôme minimal (ou caractéristique) d'un élément F de A est une opération d'*élimination*. Si F est une forme linéaire, ce polynôme minimal détermine (par ses racines) l'image de la projection sur l'axe de coordonnées correspondant.

Éléments primitifs, éléments séparants et paramétrisations. Supposons pour simplifier que l'idéal I est radical. La dimension de A comme k-espace vectoriel est alors égale au nombre de solutions de $\mathbf{F}$, que l'on appelle D. Les notions introduites ci-dessus vont permettre de compléter la traduction algébrique des constructions géométriques vues sur l'exemple.

Soit U un élément de A qui *sépare les points de* V, c'est-à-dire tel que $U(P) \neq U(Q)$ pour P et Q dans V tels que $P \neq Q$. Soit m_U le polynôme minimal de (l'application de multiplication par) U. D'après les paragraphes précédents, m_U s'annule sur toutes les valeurs $U(P)$, pour P dans V, donc il a D racines distinctes. Puisque A a dimension D, m_U a degré exactement D.

La définition du polynôme minimal montre alors que la famille $1, \ldots, U^{D-1}$ est une base de A comme k-espace vectoriel, donc tous les éléments de A s'écrivent comme des polynômes en U. Ainsi, il existe des polynômes univariés $G_1, \ldots, G_n$ tels que $X_i = G_i(U)$ dans A, pour $i = 1, \ldots, n$. On retrouve ainsi la notion de *formule d'interpolation* présentée dans l'exemple de la sous-section précédente.

Il est facile de voir que puisque I est radical, il y a en fait équivalence entre :

– U sépare les points de V ;

– la famille $1, \ldots, U^{D-1}$ est une base de A comme k-espace vectoriel.

De ce fait, on appelle U un élément *séparant* (en vertu de la première propriété) ou *primitif* (en vertu de la seconde, par analogie avec la théorie des corps). Les polynômes G_i

sont naturellement nommés *paramétrisations*. La donnée d'un tel élément U (que l'on choisit systématiquement comme étant une forme linéaire), de son polynôme minimal et des paramétrisations constitue la sortie de nombreux algorithmes de résolution de systèmes polynomiaux (représentation univariée rationnelle, résolution géométrique, ...).

Ces définitions demandent néanmoins quelques commentaires.

- Même si I est radical, il n'existe pas toujours d'élément séparant / primitif dans $k[X_1, \ldots, X_n]$ si le corps de base k est trop petit. C'est le cas de l'exemple sur $k = \mathbb{F}_2$, avec $F_1 = X_1^2 - X_1, F_2 = X_2^2 - X_2$ dans $\mathbb{F}_2[X_1, X_2]$.
- Si l'idéal I n'est pas radical, les notions d'élément séparant et d'élément primitif subsistent, mais ne sont plus équivalentes. Dans ce cas, la bonne notion est celle d'élément séparant. En effet, si par exemple le corps de base est infini, il peut ne pas exister d'élément primitif dans $k[X_1, \ldots, X_n]$, alors qu'il existera toujours un élément séparant. C'est le cas de l'exemple dans $\mathbb{Q}[X_1, X_2]$ donné par $F_1 = X_1^2, F_2 = X_2^2, F_3 = X_1X_2$.

  Dans ce cas, associées à la donnée d'un élément séparant U, il existe également des paramétrisations G_i donnant les valeurs des coordonnées X_i en fonction de U. Il s'agit cette fois d'égalités de nature géométriques, puisque les relations $X_i = G_i(U)$ ne sont plus satisfaites dans A : elles sont satisfaites modulo $\sqrt{I}$, et non plus I lui-même.

Compléments : forme de Chow. Dans ce paragraphe, que l'on peut sauter en première lecture, on suppose toujours que l'idéal I est radical. On discute des liens entre une *forme de Chow* de I et les représentations des solutions introduites ci-dessus.

Soient U un élément primitif pour I et P son polynôme minimal. Les paramétrisations G_i introduites ci-dessus sont alors des polynômes G_i tels que $X_i - G_i(U)$ est nul dans A, pour $1 \leq i \leq n$.

Il existe en fait une multitude d'autres paramétrisations possibles : pour tous polynômes N_i, D_i tels que D_i est premier avec P et tels que $G_i = N_i/D_i$ modulo P, la fraction N_i/D_i est une forme alternative de paramétrisation, tout aussi valable. Se pose alors la question de *choisir* quelle forme on retient.

Une réponse provient de l'étude de la *forme de Chow* associée à l'idéal I. Dans notre situation de dimension zéro, on peut en donner une définition simple (et calculatoire) : étant données de nouvelles indéterminées $U_1, \ldots, U_n$, la forme de Chow de I est *le polynôme minimal de la forme linéaire générique* $-U_1X_1 \cdots - U_nX_n$ (précisément : calculé dans $A \otimes_k k(U_1, \ldots, U_n)$). C'est donc un polynôme C de $k(U_1, \ldots, U_n)[T]$.

Le lien avec le calcul des paramétrisations est une remarque que l'on trouve chez Kronecker (et qui semble remonter au moins à Liouville). Par définition, $C(-U_1X_1 \cdots - U_nX_n) = 0$ dans A. Dérivons cette égalité par rapport à U_1 ; par dérivation des fonctions composées on obtient

$$\frac{\partial C}{\partial U_1} - X_1 \frac{\partial C}{\partial T} = 0 \text{ dans } A, \text{ soit } \frac{\partial C}{\partial T} X_1 = \frac{\partial C}{\partial U_1}.$$

Soient maintenant des scalaires u_i tels que $\sum u_iX_i$ est primitif pour I. Par évaluation des U_i en les u_i, C se spécialise en le polynôme minimal de $U = -\sum u_iX_i$; la formulation précédente suggère donc un choix alternatif de paramétrisation, par une *fraction rationnelle* ayant pour dénominateur *la dérivée du polynôme minimal de U.*

L'intérêt de ce choix a été souligné par Alonso, Becker, Roy et Wörmann [5] puis Rouillier [116, 117] : si le corps de base est le corps des rationnels, les coefficients qui interviennent dans ce type de paramétrisations sont de taille « modérée », nettement inférieure à celle que l'on observe dans les paramétrisations initialement introduites, les polynômes G_i. Ce choix est donc très largement privilégié en pratique.

> La remarque précédente sur la taille des coefficients se quantifie *via* la notion de *hauteur* d'une variété algébrique définie sur $\mathbb{Q}$, qui est une forme de « complexité arithmétique ». Intuitivement, toute la variété V est codée dans sa forme de Chow C : la taille des coefficients de C (et de ses dérivées partielles, qui sont utilisées ici) s'exprime donc facilement en fonction de la hauteur de V.

3.3 Une implantation en quelques lignes

Pour conclure, nous allons reprendre les définitions ci-dessus, et les appliquer à l'exemple jouet introductif. On donne pour cela des commandes dans le langage de calcul formel magma, sans se soucier de l'algorithmique sous-jacente.

On déclare le système $\mathbf{F}$, l'idéal I et le quotient A de la manière suivante (noter que l'on fait la distinction entre les X_i qui vivent dans une algèbre de polynômes, et les x_i, qui sont dans un quotient de cette algèbre).

```
> M<X1,X2>:=PolynomialRing(Rationals(),2);
> F1:=X1^2-1;
> F2:=X1^2+2*X2^2-2;
> I:=Ideal([F1,F2]);
> A<x1,x2>:=quo<M|I>;
```

Des bases privilégiées pour des algèbres telles que A sont les *bases monomiales*, c'est-à-dire formées par des (classes d'équivalence de) monômes. La commande

```
> MonomialBasis(A);
{@
    1,
    x2,
    x1,
    x1*x2
@}
```

donne ainsi une base de A comme espace vectoriel, en l'occurence $1, x_2, x_1, x_1x_2$. Les matrices de multiplication par x_1 et x_2 dans cette base sont

$$M_{x_1} = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix}, \quad M_{x_2} = \begin{bmatrix} 0 & 1/2 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1/2 \\ 0 & 0 & 1 & 0 \end{bmatrix}.$$

On y accède par les commandes

```
> M1:=RepresentationMatrix(x1);
> M2:=RepresentationMatrix(x2);
```

De là, on peut obtenir le polynôme minimal de x_1 :

```
> U<T>:=PolynomialRing(Rationals());
> MinimalPolynomial(M1);
T^2 - 1;
```

De manière équivalente, il aurait été possible de taper directement `MinimalPolynomial(x1)`.

Après s'être assuré que I est bien un idéal radical, il reste à montrer comment obtenir la résolution en elle-même, à partir de l'introduction d'un élément primitif u. Pour réaliser ceci en magma, il faut explicitement introduire la structure d'espace vectoriel sur A. On tapera donc les commandes :

```
> IsRadical(I);
true
> D:=Dimension(A);
> u:=x1+x2;
> MinimalPolynomial(u);
T^4 - 3*T^2 + 1/4
> V,h:=VectorSpace(A);
> base:=[h(u^i) : i in [0..D-1]];
> W:=VectorSpaceWithBasis(base);
> Coordinates(W,h(x1));
[ 0, 7/2, 0, -1 ]
> Coordinates(W,h(x2));
[ 0, -5/2, 0, 1 ]
```

Ceci mérite quelque explications. V est l'espace vectoriel « abstrait » sous-jacent à A, et h l'application de A vers V. La variable W désigne l'espace vectoriel engendré par les puissances de u. Puisqu'en fait $V = W$, les dernières commandes nous donnent les coordonnées de x_1 et x_2 sur la base des puissances de u. On retrouve les coefficients des polynômes G_1 et G_2 donnés dans la section précédente (et il ne serait pas très difficile d'en faire de véritables polynômes).

Sous le capot. Ce type de manipulation permet de « résoudre » des systèmes sans avoir à se soucier de calculs effectués en dessous. On peut néanmoins donner quelques détails.

- Une première phase de calculs a lieu à la création du quotient A : une *base de Gröbner* de l'idéal I est calculée. On verra dans la section 4 que ce pré-calcul fournit une base de A, et les règles nécessaires pour y effectuer les opérations $+, -, \times$.
- Le second effort calculatoire consiste à obtenir le polynôme minimal de u et les paramétrisations associées. Sous notre hypothèse simplificatrice de radicalité, il « suffit » d'effectuer des opérations d'algèbre linéaire sur des matrices 4×4 (et plus généralement

$D \times D$) pour y parvenir (l'algorithme FGLM de Faugère, Gianni, Lazard et Mora [50] fait une utilisation analogue de l'algèbre linéaire).

Pour un traitement du cas général (qui fournit également une algorithmique plus efficace que l'algèbre linéaire), on renvoie aux algorithmes de calcul de représentation univariée rationnelle donnés par Rouillier [116, 117] et implantés dans le logiciel RS [3] (voir également [18]), qu'on ne développe pas ici faute de place.

4 Bases de Gröbner

Dans cette section, on décrit comment rendre effectives les opérations $+, -, \times$ dans le quotient d'un anneau de polynômes par un idéal, question laissée en suspens ci-dessus.

Si I est un idéal de $k[X_1, \ldots, X_n]$, il possède des familles génératrices particulières, appelées *bases de Gröbner*, qui permettent d'effectuer des divisions avec reste modulo I (et donc de rendre effectives les opérations $+, -$ et $\times$ modulo I). On peut considérer ce type de division comme une généralisation de l'algorithme de division euclidienne, qui s'applique pour des polynômes à une variable.

Les applications des bases de Gröbner sont très nombreuses, et vont bien au-delà de la question de « résoudre » des systèmes de polynômes. Une large partie des questions sur les idéaux dans les anneaux de polynômes (calcul de radical, décomposition primaire, intersection, saturation, ...), sur les modules sur des anneaux de polynômes (fonction de Hilbert, résolutions, ...), peuvent être résolues effectivement en utilisant ces techniques.

Cette généralité a un prix : le calcul d'une base de Gröbner est un problème EXPSPACE-complet. D'un autre côté, les systèmes dont on a prouvé qu'ils menaient à un comportement de type exponentiel sont d'origine plus combinatoire que géométrique. Au final, le comportement pratique de ces calculs n'est pas aussi dramatique que le laisserait supposer la borne ci-dessus pour de nombreux problèmes d'origine géométrique.

Plan de la section. Dans un premier temps, on motive l'introduction d'une notion d'*ordre monomial* pour les polynômes en plusieurs variables, en revenant d'abord sur le cas des polynômes en une variable. On donne ensuite la définition d'une base de Gröbner, puis l'on décrit brièvement l'algorithme de Buchberger. Les derniers paragraphes sont dédiés à une discussion moins formelle, évoquant quelques résultats de complexité, des améliorations récentes, ...

Bibliographie. La littérature sur les bases de Gröbner est extrêmement vaste. Les ouvrages de Cox, Little et O'Shea [33, 34] forment une bonne introduction ; on pourra également consulter [13] et [41] pour aborder des sujets plus avancés.

Du point de vue logiciel, la plupart des systèmes de calcul formel généralistes permettent de calculer des bases de Gröbner : maple, magma, reduce, singular, cocoa, macaulay, ... Les efficacités relatives sont extrêmement variables, on cite souvent magma et singular parmi les plus rapides. Du côté des logiciels spécialisés, mentionnons les logiciels GB et FGB développés par Faugère [1], GB étant interfacé avec maple, mupad et la bibliothèque C++ SYNAPS [4, 114].

4.1 Introduction

Commençons par rappeler comment effectuer les opérations de réduction modulo un idéal dans le cas simple d'une seule variable.

Dans $k[X]$, tout idéal est engendré par un seul polynôme. Soit donc F dans $k[X]$, de degré D. Le quotient $k[X]/(F)$ est un k-espace vectoriel de dimension D, dont une base est $1, X, \ldots, X^{D-1}$.

L'anneau $k[X]$ est euclidien : pour tout G de $k[X]$, il existe d'uniques quotient et reste Q, R tels que $G = QF + R$, avec $\deg R < \deg F$. Cette division euclidienne fournit ainsi une écriture de la classe de G modulo (F) sur la base $1, X, \ldots, X^{D-1}$ du quotient $k[X]/(F)$, il s'agit simplement du reste R ; on dira que R est la *forme normale* de G modulo (F). Sachant effectuer la réduction modulo (F), on sait donc additioner, multiplier, et effectuer le test à 0 modulo (F).

Dans sa version « classique » ci-dessous, l'algorithme de division euclidienne est un processus itératif permettant de calculer le reste R (il existe des algorithmes de type « diviser pour régner » [131], mais là n'est pas notre propos). On suppose sans perte de généralité que le polynôme F est unitaire, ce qui nous permet d'écrire

$$F = X^D + T_F, \quad \text{avec} \quad \deg T_F < D \qquad \text{et} \qquad G = aX^{D'} + T_G, \quad \text{avec} \quad \deg T_G < D'.$$

Supposons que D' est au moins égal à D. La boucle élémentaire de la division euclidienne de G par F consiste à *réécrire* G modulo F de manière à faire disparaître son terme de plus haut degré. Pour cela, écrivons $\delta = D' - D$, et effectuons la soustraction

$$G - aX^\delta F = aX^{D'} + T_G - aX^{\delta+D} - aX^\delta T_F = T_G - aX^\delta T_F.$$

On a ainsi obtenu un polynôme congru à G modulo (F), mais de degré inférieur. Il n'y a plus qu'à itèrer ce procédé jusqu'à obtenir un polynôme de degré strictement inférieur à D, qui reste dans la même classe que G modulo (F).

Avec plus de recul, l'idée principale derrière ce procédé consiste à associer aux polynômes F et G des *termes de tête*, à savoir les termes de plus haut degré. La division avec reste de G par F passe alors par une suite de réécritures : tant que le terme de tête de F divise le terme de tête de G, on peut annuler ce dernier par une combinaison linéaire adéquate. La technologie des bases de Gröbner fournit une extension de ce processus dans des anneaux de polynômes à plusieurs variables.

4.2 Ordres monomiaux et division avec reste

Pour étendre les techniques de division avec reste, il faut pouvoir associer à tout polynôme un *terme de tête*. Pour cela, il faut définir une relation d'ordre sur les monômes : On appelle *ordre monomial* une relation $<$ sur les monômes de $k[X_1, \ldots, X_n]$ telle que :

– $<$ est une relation d'ordre total ;

– pour tous monômes n, m, m' de $k[X_1, \ldots, X_n]$, $m < m'$ implique $nm < nm'$;

– pour tous monômes n, m de $k[X_1, \ldots, X_n]$, si $n \neq 1$ alors $m < nm$.

La première condition est naturelle : il faut pouvoir comparer tous les monômes. La seconde a été implicitement utilisée dans l'exemple de la section précédente, où l'on a vu que le « terme de tête » de $X^\delta F$ était X^δ fois le terme de tête de F. La troisième condition exprime que $<$ raffine l'ordre partiel de la divisibilité.

Exemples. Dans la littérature et dans la pratique, des ordres parmi les plus fréquemment utilisés sont les suivants ; ils jouissent de nombreuses propriétés, sur lesquelles nous ne nous étendrons pas. Soient donc $\mathbf{X}^\alpha$ et $\mathbf{X}^\beta$ deux monômes, avec $\alpha = (\alpha_1, \ldots, \alpha_n)$ et $\beta = (\beta_1, \ldots, \beta_n)$.

Ordre lexicographique (lex). On dit que $\mathbf{X}^\beta < \mathbf{X}^\alpha$ pour l'ordre lexicographique si $\alpha_i > \beta_i$ pour le premier indice i tel que $\alpha_i \neq \beta_i$. C'est l'ordre du dictionnaire sur les « mots » α et β.

Ordre du degré, raffiné par l'ordre lexicographique inverse (grevlex). On dit que $\mathbf{X}^\beta < \mathbf{X}^\alpha$ pour cet ordre si $\sum_i \beta_i < \sum_i \alpha_i$, ou, en cas d'égalité, si $\alpha_i < \beta_i$ pour le dernier indice i tel que $\alpha_i \neq \beta_i$.

Division. La notion d'ordre monomial permet d'associer à tout polynôme F son *terme de tête*, c'est le terme correspondant au monôme le plus grand (apparaissant avec un coefficient non nul dans F). On peut dès lors écrire un algorithme de division avec reste pour les polynômes à plusieurs variables, qui généralise celui donné dans le cas d'une variable ; étant donnés des polynômes $F_1, \ldots, F_s$ et G, cet algorithme calcule des quotients $Q_1, \ldots, Q_s$ et un reste R tels que $G = Q_1F_1 + \cdots + Q_sF_s + R$.

L'algorithme est donné dans la Figure 5. Le terme de tête du polynôme F_i est noté L_i, pour $i \leq s$.

L'exemple suivant montre cependant que sans prendre de précaution supplémentaire, cette division ne permet pas de calculer de *forme normale* : le reste n'est pas uniquement défini.

Dans $\mathbb{Q}[X_1, X_2]$ muni de l'ordre lexicographique, considérons les polynômes $F_1 = \underline{X_1X_2} + 1$, $F_2 = \underline{X_2^2} - 1$ et $G = \underline{X_1X_2^2} - X_1$ (le trait indique le monôme de tête). L'écriture $G = X_1(X_2^2 - 1)$ montre que G appartient à (F_1, F_2) : si lors de la division, on effectue d'abord la réduction par F_2, on obtient donc bien 0. Par contre, si on réduit d'abord par F_1, on obtient comme reste $R = -X_2 - X_1$, qui ne se réduit pas à zéro.

4.3 Bases de Gröbner et algorithme de Buchberger

Dans l'exemple précédent, le processus de calcul de forme normale a été mis en défaut par l'existence de $R = -X_1 - X_2$ appartenant à l'idéal (F_1, F_2), mais dont le terme de tête n'est divisible ni par le terme de tête de F_1 ni par celui de F_2. La définition d'une base de Gröbner vise précisément à éviter ce genre de phénomène.

Input : $F_1, \ldots, F_s, G$ et un ordre monomial
Output : des polynômes $Q_1, \ldots, Q_s, R$ tels que $G = Q_1F_1 + \cdots + Q_sF_s + R$.

Poser $Q_1 = \cdots = Q_S = R = 0$
Tant que $G \neq 0$, **faire**
  Soit L_G le terme de tête de G
  Si il existe i tel que L_G est divisible par L_i **faire**
    $Q_i = Q_i + L_G/L_i$
    $G = G - (L_G/L_i)F_i$
  Sinon
    $G = G - L_G$
    $R = R + L_G$

FIG. 5 – Division avec reste dans $k[X_1, \ldots, X_n]$

Définition et propriétés. Soit $<$ un ordre monomial sur $k[X_1, \ldots, X_n]$ et I un idéal de $k[X_1, \ldots, X_n]$. Une famille de polynômes $G_1, \ldots, G_s$ forme une *base de Gröbner* de I si le terme de tête de tout élément de I est divisible par le terme de tête de l'un des G_i.

L'existence d'une base de Gröbner n'est pas *a priori* évidente. Le *lemme de Dickson* montre cependant que tout idéal monomial est finiment engendré ; il en résulte que pour tout ordre monomial, tout idéal admet une base de Gröbner finie. En outre, quitte à rajouter des conditions naturelles de minimalité (essentiellement, aucun terme d'aucun polynôme de la base ne doit être réductible par la base), cette base de Gröbner devient unique.

Les résultats suivants montrent que les bases de Gröbner rendent effectivement effectifs les opérations $+, -, \times$ modulo un idéal : Soient $<$ un ordre monomial sur $k[X_1, \ldots, X_n]$, I un idéal de $k[X_1, \ldots, X_n]$ et $\{G_1, \ldots, G_s\}$ une base de Gröbner de I. Alors :

- L'ensemble des monômes qui ne sont divisibles par le terme de tête d'aucun des G_i forme une base du quotient $k[X_1, \ldots, X_n]/I$, notée B.
- Soit F dans $k[X_1, \ldots, X_n]$. Il existe un unique R, combinaison k-linéaire des monômes de B, tel que $F = Q + R$, avec $Q \in I$. L'algorithme de division appliqué à F et $\{G_1, \ldots, G_s\}$ calcule R, quels que soit les choix effectués au cours de l'algorithme.

Calculs. Il reste à dire comment calculer une base de Gröbner. La philosophie derrière ces manipulations est une philosophie de *réécriture* : on voit les polynômes définissant un idéal comme des règles de réécriture. On passe alors d'un système de générateurs à une base de Gröbner par un processus de *complétion* de ces règles de réécriture, par l'adjonction de *paires critiques*, que l'on appelle S-polynômes ou syzygies.

Soient F et G deux polynômes de $k[X_1, \ldots, X_n]$, $<$ un ordre monomial sur $k[X_1, \ldots, X_n]$, et $a\mathbf{X}^\alpha = aX_1^{\alpha_1} \cdots X_n^{\alpha_n}$ et $b\mathbf{X}^\beta = bX_1^{\beta_1} \cdots X_n^{\beta_n}$ les termes de tête de F et G. Pour $i = 1, \ldots, n$, soit $\gamma_i = \max(\alpha_1, \beta_i)$, et notons $\mathbf{X}^\gamma = X_1^{\gamma_1} \cdots X_n^{\gamma_n}$. On appelle *syzygie* de F et G le polynôme

$$\frac{\mathbf{X}^\gamma}{a\mathbf{X}^\alpha} F - \frac{\mathbf{X}^\gamma}{b\mathbf{X}^\beta} G.$$

L'idée derrière cette définition est de forcer l'annulation des termes de tête des deux membres par collision.

À l'aide de cette définition, on peut énoncer l'algorithme de Buchberger de manière très rapide, voir la Figure 6. Il consiste à adjoindre au système initial de générateurs toutes les syzygies que l'on peut en déduire, jusqu'à stabilisation du processus. Le *critère de Buchberger* montre que l'algorithme processus calcule une base de Gröbner.

Input : $F_1, \ldots, F_s$ et un ordre monomial
Output : Une base de Gröbner de $(F_1, \ldots, F_s)$.

Poser $G = F$
Faire
  Calculer toutes les syzygies des polynomes de G pris deux a deux
  Reduire ces syzygies par G
  Rajouter les restes non nuls a G
Jusqu'a ce que tous les restes obtenus soient nuls.

FIG. 6 – Algorithme de Buchberger.

4.4 État de l'art

Bref historique. Les bases de Gröbner ont été introduites sous ce nom par Buchberger (élève de Gröbner) dans sa thèse [21, 20], mais certaines des idées sous-jacentes avaient déjà fait leur apparition auparavant.

Macaulay a ainsi introduit la notion d'ordre sur les monômes dans les années 1920 [97] ; Gröbner lui-même a utilisé cette idée pour construire des systèmes de générateurs d'idéaux de polynômes. Citons également l'introduction préalable par Hironaka des *bases standard* [73], qui sont des analogues *locaux* des bases de Gröbner (dans des anneaux de séries). Néanmoins, les notions de paire critique, la définition et l'algorithme de calcul d'une base de Gröbner apparaissent bel et bien chez Buchberger.

Depuis, ces méthodes ont été étendues dans de nombreuses directions : calculs dans les modules sur des anneaux de polynômes ou des anneaux locaux, bases de Gröbner pour des polynômes à coefficients dans un anneau euclidien (typiquement $\mathbb{Z}$, pour permettre le calcul effectif dans des schémas arithmétiques), analogues des bases de Gröbner pour des idéaux différentiels, ...

Complexité et efficacité pratique. Même pour le cas le plus élémentaire, le calcul dans $k[X_1, \ldots, X_n]$, les questions de complexité relatives aux bases de Gröbner ne sont encore que partiellement élucidées.

En toute généralité, Mayr et Meyer [103] ont montré que le calcul d'une base de Gröbner d'un idéal engendré par des polynômes de degré d pouvait faire intervenir des polynômes de degré 2^{2^d} (voir aussi [38]). Les résultats de Mayr [102] et Kühnle et Mayr [84] ont depuis montré que le calcul d'une base de Gröbner est EXPSPACE-complet (classe de complexité exponentielle en espace). Des résultats plus optimistes sont néanmoins connus dans des situations moins générales.

- Bayer et Stillman [12] ont suggéré le lien entre la complexité des calculs de base de Gröbner et la *régularité de Castelnuovo-Mumford* des idéaux. Ils montrent que sous certaines hypothèses de généricité, les polynômes dans une base de Gröbner pour l'ordre grevlex ont un degré borné par cet invariant. Ce résultat corrobore une observation empirique : les calculs pour cet ordre sont souvent les plus économiques, pour des problèmes « naturels » (d'origine géométrique et non combinatoire).
- Giusti a montré dans [57, 58] que pour un idéal I (homogène) de dimension (projective) zéro, les degrés des polynômes apparaissant dans une base de Gröbner, pour n'importe quel ordre, sont majorés par le nombre de Bézout d^n, si d est une borne sur les degrés d'un système de générateurs de I.

Quant à l'algorithme de Buchberger lui-même, même si sa complexité reste un problème ouvert, il a connu de nombreuses améliorations : utilisation de la fonction de Hilbert [127], du calcul modulaire [126] ; des algorithmes de changement d'ordre sont apparus [50, 31] et sont largement utilisés en pratique, ...

Les algorithmes F4 et F5 de Faugère [48, 49] ont apporté d'importants progrès théoriques et pratiques. Le premier d'entre eux repose sur l'interprétation du calcul d'une base de Gröbner comme un calcul d'algèbre linéaire [89], de sorte qu'il est possible d'utiliser des techniques d'algèbre linéaire rapide, afin notamment de réduire plusieurs syzygies simultanément.

Le second (déjà évoqué dans la section 1.4) propose un nouveau critère permettant d'éviter le calcul de syzygies inutiles, puisqu'il permet de prédire qu'elles se réduiront à zéro. Il en découle un algorithme qui n'effectue pas de réduction à zéro inutile si le système d'entrée forme une suite régulière. Actuellement, les implantations par Faugère de ces algorithmes semblent être les seules à avoir traité des systèmes tels que Cyclic 10 [16].

Enfin, il est possible de calculer des formes normales sans passer par le calcul d'une base de Gröbner, en calculant incrémentalement une base du quotient (donc sans introduire d'ordre monomial). Pour une présentation de ces résultats récents, qui englobent les calculs de bases de Gröbner comme cas particulier, voir [106, 109, 128].

5 Résultants

Le résultant de Sylvester est défini pour *deux* polynômes F_0, F_1 à *une* variable : c'est un élément du corps (ou de l'anneau) de base, et sa nullité est équivalente à l'existence de racines communes à F_0 et F_1. À partir de cette définition, portant sur un système surdéterminé en une variable, on sait déduire un moyen de résoudre les systèmes de 2 équations à 2 inconnues.

Il existe de nombreuses généralisations de cette notion de résultant à des polynômes en n variables : résultant de polynômes homogènes, résultant creux (torique), résiduel, ... , qui vont permettre d'aborder les systèmes multivariés, et d'établir des liens avec les notions introduites dans la section 3.

La définition qu'on donne ici du résultant de polynômes homogènes n'est pas aussi constructive que celle de Sylvester, puisqu'elle passe par la générisation des coefficients, et des considérations géométriques sur l'espace de ces coefficients. Il est néamoins possible d'en obtenir une expression à partir de déterminants extraits d'une matrice, dite de Macaulay.

Même si cette construction n'est pas utilisée en pratique, elle montre que le problème se ramène à des questions d'algèbre linéaire. Une problématique importante à l'heure actuelle est ainsi d'obtenir des formules (déterminantielles) simples pour exprimer l'ensemble des résultants connus, afin d'utiliser les techniques d'algèbre linéaire structurée [45].

Plan de la section. Dans un premier, on rappelle comment est défini le résultant « classique » de Sylvester. On en donne une définition comme d'un objet universel : c'est cette définition que l'étend ensuite au cas multivarié. On présente alors les liens entre ce résultant et la résolution des systèmes, et la construction matricielle de Macaulay pour le calculer.

On suppose connus dans cette section des rudiments de géométrie projective : l'utilisation des coordonnées projectives permet de considérer les solutions à l'infini des systèmes ; les variétés correspondantes vivent dans des *espaces projectifs* $\mathbb{P}^n(\overline{k})$ et sont définies par des polynômes *homogènes* (l'introduction de polynômes homogènes entraîne à l'occasion des indexations légèrement surprenantes dans la suite).

Bibliographie. Le livre de Cox, Little et O'Shea [34] donne les principaux traits de la construction originale de Macaulay [95], ainsi que l'essentiel des références dans la littérature récente. Le sujet est également abordé par Lang [87] avec plus de détails. Pour aller plus loin, on pourra consulter le livre de Gelfand, Kapranov et Zelevinsky [55].

De point de vue logiciel, des méthodes de résolution utilisant des des calculs de résultants (et de bézoutiens, sujet connexe mais non abordé ici) sont accessibles au sein l'environnement SYNAPS [4, 114].

5.1 Le résultant de Sylvester

Le résultant : une condition de consistance. Soient F_0 et F_1 des polynômes en une variable X sur un corps k. Dans sa définition classique, leur *résultant* $\mathrm{Res}(F_0, F_1)$ est un élément de k, ayant la propriété suivante : $\mathrm{Res}(F_0, F_1) = 0$ si et seulement si F_0 et F_1 ont une racine commune (dans $\overline{k}$).

La définition du résultant passe par la construction de la *matrice de Sylvester* associée à F_0 et F_1. Soient d_0 et d_1 les degrés respectifs de F_0 et F_1 ; pour $d \geq 0$, on note par $k[X]_{<d}$

l'espace vectoriel des polynômes de degré inférieur à d. On considère alors l'application

$$\begin{array}{rccc} \varphi \ : & k[X]_{<d_1} \times k[X]_{<d_0} & \to & k[X]_{<d_0+d_1} \\ & G_0, G_1 & \mapsto & F_0G_0 + F_1G_1. \end{array}$$

On définit le résultant de F_0, F_1 comme le déterminant de la matrice de φ dans les bases monomiales, que l'on appelle matrice de Sylvester ; $\mathrm{Res}(F_0, F_1)$ est donc bien un élément de k. Il n'est pas très difficile de s'assurer que $\mathrm{Res}(F_0, F_1)$ est différent de zéro si et seulement si 1 appartient à l'image de φ, donc si F_0 et F_1 n'ont pas de racine commune.

Le résultant comme un objet universel. Pour préparer les définitions de la section suivante, il est bon de prendre un peu de recul : au lieu de considérer *un* couple de polynômes, on les considère en famille. Fixons donc des degrés d_0 et d_1 et introduisons des indéterminées $m_{0,0}, \dots, m_{0,d_1}$ et $m_{1,0}, \dots, m_{1,d_2}$, que l'on pense être les coefficients de polynômes (génériques) $\mathfrak{F}_0$ et $\mathfrak{F}_1$.

Il est tout-à-fait possible de reprendre la définition de la matrice de Sylvester dans ce cadre. Le déterminant correspondant est un polynôme de $k[m_{0,0}, \dots, m_{0,d_1}, m_{1,0}, \dots, m_{1,d_2}]$, noté Res_{d_0,d_1}. Nous allons voir que son lieu d'annulation décrit l'ensemble des systèmes (F_0, F_1) consistants, c'est-à-dire ayant une racine commune.

Le polynôme Res_{d_0,d_1} a en effet de bonnes propriétés de spécialisation, à condition de prendre garde aux possibles chutes de degrés de F_0 et F_1. Ainsi, si F_0 et F_1 sont de degrés d_0 et d_1, alors $\mathrm{Res}(F_0, F_1)$ est évidemment la valeur de Res_{d_0,d_1} spécialisé en les coefficients de F_0 et F_1. Cependant, si F_0 et F_1 sont de degrés tous deux inférieurs à respectivement d_0 et d_1, alors Res_{d_0,d_1} s'annule, une fois la spécialisation faite (le visualiser sur la matrice de Sylvester).

Pour obtenir un point de vue unifié, un bon moyen consiste à considérer en fait des *polynômes homogènes* F_0^h et F_1^h de degrés respectifs d_0 et d_1. Une rapide inspection des cas de chutes de degrés possibles montre alors que Res_{d_0,d_1} s'annule sur les coefficients de F_0^h et F_1^h si et seulement si ils ont (au moins) une racine commune dans l'espace projectif $\mathbb{P}^1(\overline{k})$.

Résolution de systèmes 2×2. Il reste à dire comment résoudre des systèmes à l'aide du résultant. Pour cela, on utilise une idée qui sera reprise plus loin sous le nom de *variable cachée.*

Si F_0 et F_1 sont des polynômes en X_0, X_1, rien ne nous empêche de les considérer comme des polynômes en X_1, à coefficients des polynômes en X_0. La propriété de spécialisation du résultant montre que dans cette interprétation, le résultant de F_0 et F_1 « par rapport à X_1 » est un polynôme en X_0, dont les racines sont les valeurs x_0 pour lesquelles $F_0(x_0, X_1)$ et $F_1(x_0, X_1)$ ont une racine commune (ou chutent de degré tous les deux).

Ainsi, le résultant permet de déterminer les projections sur l'axe X_0 des solutions du système $F_0 = F_1 = 0$. Pour suivre jusqu'au bout la philosophie introduite dans la section 3, on voudrait obtenir des *paramétrisations*, c'est-à-dire les coordonnées X_1 de ces solutions ; c'est possible, grâce à la notion de *sous-résultant* [67, 32], que l'on ne développera pas ici (voir [131] et les références qui s'y trouvent).

Faisons quelques remarques d'algorithmique pour finir. Le résultant de deux polynômes à coefficients dans un anneau peut se calculer en un nombre d'opérations dans l'anneau quasiment linéaire en les degrés, *via* un algorithme d'Euclide rapide, ce qui est quasiment optimal [131]. Pour des résultants de polynômes bivariés, la situation est moins claire : on ne dispose pas d'algorithmes optimaux pour l'instant.

5.2 Le résultant $n \times n$

Une première généralisation du résultant de Sylvester est le *résultant* $n \times n$, introduit par Cayley [28] et Macaulay [95]. La définition s'inspire de celle de Sylvester, mais on considère cette fois $n+1$ polynômes $F_0, \dots, F_n$; pour suivre la remarque faite plus haut, on prend ces polynômes *homogènes* en $n+1$ variables $X_0, \dots, X_n$.

Les systèmes ainsi considérés sont *surdéterminés* ($n+1$ variables homogènes « équivalent » à n variables affines, auxquelles on adjoint des solutions « à l'infini »). Le résultant $n \times n$ de ces polynômes sera une condition de compatibilité pour le système.

Dans les paragraphes qui suivent, on donne brièvement la définition de ce résultant général, et on mentionne ses liens avec la résolution de systèmes. La question du calcul effectif est abordée dans la sous-section suivante.

Définition. La construction de ce résultant est géométrique, et passe par la *générisation* des coefficients : le résultant est défini dans le cas général comme un objet universel, comme cela a pu être fait dans le cas particulier du résultant de Sylvester.

Choisissons des degrés $d_0, \dots, d_n$, et considérons des polynômes homogènes de degrés $d_0, \dots, d_n$ à coefficients génériques. On note ces coefficients $m_{i,\alpha}$, et on désigne par N le nombre total de ces nouvelles indéterminées ; de la sorte, on peut associer à tout point de $\overline{k}^N$ un système homogène $F_0, \dots, F_n$.

On considère alors la variété V de $\overline{k}^N \times \mathbb{P}^n(\overline{k})$ formée des couples notés informellement par (système, solution du système). La projection de V sur le premier facteur $\overline{k}^N$ décrit donc l'ensemble des systèmes consistants, et on montre qu'elle est *exactement* définie comme les zéros d'un polynôme en N variables (l'homogénéité des systèmes considérés intervient précisément ici).

On appelle *résultant* (générique) un tel polynôme, c'est donc un polynôme en les variables $m_{i,\alpha}$, noté $\mathrm{Res}_{d_0,\dots,d_n}$. Le résultant de $F_0, \dots, F_n$ est l'évaluation du résultant générique en les coefficients de $F_0, \dots, F_n$.

Résolution de systèmes : variables cachées. Supposons maintenant que l'on veuille résoudre un système de n équations $F_0, \dots, F_{n-1}$ à n inconnues $X_0, \dots, X_{n-1}$. Pour appliquer le formalisme du résultant introduit ci-dessus, il faut se ramener à une situation surdéterminée. Pour cela, on peut soit rajouter une équation, soit supprimer une variable : c'est cette deuxième solution qui porte le nom de *variable cachée.*

Isolons donc une des variables, disons X_0 : on voit les polynômes $F_0, \dots, F_{n-1}$ comme des polynômes à $n-1$ variables $X_1, \dots, X_{n-1}$ à coefficients dans $k[X_0]$. Notons $d_0, \dots, d_{n-1}$ les degrés de $F_0, \dots, F_{n-1}$ et $F_0^h, \dots, F_{n-1}^h$ les polynômes $F_0, \dots, F_{n-1}$ homogénéisés par rapport aux variables $X_1, \dots, X_{n-1}$.

Par définition, les variables dont dépend le résultant $\mathrm{Res}_{d_0,\dots,d_{n-1}}$ sont les coefficients de polynômes « génériques » en $n-1$ variables et de dégres $d_0, \dots, d_{n-1}$. Spécialisant ces variables en les coefficients de $F_0, \dots, F_{n-1}$ vus dans $k[X_0][X_1, \dots, X_{n-1}]$, on obtient un polynôme dans $k[X_0]$. Par construction du résultant, ce polynôme s'annule pour les valeurs de X_0 au-dessus desquelles $F_0^h, \dots, F_{n-1}^h$ ont une racine commune dans $\mathbb{P}^{n-1}(\overline{k})$

Si l'on oublie les cas dégénérés dans lesquels cette racine est à l'infini, on a ainsi trouvé un moyen de calculer les coordonnées X_0 des solutions du système initial $F_0, \dots, F_{n-1}$. Comme dans le cas du résultant de Sylvester, il existe une notion de *sous-résultant multivarié* qui permet d'obtenir des paramétrisations ; on renvoie à [65, 29].

Résolution de systèmes : le u-résultant. La seconde solution pour se ramener à une situation surdéterminée est d'introduire une nouvelle équation. On va faire un choix bien particulier : on introduit de nouvelles indéterminées $U_0, U_1, \dots, U_n$, et un nouveau polynôme $F_n = U_0X_0 + \cdots + U_{n-1}X_{n-1} + U_n$.

Le résultant des polynômes $F_0, \dots, F_n$, homogénéisés et vus dans $k(U_0, \dots, U_n)[X_0, \dots, X_n]$, est un élément de $k(U_0, \dots, U_n)$. Sous les hypothèses que toutes les racines de $F_0, \dots, F_{n-1}$ sont simples, et que le système homogénéisé $F_0^h, \dots, F_{n-1}^h$ n'a pas de solution à l'infini, on montre alors la propriété suivante : *À un facteur constant près, le résultant de $F_0, \dots, F_n$ est égal à la forme de Chow de $F_0, \dots, F_{n-1}$*

La forme de Chow et son utilisation pour résoudre des systèmes ont été décrits dans la section 3 ; sa connaissance donne toute l'information dont on peut avoir besoin pour le calcul d'une représentation par polynôme éliminant et paramétrisations. De la sorte, le calcul de résultant permet de « résoudre » des systèmes au sens donné dans la section 3.

Les coefficients de la forme linéaire F_n sont traditionnellement notés U_i : ceci explique le nom de u-résultant qui est donné à cette construction. Il reste néanmoins à décrire comment on calcule des résultants.

5.3 Formulation matricielle

Tout comme le résultant de Sylvester se calcule à partir de la matrice de Sylvester, il est possible de construire le résultant $n \times n$ à partir d'une formulation matricielle, en suivant une construction de Macaulay [95].

L'idée sous-jacente est la même que celle de Sylvester : on considére une application linéaire de type $\varphi\ : (G_0, \dots, G_n) \mapsto (F_0G_0 + \cdots + F_nG_n)$. Si le résultant des F_i s'annule, le système $F_0, \dots, F_n$ est consistant, donc le monôme 1 n'appartient pas à l'image de l'application φ ; on en tire des propriétés de chute de rang, donc d'annulation de certains déterminants. Ces déterminants mènent alors au résultant.

Au-delà de l'intuition, il s'agit de donner une définition précise de l'application linéaire que l'on considère, et donc de préciser dans quel domaine on choisit les polynômes $G_0, \dots, G_n$.

Notons $d_0, \ldots, d_n$ les degrés des polynômes $F_0, \ldots, F_n$. Posons $d = \sum_{i=0}^{n} d_i - n$, et soit N le nombre de monômes homogènes de degré d en $n+1$ variables. Intuitivement, d est une forme de *régularité* du système : les monômes de degré plus grand que d peuvent se réécrire en fonction de $F_0, \ldots, F_n$, il suffit donc de faire de l'algèbre linéaire en degré d.

Plus précisément, la construction de l'application φ ci-dessus passe par une partition judicieuse de l'ensemble des monômes de degré d. On définit pour commencer S_0 comme l'ensemble des monômes homogènes de degré d divisibles par $X_0^{d_0}$. Parmi les monômes restant, on définit S_1 comme ceux étant divisibles par $X_1^{d_1}$, etc ..., pour arriver à S_n, et l'on montre que les monômes de S_n sont divisibles par $X_n^{d_n}$.

On passe des ensembles de monômes S_i aux ensembles T_i, en divisant tous les monômes de S_i par $X_i^{d_i}$. On note ensuite W_i l'espace vectoriel engendré par T_i pour $0 \leq i \leq n$ et W l'espace des polynômes homogènes de degré N. Alors l'application φ est définie par :

$$\begin{array}{rccc} \varphi : & W_0 \times \cdots \times W_n & \to & W \\ & G_0, \ldots, G_n & \mapsto & \sum_{i=0}^{n} G_i F_i. \end{array}$$

La matrice S de l'application φ dans les bases canoniques est de taille $N \times N$. C'est cette matrice qui porte le nom de matrice de Macaulay.

On montre ensuite que le résultant de $F_0, \ldots, F_n$ divise le déterminant de S. Il est même possible de caractériser le quotient, comme un déterminant extrait de S. Lorsque ce déterminant n'est pas nul, cette relation fournit un moyen théorique de calcul du résultant.

> Si $\mathbf{F} = F_0, \ldots, F_{n-1}$ sont des polynômes quelconques, on peut obtenir la *matrice de multiplication* de n'importe quel polynôme G modulo $\mathbf{F}$, par des calculs d'algèbre linéaire (complément de Schur) dans la matrice de Macaulay associée au système $F_0, \ldots, F_{n-1}, G$ (une fois homogénéisé), voir [108].

5.4 État de l'art

L'introduction du résultant remonte à Cayley et Sylvester ; Macaulay a ensuite donné les premières expressions du résultant $n \times n$ en termes d'algèbre linéaire.

Ces méthodes sont revenues sur le devant dans les années 1970 et 1980 [88], et depuis lors, de nouvelles familles de résultants ont été introduites : résultant creux (ou toriques) [55, 24, 46, 43], résultants résiduels [22], ...

> Le résultant $n \times n$ décrit ci-dessus est un objet universel, attaché à l'ensemble des systèmes de polynômes de *degrés* fixés. Ces nouveaux résultant sont liés aux familles de polynômes à *support* fixé, voire qui s'écrivent comme combinaisons linéaires de *polynômes* fixés.

Les définitions de ces objets ne rendent pas leur calcul facile. Même si elle est très creuse, la matrice de Macaulay a une taille qui rend vite les calculs impossibles (pour 10 quadriques, qui définissent donc au plus 1024 solutions, la matrice a une taille supérieure à 300000). De nombreux efforts visent donc à trouver des expressions déterminantielles simples (si possible par un seul déterminant ou le quotient de deux déterminants) pour tous les types de résultants ci-dessus.

De premiers résultats dans ce domaine sont dus à Demazure [39] pour le résultant $n \times n$. Depuis, de nombreux travaux ont vu le jour, voir notamment Jouanolou [77], D'Andrea et Dickenstein [36], D'Andrea [35], ...

Les formules déterminantielles proposées dans les références ci-dessus font également intervenir des formulations à la *Bézout* du résultant, sujet qui n'a pas été abordé ici faute de place. Pour une présentation des matrices de bézoutiens, et leurs liens avec les méthodes de résolution par dualité dans les algèbres quotient, on renvoie notamment aux travaux de Mourrain et de ses coauteurs [107, 44, 108].

6 Résolution géométrique

Depuis le milieu des années 1990, une série d'articles ont proposé des algorithmes de résolution de systèmes polynomiaux [60, 62, 61, 59, 70, 63], reposant sur une philosophie d'intersection incrémentale, adjointe à l'usage de structures de données adéquates, la représentation des polynômes *en évaluation*.

Les idées sous-jacentes à ces algorithmes s'expriment naturellement de façon géométrique ; la combinatoire des monômes et sa traduction en termes d'algèbre linéaire jouent un rôle moins marqué que pour les méthodes précédentes. Ceci explique en partie pourquoi cette famille d'algorithmes porte le nom de *résolution géométrique*.

Leur forme la plus récente [94] est un algorithme probabiliste de *décomposition équidimensionnelle*, de complexité polynomiale (en le nombre de variables, le nombre d'équations et leur degré, la *complexité d'évaluation* du système d'entrée, et des degrés algébriques qui lui sont intrinsèquement associés). C'est le meilleur résultat connu pour ce problème, et également pour notre probléme plus spécifique de résolution de systèmes de dimension zéro.

Plan de la section. Les algorithmes présentés ici utilisent une structure de donnée non « conventionnelle » pour les problèmes d'élimination, aussi en rappelle-t-on brièvement les principales idées dans un premier temps. On expose ensuite les grandes lignes de l'algorithme de résolution, dans une version simplifiée ; on termine la section sur une discussion historique et une présentation de travaux connexes.

Bibliographie. Pour une présentation plus détaillée de ces algorithmes, on renvoie aux articles originaux [60, 62, 61, 59, 70, 63] ainsi qu'à la thèse de Lecerf [94]. Une présentation historique plus complète de l'utilisation de la structure de données en évaluation pour les problèmes d'élimination se trouve dans [111].

En termes logiciels, l'algorithme de décomposition équidimensionnelle mentionné ci-dessus a été implanté par Lecerf dans le langage magma, dans le package « Kronecker » [93]. Ainsi, on ne dispose pas encore d'une implantation dans un langage de bas niveau des algorithmes présentés ici.

6.1 La représentation en évaluation

Traditionnellement, on représente les polynômes par la liste de leurs coefficients sur la base monomiale, c'est la représentation *dense* (il existe également une représentation *creuse*, où l'on représente uniquement les monômes apparaissant avec un coefficient non nul) ; ces représentations sont bien adaptées aux méthodes de réécriture présentées dans les sections 4 et 5.

La représentation en évaluation adopte un point de vue différent : les polynômes peuvent être représentés comme des *programmes* qui les évaluent, des *calculs d'évaluation* ou *Straight-Line Programs*. Il s'agit de programmes sans tests ni divisions, qui effectuent uniquement des opérations $+, -, \times$. La taille d'un tel programme, c'est-à-dire le nombre d'opérations qu'il effectue, est appelé sa *complexité d'évaluation*.

> Il faut approximativement un millier de « cases mémoire » pour représenter le polynôme $(X+1)^{1024}$ de manière dense (ou creuse). Ce polynôme s'évalue de manière beaucoup plus rapide par le *Straight-Line Program* suivant :
>
> $$\begin{array}{lllll} X_1 = X+1; & X_2 = X_1^2; & X_3 = X_2^2; & X_4 = X_3^2; & X_5 = X_4^2; \\ X_6 = X_5^2; & X_7 = X_6^2; & X_8 = X_7^2; & X_9 = X_8^2; & X_{10} = X_9^2; \end{array}$$

Il est à noter que parmi les langages de calcul formel généralistes, maple seul propose une structure de données analogue. Les *expressions* de maple sont en effet stockées comme des Graphes Orientés Acycliques, qu'il est possible de transformer en *Straight-Line Program* par les commandes du package codegen.

> Pour expérimenter sur l'exemple précédent, on pourra taper la commande `codegen[optimize]((x+1)^1024);` en maple.

À titre culturel (ce point n'intervient pas dans la suite), il faut remarquer qu'un point délicat lié à la représentation en évaluation est la difficulté du *test à zéro* : comment décider qu'un programme représente le polynôme nul, en l'évaluant un nombre « raisonable » de fois ? Des réponses probabilistes existent, on renvoie à [111] pour les détails.

6.2 Les grandes lignes de l'algorithme

L'algorithme est incrémental. Supposons que l'on cherche à résoudre le système $F_1 = \cdots = F_n = 0$. Intuitivement, à chaque étape de calcul, on va calculer une représentation de la variété définie par $(F_1, \ldots, F_i)$ qui permette de calculer « facilement » son intersection avec le lieu des zéros de F_{i+1}.

Plus encore qu'en dimension zéro, représenter une variété de dimension positive est une question difficile. La raison est combinatoire : si l'on utilise la représentation dense des polynômes, le nombre de monômes nécessaires devient rapidement prohibitif.

Pour illustrer ce point, considérons une variété V de dimension $n/2$ dans un espace de dimension n (n entier pair). On suppose que V est définie par $n/2$ polynômes de degré d, de sorte que son *degré* est $d^{n/2}$.

Un *polynôme éliminant* est l'équation définissant la projection de V sur un espace affine de dimension $n/2+1$ (le choix des dimensions entraîne que cette projection est une hypersurface). Un tel polynôme a ainsi degré $d^{n/2}$ et dépend de $n/2+1$ variables : il peut donc avoir de l'ordre de $d^{O(n^2)}$ monômes. L'espace nécessaire pour le représenter n'est plus polynomial en le nombre de Bézout d^n.

Historiquement, cette difficulté a été contournée en utilisant des *Straight-Line Programs* pour représenter d'une part le système d'entrée, mais également les polynômes qui apparaissent au cours du calcul. En effet, bien que leur *écriture* puisse nécessiter un nombre de monômes considérable, les polynômes issus des problèmes d'élimination peuvent être évalués rapidement.

Un exemple typique de polynôme provenant d'un calcul d'élimination est le déterminant. Le déterminant d'une matrice de $n \times n$ à entrées génériques $X_{1,1}, \ldots, X_{n,n}$ est un polynôme comportant $n!$ monômes. Cependant, il existe un *Straight-Line Program* de taille $O(n^4)$ permettant de l'évaluer [14].

Dans la version présentée ici, tirée de [63], cette représentation a en partie disparu : les *Straight-Line Programs* rencontrés en cours de calcul sont remplacés par l'utilisation de spécialisations judicieuses, et seul le système d'entrée est représenté en évaluation. L'algorithme ne manipule plus que des polynômes en une ou deux variables ; géométriquement, il ne travaille qu'avec des variétés de dimension zéro ou un.

Les grandes lignes de cet algorithme sont données dans la Figure 7. On y note par V_i la variété définie par $F_1, \ldots, F_i$. Dans cette version simplifiée, on suppose que les polynômes $F_1, \ldots, F_n$ forment une *suite régulière réduite*. Cette condition technique signifie que pour tout i, l'adjonction du polynôme F_{i+1} au système $F_1, \ldots, F_i$ fait chuter sa dimension de 1 et ne crée pas de multiplicité. Autrement dit, pour tout i, la variété définie par $(F_1, \ldots, F_i)$ a dimension $n-i$, et l'idéal $(F_1, \ldots, F_i)$ est radical.

Description de l'algorithme. À l'étape i, par hypothèse, la variété V_i est de dimension $n-i$. Aussi, son intersection avec un sous-espace affine de dimension i « générique » a dimension zéro. Si l'on est en coordonnées génériques, l'intersection de V_i avec l'espace défini par $X_1 = \cdots = X_{n-i} = 0$ est donc un nombre fini de points. Sur la Figure 8, les variables $X_{n-i+1}, \ldots, X_n$ sont regroupées sur l'axe vertical et l'axe horizontal correspond à X_{n-i} ; les autres variables n'interviennent pas, et ne sont pas représentées.

L'étape de remontée est un passage d'une information ponctuelle à une information globale, qui consiste à « déspécialiser » la variable X_{n-i} qui était spécialisée à 0. À la sortie, on dispose de la représentation d'une *courbe*, puisqu'on a « rajouté » une dimension en partant d'un ensemble fini de points : on a représenté cette courbe sur la Figure 9.

La dernière étape permet de conserver un invariant de boucle : l'intersection avec la nouvelle équation F_{i+1} produit un espace de dimension zéro, à savoir l'intersection de V_{i+1} avec l'espace affine défini par $X_1 = \cdots = X_{n-i-1} = 0$: ces points sont représentés sur la Figure 10.

– Application d'un changement de variables générique.

– Calcul incrémental d'une résolution. Pour $i = 1, \dots, n$, faire :

Résolution de V_i au-dessus des valeurs de spécialisation
$$X_1 = \cdots = X_{n-i} = 0$$

$\downarrow$ **Remontée de la variable $\mathbf{X_{n-i}}$**

Résolution de V_i au-dessus des valeurs de spécialisation
$$X_1 = \cdots = X_{n-i-1} = 0$$

$\downarrow$ **Intersection avec $\mathbf{F_{i+1}}$**

Résolution de V_{i+1} au-dessus des valeurs de spécialisation
$$X_1 = \cdots = X_{n-i-1} = 0$$

– Retour dans les variables d'origine.

FIG. 7 – Résolution géométrique.

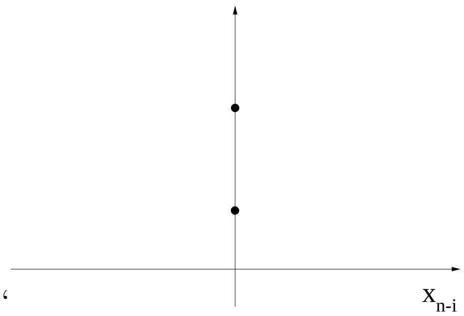

FIG. 8 – La situation à l'entrée de l'étape i.

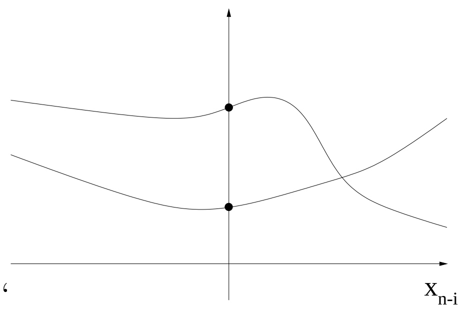

FIG. 9 – Après la remontée de la variable X_{n-i}.

Représentation des données. Dans la section 3, on a montré comment représenter les variétés de dimension zéro, qui sont des ensembles finis de points, par la donnée d'un poly-

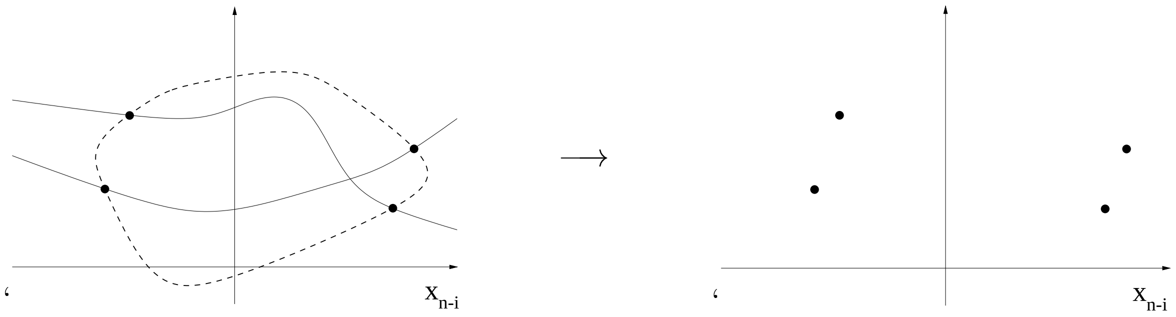

FIG. 10 – Après l'intersection avec la nouvelle équation.

nôme éliminant et de paramétrisations :

$$P(U) = 0, \quad \begin{cases} X_1 &= G_1(U) \\ &\vdots \\ X_n &= G_n(U), \end{cases}$$

où U est une combinaison linéaire des variables. C'est sous cette forme que l'algorithme de résolution géométrique présente les solutions du système $F_1, \ldots, F_n$ et, plus généralement, toutes les variétés de dimension zéro rencontrées au cours du calcul. Ainsi, l'intersection de V_i avec l'espace défini par $X_1 = \cdots = X_{n-i} = 0$ est donnée sous la forme

$$P^{(i)}(U) = 0, \quad \begin{cases} X_{n-i+1} &= G^{(i)}_{n-i+1}(U) \\ &\vdots \\ X_n &= G^{(i)}_n(U). \end{cases} \tag{1}$$

Il faut également manipuler des courbes, qui sont des variétés de dimension 1. On utilise le même type de représentation, mais il faut expliciter la dépendance en une variable libre. Puisque qu'on a effectué un changement de variable générique, toutes les courbes qui apparaissent peuvent être représentées de la sorte (on représente ici l'étape i) :

$$P^{(i)}(X_{n-i}, U) = 0, \quad \begin{cases} X_{n-i+1} &= G^{(i)}_{n-i+1}(X_{n-i}, U) \\ &\vdots \\ X_n &= G^{(i)}_n(X_{n-i}, U). \end{cases} \tag{2}$$

L'opérateur de Newton. L'étape de remontée consiste à passer de la représentation d'un objet de dimension zéro donné sous la forme (1) ci-dessus, à la représentation d'un objet de dimension un, donné sous la forme (2) ci-dessus (c'est-à-dire de la Figure 8 à la Figure 9). Pour cela, on utilise un *opérateur de Newton formel*.

L'idée sous-jacente est que les formes (1) et (2) ci-dessus décrivent des solutions du système $F_1 = \cdots = F_i = 0$, la première à précision 1 en X_{n-i} (*i.e.*, avec $X_{n-i} = 0$), et la seconde à précision infinie en X_{n-i}. Tout comme l'opérateur de Newton « classique » permet d'améliorer la précision d'approximations numériques des solutions d'un système, sa version formelle permet de retrouver la dépendance en X_{n-i} que l'on avait perdue.

L'opérateur de Newton numérique associé à un système $\mathbf{F} : \mathbb{C}^n \to \mathbb{C}^n$ est donné par la formule

$$\begin{array}{ccc} \mathbb{C}^n & \to & \mathbb{C}^n \\ z & \mapsto & z - \mathbf{Jac}(\mathbf{F})(z)^{-1}\mathbf{F}(z), \end{array}$$

où $\mathbf{Jac}(\mathbf{F})(z)$ désigne la matrice jacobienne de l'application $\mathbf{F}$. Les calculs dans la version formelle sont très semblables mais plus techniques, on renvoie à [63, 121] pour les détails.

L'opérateur numérique ci-dessus nécessite d'*évaluer* le système $\mathbf{F}$ et sa jacobienne au point z. Dans la version formelle, cette idée reste présente. C'est précisément pour prendre cet aspect en compte que l'on représente le système d'entrée *évaluation*, c'est-à-dire par un *Straight-Line Program*. Ainsi, la complexité du processus peut-elle s'exprimer en fonction de la complexité d'évaluation du système d'entrée.

Pour conclure, notons où intervient l'hypothèse d'intersection complète *réduite* : pour pouvoir appliquer l'opérateur décrit ci-dessus, il faut pouvoir inverser la matrice jacobienne du système $\mathbf{F}$ sur les points définis par $F_1 = \cdots = F_i = X_1 = \cdots = X_{n-i} = 0$. Cette condition est précisément assurée par l'hypothèse ci-dessus.

6.3 Historique

L'idée de procéder incrémentalement pour résoudre des systèmes de polynômes apparaît dès les travaux de Kronecker [83] et König [82] ; on trouve mention de ces travaux chez Macaulay [96].

Ce point de vue a été retrouvé indépendamment par Giusti, Heintz, Pardo et leurs collaborateurs depuis le début des années 1990 [60, 62, 61, 59]. L'algorithme présenté ci-dessus est tiré de [63], qui simplifie la présentation et améliore sensiblement la complexité des précédents travaux (voir également [70]).

Les algorithmes évoqués jusqu'ici fonctionnent tous sous les hypothèses, déjà mentionnées plus haut, que le système d'entrée forme une suite régulière réduite. Dans [94], Lecerf montre comment s'affranchir de ces hypothèses, et calculer la décomposition équidimensionnelle d'un système polynomial quelconque. L'algorithme qui en résulte est de complexité polynomiale en le nombre de variables, le nombre d'équations et leur degré, la complexité d'évaluation du système, et des degrés algébriques qui lui sont intrinsèquement associés.

Enfin, mentionnons que cette philosophie de l'élimination a été également utilisée pour traiter des systèmes à paramètres [64, 69, 120, 121], des problèmes de géométrie réelle [9, 10], la représentation triangulaire [120] ...Il est également possible d'aborder certains problèmes d'algèbre différentielle par ce biais : test d'observabilité en automatique [122], calcul de la fonction de Hilbert différentielle [99] ...

Références

[1] GB. http ://fgbrs.lip6.fr/.

[2] Magma. http ://www.maths.usyd.edu.au :8000/u/magma/.

[3] RS. http ://fgbrs.lip6.fr/.

[4] Synaps. http ://www-sop.inria.fr/galaad/logiciels/synaps/.

[5] M. E. Alonso, E. Becker, M.-F. Roy, and T. Wörmann. Zeros, multiplicities and idempotents for zerodimensional systems. In *Proceedings of MEGA 94*, volume 142 of *Progress in Mathematics*, pages 1–15. Birkhäuser, 1996.

[6] A. O. L. Atkin. The number of points on an elliptic curve modulo a prime. Series of e-mails to the `NMBRTHRY` mailing list, 1992.

[7] P. Aubry. *Ensembles triangulaires de polynômes et résolution de systèmes algébriques. Implantation en Axiom.* PhD thesis, Université Paris VI, 1999.

[8] P. Aubry, D. Lazard, and M. Moreno Maza. On the theories of triangular sets. *Journal of Symbolic Computation*, 28 :45–124, 1999.

[9] B. Bank, M. Giusti, J. Heintz, and G.-M. Mbakop. Polar varieties and efficient real equation solving : the hypersurface case. *Journal of Complexity*, 13 :5–27, 1997.

[10] B. Bank, M. Giusti, J. Heintz, and G.-M. Mbakop. Polar varieties and efficient real elimination. *Mathematische Zeitschrift*, 238 :115–144, 2001.

[11] S. Basu, R. Pollack, and M.-F. Roy. On the combinatorial and algebraic complexity of quantifier elimination. *Journal of the ACM*, 43 :1002–1045, 1996.

[12] D. Bayer and M. Stillman. On the complexity of computing syzygies. *Journal of Symbolic Computation*, 6 :135–147, 1988.

[13] T. Becker and V. Weispfennning. *Gröbner bases.* Springer-Verlag, 1993.

[14] S. J. Berkowitz. On computing the determinant in small parallel time using a small number of processors. *Information Processing Letters*, 18 :147–150, 1984.

[15] D. Bernstein. The number of roots of a system of equations. *Functional Anal. Appl.*, 9 :183–185, 1975.

[16] G. Björck. Functions of modulus 1 on Z_n whose Fourier transforms have constant modulus, and "cyclic n-roots". In *Recent advances in Fourier analysis and its applications (Il Ciocco, 1989)*, volume 315 of *NATO Adv. Sci. Inst. Ser. C Math. Phys. Sci.*, pages 131–140. Kluwer Acad. Publ., 1990.

[17] A. Bostan, P. Flajolet, B. Salvy, and É. Schost. Fast computation with two algebraic numbers. Research Report 4579, INRIA, 2002.

[18] A. Bostan, B. Salvy, and É. Schost. Fast algorithms for zero-dimensional polynomial systems using duality. Research Report 4291, INRIA, oct 2001.

[19] J. V. Brawley and L. Carlitz. Irreducibles and the composed product for polynomials over a finite field. *Discrete Mathematics*, 65 :115–139, 1987.

[20] B. Buchberger. *Ein Algorithmus zum Auffinden der Basiselemente des Restklassenringes nach einem nulldimensionalen Polynomideal.* PhD thesis, Université d'Innsbruck, 1965.

[21] B. Buchberger. Gröbner bases : An algorithmic method in polynomial ideal theory. In *Multidimensional System Theory*, pages 374–383. Reidel, Dordrecht, 1985.

[22] L. Busé, M. Elkadi, and B. Mourrain. Residual resultant of complete intersection. *Journal of Pure and Applied Algebra*, 164 :35–57, 2001.

[23] J. Canny. Some algebraic and geometric problems in PSPACE. In *Proceedings 20 ACM STOC*, pages 460–467, 1988.

[24] J. Canny and I. Emiris. An efficient algorithm for the sparse mixed resultant. In *Proceedings of AAECC 10*, volume 673 of *Lecture Notes in Computer Science*, pages 89–104. Springer-Verlag, 1993.

[25] J. Canny, E. Kaltofen, and Y. Lakshman. Solving systems of non-linear polynomial equations faster. In *Proceedings of ISSAC 89*, pages 121–128. ACM Press, 1989.

[26] D. Castro, K. Hägele, J. E. Morais, and L. M. Pardo. Kronecker's and Newton's approaches to solving : A first comparison. *Journal of Complexity*, 17 :212–303, 2001.

[27] D. Castro, J. L. Montana, L. M. Pardo, and J. San Martin. The distribution of condition numbers of rational data of bounded bit length. *Foundations of Computational Mathematics*, 2 :1–52, 2002.

[28] A. Cayley. On the theory of elimination. *Cambridge and Dublin Math Journal*, 3 :116–120, 1948.

[29] M. Chardin. Multivariate subresultants. *Journal of Pure and Applied Algebra*, 101 :129–138, 1995.

[30] A. L. Chistov and D. Y. Grigoriev. Polynomial-time factoring of multivariable polynomials over a global field. LOMI preprint E-5-82, Steklov Institute, Leningrad, 1982.

[31] S. Collart, M. Kalkbrener, and D. Mall. Converting bases with the Gröbner walk. *Journal of Symbolic Computation*, 24 :465–469, 1997.

[32] G. E. Collins. Subresultants and reduced polynomial remainder sequences. *Journal of the ACM*, 14 :128–142, 1967.

[33] D. Cox, J. Little, and D. O'Shea. *Ideals, varieties and algorithms.* Springer-Verlag, 1992.

[34] D. Cox, J. Little, and D. O'Shea. *Using algebraic geometry.* Springer, 1998.

[35] C. D'Andrea. Macaulay style formulas for the sparse resultant. *Transactions of the AMS*, À paraître.

[36] C. D'Andrea and A. Dickenstein. Explicit formulas for the multivariate resultant. *Journal of Pure and Applied Algebra*, 164 :59–86, 2001.

[37] S. Dellière. *Triangularisation de systèmes constructibles — Application à l'évaluation dynamique*. PhD thesis, Université de Limoges, 1999.

[38] M. Demazure. Le théorème de complexité de Mayr et Meyer. In *Géométrie algébrique et applications, I (La Rábida, 1984)*, volume 22 of *Travaux en Cours*, pages 35–58. Hermann, 1987.

[39] M. Demazure. Notes informelles de calcul formel. Centre de mathématiques, École polytechnique, 1987.

[40] M. Demazure. Sur deux problèmes de reconstruction. Research Report 882, INRIA, 1988.

[41] D. Eisenbud. *Commutative Algebra with a view toward Algebraic Geometry.* Springer, 1996.

[42] N. Elkies. Elliptic and modular curves over finite fields and related computational issues. In *Computational Perspectives on Number Theory*, pages 21–76. AMS/International Press, 1998.

[43] I. Z. Emiris. On the complexity of sparse elimination. *Journal of Complexity*, 12 :134–166, 1996.

[44] I. Z. Emiris and B. Mourrain. Matrices in elimination theory. *Journal of Symbolic Computation*, 28 :3–44, 1999.

[45] I. Z. Emiris and V. Y. Pan. Symbolic and numeric methods for exploiting structure in constructing resultant matrices. *Journal of Symbolic Computation*, 33 :393–413, 2002.

[46] I. Z. Emiris and A. Rege. Monomial bases and polynomial system solving. In *Proceedings ISSAC 94*, pages 114–122. ACM Press, 1994.

[47] O. Faugeras. *Three-dimensional computer vision : a geometric viewpoint.* MIT Press, 1993.

[48] J.-C. Faugère. A new efficient algorithm for computing Gröbner bases (F4). *Journal of Pure and Applied Algebra*, 1–3 :61–88, 1998.

[49] J.-C. Faugère. A new efficient algorithm for computing Gröbner bases without reduction to zero (F5). In *Proceedings of ISSAC 02*, pages 75 – 83. ACM Press, 2002.

[50] J.-C. Faugère, P. Gianni, D. Lazard, and T. Mora. Efficient computation of zero-dimensional Gröbner bases by change of ordering. *Journal of Symbolic Computation*, 16 :329–344, 1993.

[51] J.-C. Faugère, F. Moreau de Saint-Martin, and F. Rouillier. Design of regular nonseparable bidimensional wavelets using Gröbner basis techniques. *IEEE Transactions on Signal Processing*, 46 :845–856, 1998.

[52] W. Fulton. *Intersection Theory.* Springer Verlag, 1984.

[53] G. Gallo and B. Mishra. Efficient algorithms and bounds for Wu-Ritt characteristic sets. In *Proceedings MEGA '90.* Birkhäuser, 1990.

[54] P. Gaudry and R. Harley. Counting points on hyperelliptic curves over finite fields. In *Proceedings of ANTS IV*, volume 1838 of *Lecture Notes in Computer Science*, pages 313–332. Springer-Verlag, 2000.

[55] I. M. Gelfand, M. M. Kapranov, and A. V. Zelevinsky. *Discriminants, resultants, and multidimensional determinants.* Mathematics : Theory & Applications. Birkhäuser Boston Inc., Boston, MA, 1994.

[56] P. Gianni and T. Mora. Algebraic solution of systems of polynomial equations using Gröbner bases. In *Proceedings of AAECC 5*, volume 356 of *Lecture Notes in Computer Science*, pages 247–257. Springer-Verlag, 1989.

[57] M. Giusti. Complexity of standard bases in projective dimension zero. In *Proceedings of EUROCAL 87*, volume 378 of *Lecture Notes in Computer Science*, pages 333–335. Springer-Verlag, 1989.

[58] M. Giusti. Complexity of standard bases in projective dimension zero II. In *Proceedings of AAECC 8*, volume 508 of *Lecture Notes in Computer Science*, pages 322–329. Springer-Verlag, 1993.

[59] M. Giusti, K. Hägele, J. Heintz, J.-E. Morais, J.-L. Montaña, and L.-M. Pardo. Lower bounds for Diophantine approximation. *Journal of Pure and Applied Algebra*, 117, 118 :277–317, 1997.

[60] M. Giusti and J. Heintz. La détermination des points isolés et de la dimension d'une variété algébrique peut se faire en temps polynomial. In *Computational Algebraic Geometry and Commutative Algebra*, volume XXXIV of *Symposia Matematica*, pages 216–256. Cambridge University Press, 1993.

[61] M. Giusti, J. Heintz, J.-E. Morais, J. Morgenstern, and L.-M. Pardo. Straight-line programs in geometric elimination theory. *Journal of Pure and Applied Algebra*, 124 :101–146, 1998.

[62] M. Giusti, J. Heintz, J.-E. Morais, and L.-M. Pardo. When polynomial equation systems can be solved fast ? In *Proceedings of AAECC 11*, volume 948 of *Lecture Notes in Computer Science*, pages 205–231. Springer-Verlag, 1995.

[63] M. Giusti, G. Lecerf, and B. Salvy. A Gröbner-free alternative for polynomial system solving. *Journal of Complexity*, 17 :154–211, 2001.

[64] M. Giusti and É. Schost. Solving some overdetermined polynomial systems. In *Proceedings of ISSAC 99*, pages 1–8. ACM Press, 1999.

[65] L. González-Vega. A subresultant theory for multivariate polynomials. *Extracta Mathematicae*, 5 :150–152, 1990.

[66] D. Grigoriev and N. Vorobjov. Solving systems of polynomials inequalities in subexponential time. *Journal of Symbolic Computation*, 5 :37–64, 1988.

[67] W. Habicht. Eine Verallgemeinung des Sturmschen Wurzelzählverfahrens. *Commentarii Mathematici Helvetici*, 21 :99–116, 1948.

[68] J. Heintz. Definability and fast quantifier elimination in algebraically closed fields. *Theor. Comput. Sci.*, 24 :239–277, 1983.

[69] J. Heintz, T. Krick, S. Puddu, J. Sabia, and A. Waissbein. Deformation techniques for efficient polynomial equation solving. *Journal of Complexity*, 16 :70–109, 2000.

[70] J. Heintz, G. Matera, and A. Waissbein. On the time-space complexity of geometric elimination procedures. *Applicable Algebra in Engineering, Communication and Computing*, 11, 2001.

[71] J. Heintz, M.-F. Roy, and P. Solerno. On the complexity of semi-algebraic sets. In *Proceedings IFIP*, 1989.

[72] J. Heintz, M.-F. Roy, and P. Solerno. On the theoretical and practical complexity of the existential theory of the reals. *The Computer Journal*, 36 :427–431, 1993.

[73] H. Hironaka. Resolution of singularities of an algebraic variety over a field of characteristic zero. *Ann. of Math.*, 79 :109–326, 1964.

[74] H.Kobayashi, S. Moritsugu, and R. W. Hogan. On solving systems of algebraic equations. In *Proceedings of ISSAC 88*, volume 358 of *Lecture Notes in Computer Science*, pages 139–149. Springer-Verlag, 1988.

[75] E. Hubert. Notes on triangular sets and triangulation-decomposition algorithms I :Polynomial systems. In *Symbolic and Numerical Scientific Computations*. Springer, to appear.

[76] K. Matsuo J. Chao and S. Tsujii. An improved baby step giant step algorithm for point counting of hyperelliptic curves over finite fields. In *Proceedings of ANTS-V*, volume 2369 of *Lecture Notes in Computer Science*, pages 461–474. Springer-Verlag, 2002.

[77] J.-P. Jouanolou. Formes d'inertie et résultant : un formulaire. *Advances in Mathematics*, 126 :119–250, 1997.

[78] M. Kalkbrener. *Three contributions to elimination theory*. PhD thesis, Kepler University, Linz, 1991.

[79] A. G. Khovanskii. Newton polytopes and toric varieties. *Functional Anal. Appl.*, 11 :289–298, 1977.

[80] H. Kobayashi, T. Fujise, and A. Furukawa. On solving systems of algebraic equations by general elimination method. *Journal of Symbolic Computation*, 5 :303–320, 1988.

[81] N. Koblitz. *Algebraic Aspects of Cryptography*. Springer-Verlag, 1998.

[82] J. König. *Einleitung in die allgemeine Theorie des algebraischen Gröszen*. Druck und Verlag von B. G. Teubner, Leipzig, 1903.

[83] L. Kronecker. Grundzüge einer arithmetischen Theorie der algebraischen Grössen. *J. reine angew. Math.*, 92 :1–122, 1882.

[84] K. Kühnle and E. Mayr. Exponential space computation of gröbner bases. In *Proceedings of ISSAC 96*, pages 63–71. ACM Press, 1996.

[85] A. G. Kushnirenko. Newton polytopes and the Bézout bound. *Functional Anal. Appl.*, 10 :233–235, 1976.

[86] Y. N. Lakshman and D. Lazard. On the complexity of zero-dimensional algebraic systems. In *Effective methods in algebraic geometry*, volume 94 of *Progress in Mathematics*, pages 217–225. Birkhäuser, 1991.

[87] S. Lang. *Algebra, 3rd edition*. Addison Wesley, 1993.

[88] D. Lazard. Algèbre linéaire sur $k[x_1, \ldots, x_n]$ et application. *Bull. Soc. math. France*, 105 :165–190, 1977.

[89] D. Lazard. Gaussian elimination and resolution of systems of algebraic equations. In *Proceedings of EUROCAL 83*, volume 162 of *Lecture Notes in Computer Science*, pages 146–157. Springer-Verlag, 1983.

[90] D. Lazard. Solving zero-dimensional algebraic systems. *Journal of Symbolic Computation*, 13 :117–133, 1992.

[91] D. Lazard. Stewart platform and Gröbner bases. In *Proceedings of the third international workshop on advances of robot kinematics*, pages 136–142. Ferrare, 1992.

[92] J. Lebrun and M. Vetterli. High-order balanced multiwavelets : theory, factorization, and design. *IEEE Transactions on Signal Processing*, 49 :1918–1930, 2001.

[93] G. Lecerf. *Kronecker, a package for magma for polynomial system solving.* UMS MEDICIS, Laboratoire GAGE, http ://kronecker.medicis.polytechnique.fr/, 1999.

[94] G. Lecerf. *Une alternative aux méthodes de réécriture pour la résolution des systèmes algébriques.* PhD thesis, École polytechnique, 2001.

[95] F. S. Macaulay. On some formulas in elimination. *Proc. London Math. Soc.*, 3, 1902.

[96] F. S. Macaulay. *The Algebraic Theory of Modular Systems.* Cambridge University Press, 1916.

[97] F. S. Macaulay. Some properties of enumeration in the theory of modular systems. *Proc. London Math. Soc.*, 26 :372–393, 1927.

[98] S. Mallat. Foveal approximations for singularities. *Journal of Applied and Computational Harmonic Analysis*, 2003. À paraître.

[99] G. Matera and A. Sedoglavic. The differential Hilbert function of a differential rational mapping can be computed in polynomial time. In *Proceedings of ISSAC 02*, pages 184–191. ACM Press, 2002.

[100] T. Matsumoto and H. Imai. A class of asymmetric cryptosystems based on polynomials over finite rings. In *Proceedings of IEEE International Symposium on Information Theory*, pages 131–132, 1983.

[101] S. Maybank and O. Faugeras. A theory of self-calibration of a moving camera. *International Journal of Computer Vision*, 8(2) :123–151, 1992.

[102] E. Mayr. Membership in polynomial ideals over $\mathbb{Q}$ is exponential space complete. In *Proceedings of STACS 89*, volume 349 of *Lecture Notes in Computer Science*, pages 400–406. Springer-Verlag, 1989.

[103] E. Mayr and A. R. Meyer. The complexity of the word problem for commutative semigroups and polynomial ideals. *Advances in Maths*, 46 :305–329, 1982.

[104] M. Moreno Maza. *Calculs de pgcd au-dessus des tours d'extensions simples et résolution des systèmes d'équations algébriques.* PhD thesis, Université Paris VI, 1997.

[105] B. Mourrain. Enumeration problems in Geometry, Robotics and Vision. In L. González-Vega and T. Recio, editors, *Algorithms in Algebraic Geometry and Applications*, volume 143 of *Prog. in Math.*, pages 285–306. Birkhäuser, 1996.

[106] B. Mourrain. A new criterion for normal form algorithms. In *Proceedings of AAECC 13*, volume 1719 of *Lecture Notes in Computer Science*, pages 430–443. Springer-Verlag, 1999.

[107] B. Mourrain and V. Y. Pan. Solving special polynomial systems by using structured matrices and algebraic residues. In *Proceedings of FOCM 97*, pages 287–304. Springer, 1997.

[108] B. Mourrain and V. Y. Pan. Multivariate polynomials, duality, and structured matrices. *Journal of Complexity*, 16 :110–180, 2000.

[109] B. Mourrain and Ph. Trébuchet. Solving projective complete intersection faster. In *Proceedings of ISSAC 00*, pages 231–238. ACM Press, 2000.

[110] R. Narasimham. *Introduction to the theory of analytic spaces*, volume 25 of *Lecture notes in Mathematics*. Springer-verlag, 1966.

[111] L. M. Pardo. How lower and upper complexity bounds meet in elimination theory. In *Proceedings of AAECC 11*, volume 948 of *Lecture Notes in Computer Science*, pages 33–69. Springer-Verlag, 1995.

[112] J. Patatin. Hidden Field Equations (HFE) and Isomorphisms of Polynomials (IP) : two new families of asymmetric algorithms. In *Proceedings of Eurocrypt 96*, volume 1070 of *Lecture Notes in Computer Science*. Springer-Verlag, 1996.

[113] E. Le Pennec and S. Mallat. Bandelets representation of images. In *Proceedings of ICIP 2000*, 2000.

[114] G. Dos Reis, B. Mourrain, F. Rouillier, and P. Trébuchet. An environment for symbolic and numeric computation. In *Proceedings of ICMS 2002*, pages 239–249. World Scientific, 2002.

[115] J. Renegar. On the computational complexity and geometry of the first-order theory of the reals. Part I. *Journal of Symbolic Computation*, 13 :255–299, 1992.

[116] F. Rouillier. *Algorithmes efficaces pour l'étude des zéros réels des systèmes polynomiaux*. PhD thesis, Université de Rennes I, may 1996.

[117] F. Rouillier. Solving zero-dimensional systems through the Rational Univariate Representation. *Applicable Algebra in Engineering, Communication and Computing*, 9 :433–461, 1999.

[118] O. Ruatta. *Dualité algébrique, structures et applications*. PhD thesis, École doctorale Mathématiques et Informatique de Marseille, 2002.

[119] R. Schoof. Elliptic curves over finite fields and the computation of square roots mod p. *Mathematics of Computation*, 44 :483–494, 1985.

[120] É. Schost. Degree bounds and lifting techniques for triangular sets. In *Proceedings of ISSAC 02*, pages 238–245. ACM Press, 2002.

[121] É. Schost. Computing parametric geometric resolutions. *Applicable Algebra in Engineering, Communication and Computing*, 2003. À paraître.

[122] A. Sedoglavic. A probabilistic algorithm to test local algebraic observability in polynomial time. In *Proceedings of ISSAC 01*, pages 309–316. ACM Press, 2001.

[123] V. Shoup. Efficient computation of minimal polynomials in algebraic extensions of finite fields. In *Proceedings of ISSAC 99*, pages 53–58. ACM Press, 1999.

[124] S. Smale. The fundamental theorem of algebra and complexity theory. *Bull. Amer. Math. Soc*, 4 :1–36, 1981.

[125] D. Stewart. A platform with 6 degrees of freedom. In *institution of mechanical engineers*, volume 180, pages 371–386, 1965-66.

[126] C. Traverso. Gröbner trace algorithms. In *Proceedings of ISSAC 88*, volume 358 of *Lecture Notes in Computer Science*, pages 125–138. Springer-Verlag, 1988.

[127] C. Traverso. Hilbert functions and the Buchberger algorithm. *Journal of Symbolic Computation*, 22 :355–376, 1996.

[128] P. Trébuchet. *Vers une résolution stable et rapide des équations a ;lgébriques.* PhD thesis, Université de Paris 6, 2002.

[129] B. L. Van der Waerden. *Moderne Algebra I.* Berlin, 1930.

[130] J. Verschelde. Algorithm 795 : PHCpack : A general-purpose solver for polynomial systems by homotopy continuation. *ACM Transactions on Mathematical Software*, 25 :251–276, 1999.

[131] J. von zur Gathen and J. Gerhard. *Modern computer algebra.* Cambridge University Press, 1999.

[132] D. Wang. An elimination method for polynomial systems. *Journal of Symbolic Computation*, 16 :83–114, 1993.

[133] W.T. Wu. On zeros of algeraic equations — an application of Ritt principle. *Kexue Tongbao*, 31 :1–5, 1986.