Le rôle des structures de données dans les problèmes d'élimination

Marc GIUSTI, Joos HEINTZ, Jose Enrique MORAIS et Luis Miguel PARDO

MG : GAGE, École polytechnique, 91228 Palaiseau Cedex, France ; giusti@gage.polytechnique.fr
JH, JEM, LMP : Departamento de Matemáticas, Estadística y Computación, Universidad de Cantabria, E-39071 Santander, España ; heintz,morais,pardo@matsun1.matesco.unican.es

Résumé. Les problèmes d'élimination peuvent être résolus avec une faible complexité booléenne, à condition d'utiliser des structures de données adéquates pour représenter les polynômes (à savoir des calculs d'évaluation). Cette complexité est polynomiale en la taille des entrées et d'invariants intrinsèques de type degré et hauteur. De plus, cette dépendance en la hauteur disparaît si cette représentation en évaluation est étendue aux entiers. De notre borne supérieure découle, via un théorème des zéros effectif, une borne inférieure à la Liouville de la hauteur logarithmique des approximations diophantiennes des points de la variété. Nous considérons enfin les conséquences pour la résolution pratique des systèmes d'équations polynomiales.

The relevance of data structures for elimination problems.

Abstract. Solving elimination problems requires only a moderate amount of bit operations if one uses appropriate data structures (as e.g. straight-line programs) for the encoding of polynomials. We present an elimination procedure whose bit complexity is polynomial in the input size and in the value of two suitably defined invariants which reflect the geometric degree and the arithmetic height of the input system. From our complexity bound we deduce by means of an appropriate effective Nullstellensatz a multivariate and intrinsic version of Liouville's classical theorem on approximation of algebraic numbers by rationals. Consequences for practically solving systems of polynomial equations are drawn.

Abridged English Version

Let be given a system of n polynomials in n variables with integer coefficients, let us consider the following questions: first decide whether they define over the field of complex numbers only isolated points outside a given hypersurface. If so:
– **Computer algebra approach:** find a projection on a line general enough to insure that over every solution of an eliminating polynomial (to compute) lives only one point (to determine).
– **Numerical analysis approach:** given a rational ε, provide a rational approximation of every solution to the order ε.
In order to simplify the exposition, we assume from now on that no hypersurface has to be avoided and that we already start from a (reduced) regular sequence $f_1, \ldots, f_n$ of $\mathbf{Q}[x_1, \ldots, x_n]$ (with integer coefficients), i.e. such that the first s ones define a reduced complete intersection V_s ($s = 1, \ldots, n$). But all results will remain valid in the general case.
This note will sum up the work contained in [2], [3], [1], [8], [9], [10], [4], [5], [6], to which we refer for more details, and annouce some extensions.

The method will successively prepare the varieties V_s, according to following definitions:

Geometric solving. In this setting, it consists to put the coordinates in Noether position in order to present the ring of functions on the variety as a free module of finite rank over a polynomial ring, to find a primitive element of the corresponding integral ring extension, and rational parametrisations of the variety outside a codimension one discriminant locus.
Classical representation of polynomials and integers. A polynomial can be coded in dense representation as the vector of its coefficients. The (total) *degree* dominates this size via the number of monomials, whereas analogously the total degree of the univariate polynomial associated to the binary expansion of an integer dominates its bit length (*height*) (except for 0 and 1). Generally the (logarithmic) height of polynomial is the maximum of the heights of the absolute values of its coefficients.
Representation of polynomials and integers by straight-line programs. But a polynomial can be also be represented by a straight-line program evaluating it at an integral point, using only the arithmetic operations but neither tests nor branchings. A good measure of its size is then the couple of the *length* and *non-scalar depth* of the corresponding directed acyclic graph. Analogously the same representation can be used to code integers.
Complexity of the input. It will be measured by a 5-uple (n, d, h, L, ℓ), d and h being the maximum of the total degree and height of the equations, L and ℓ measuring (as defined above) a straight-line program evaluating them.
Complexity of the output. It is measured by a couple (δ, η). The first one is the maximum of the degrees of the intermediate varieties V_s (dominating the total degree of the polynomials occuring in the successive geometric resolutions), while the second one is, roughly speaking, the maximum of their heights. Anyway they are upper bounded respectively (through ordinary and arithmetic Bézout theorems) by d^n and $h\ell d^{An}$, where A is a universal constant, but they can be arbitrarily smaller.

Theorem 1 (an upper bound for the complexity of geometric solving): [6] *There exists a bounded error probabilistic Turing machine that from a (reduced) regular sequence of n polynomials with integer coefficients, of size dominated by (n, d, h, L, ℓ), outputs a geometric solving. The time complexity (number of bit operations executed) is polynomial in $ndhL\delta\eta$, if some intermediate and output polynomials are coded by straight-line programs. Furthermore if this representation by evaluation is extended to integers, the complexity is no longer dependent on η.*

This result has to be compared to the work of Shub-Smale [11]. As symbolic preprocessing of any system it improves drastically their numerical solving only valid furthermore for generic systems. Now any approximation of a solution given by a computer is rational, and a typical order of approximation needed to distinguish two solutions is 2^{-d^n}.

Theorem 2 (a lower bound à la Liouville): [6] *Assume that no solution is rational. There exists a universal constant C such that any denominator q of any rational approximation to the order 2^{-d^n} satisfies:*

$$\log_2 q \geq \frac{d^n}{(nd\delta)^C} - (h + \eta).$$

This lower bound implies that for systems with small degree and height, the length of the output of numeric methods is exponential in the number of variables. Thus it could happen that the classical coding of integers is not well suited to numerical methods for solving multivariate systems.

1. Position du problème

Étant donnés n polynômes à n variables, à coefficients entiers, considérons les questions suivantes : décider si la variété qu'ils définissent sur les complexes se réduit à des points isolés en dehors d'une hypersurface donnée. En ce cas :
– **Approche par le calcul formel :** trouver une projection suffisamment générale sur une droite pour qu'au dessus de chaque solution d'un polynôme éliminant (à calculer) ne vive qu'un point (à déterminer).
– **Approche par le calcul numérique :** Étant donné un rationnel $\varepsilon > 0$, donner de chaque solution une approximation rationnelle à l'ordre ε.
Ces problèmes présentant un intérêt pratique certain, il s'agit de les résoudre avec efficacité. Or les logiciels actuels, que ce soit de calcul formel ou de calcul numérique, sont fort loin de pouvoir traiter tous les problèmes de taille respectable posés par la description du monde réel. Exhiber des algorithmes théoriques admettant de bonnes bornes supérieures de complexité revêt donc une certaine importance ; a contrario, l'existence de bornes inférieures du même ordre permettrait de montrer les limites de ces deux approches.
Dans cet esprit cette note résume une série de travaux, publiés ou à paraître, consacrés au rôle crucial d'une structure de données particulière, les calculs d'évaluation, dans l'algorithmique et la complexité des problèmes d'élimination [2], [3], [1], [8], [9], [10], [4], [5], [6].
Afin de simplifier l'exposition, nous supposerons désormais qu'il n'y a pas d'hypersurface à éviter et que nous avons déjà affaire à une suite régulière (réduite) $f_1, \ldots, f_n$ de $\mathbf{Q}[x_1, \ldots, x_n]$, c'est-à-dire telle que les s premiers éléments définissent une intersection complète réduite V_s ($s = 1, \ldots, n$). Mais tous les résultats resteront valables *mutatis mutandis* dans le cas général.

2. Résolution géométrique

La méthode consiste à préparer successivement les variétés V_s par normalisation de Noether et une extension du lemme de l'élément primitif. À cette situation en dimension $r = n - s$ correspondent les définitions :
Normalisation de Noether : *Un changement linéaire de coordonnées* $(x_1, \ldots, x_n) \longrightarrow (y_1, \ldots, y_n)$ *met les variables en position de Noether si l'homomorphisme de* $\mathbf{Q}$*-algèbres* $A := \mathbf{Q}[y_1, \ldots, y_r] \longrightarrow \mathbf{Q}[y_1, \ldots, y_n]/(f_1, \ldots, f_{n-r}) =: B$ *est injectif et réalise une extension entière d'anneaux. Ici de plus* B *est une algèbre réduite qui en tant que* A*-module est libre de rang le degré affine de l'intersection complète.*
Élément primitif : *Il s'agit d'un élément* u *de* B *tel que le degré d'un de ses polynômes minimaux* q_u *soit égal au rang de* B *comme* A*-module.*
Résolution géométrique : *C'est la donnée de :*
– *un élément primitif* u *de l'extension, représenté par une forme* $\mathbb{Z}$*-linéaire en les variables* $y_{r+1}, \ldots, y_n$ *que nous noterons encore* u,
– *un polynôme minimal* $q_u \in \mathbb{Z}[y_1, \ldots, y_r, u]$ *de* u,
– *des paramétrisations* $\rho_{r+1}^{(u)} y_{r+1} - v_{r+1}^{(u)}(u), \ldots, \rho_n^{(u)} y_n - v_n^{(u)}(u)$, *où chaque* $\rho_i^{(u)}$ *est un*

polynôme non nul de $\mathbb{Z}[y_1, \ldots, y_r]$. *Leur produit* ρ *est appelé un discriminant du sytème d'équations. Les* $v_i^{(u)}$ *sont à coefficients entiers. L'ensemble réalise l'identité des idéaux localisés* $(f_1, \ldots, f_n)_\rho$ *et* $(q_u, \rho_{r+1}^{(u)} y_{r+1} - v_{r+1}^{(u)}(u), \ldots, \rho_n^{(u)} y_n - v_n^{(u)}(u))_\rho$.

3. Représentation des polynômes et des entiers

Représentation dense. Classiquement, un polynôme est représenté par le vecteur de ses coefficients (représentation en machine dite *dense*). Les entiers qui dominent la taille de ce codage sont le nombre de variables n et le *degré total* d via le nombre $\binom{d+n}{n}$ de ses monômes. De manière analogue, un entier naturel est classiquement représenté par un vecteur de bits correspondant à son développement polynomial en la base binaire. La mesure de la taille de son *écriture* est comme précédemment le degré de ce polynôme ou *hauteur logarithmique*. Par extension, la hauteur d'un polynôme à coefficients entiers est le maximum des hauteurs des valeurs absolues de ses coefficients.

Représentation creuse. Quand les équations sont des *oligonômes*, c'est-à-dire font intervenir relativement peu de monômes, on aimerait bien tenir compte seulement de leur nombre (représentation en machine dite *creuse*). Mais cette mesure n'est même pas stable par changement linéaire de coordonnées.

Calcul d'évaluation. Par ailleurs, un polynôme peut aussi être représenté en machine par un programme qui évalue sa valeur en un point, en ne faisant appel qu'aux opérations d'addition et de multiplication dans l'anneau des entiers, sans tests ni branchements. La longueur d'un tel programme ou *calcul d'évaluation* est beaucoup plus stable par changement de variables et englobe aisément les mesures ci-dessus. Cette longueur donne une première indication du temps nécessaire à ce calcul effectué par une machine séquentielle du type von Neumann. De manière analogue, certains entiers peuvent être représentés de manière bien plus comprimée comme résultat d'un calcul arithmétique ne faisant intervenir que de petits paramètres, par exemple $0, 1$. Pensons à $(2^{2^n} + 1)^{2^n}$ ...

Enfin, il s'avère judicieux de tenir compte de la profondeur non scalaire de ce calcul : il s'agit du plus long chemin que l'on puisse suivre dans ce calcul en négligeant les opérations scalaires (addition de polynômes et multiplication d'un polynôme par un entier). Cette notion correspondrait à une indication du temps nécessaire à ce calcul effectué par une machine parallèle disposant de suffisamment de processeurs.

4. Le rôle des quantités intrinsèques dans la complexité des sorties

Degré. Espérer que le processus d'élimination soit polynomial en n et L conduit à des affirmations comme $\mathcal{P} = \mathcal{NP}$ [7]. Il est alors naturel de penser qu'une mesure intrinsèque de la complexité de la variété doit nécessairement intervenir, et que c'est essentiellement le degré de la variété qui doit jouer ce rôle. En fait, c'est le *degré du système*, défini comme le plus grand des degrés des variétés intermédiaires V_s $(1 \leq s \leq n)$, qui intervient dans nos résultats de complexité.

Hauteur. De manière analogue, nous introduisons une notion de *hauteur* du système, variante des définitions classiques, reliée aux hauteurs des polynômes apparaissant dans les résolutions géométriques successives. Dans l'enveloppe de cette note, il est hors de propos de d'en donner la définition précise vu son caractère assez technique (voir [6] 2.2).

6. D'une borne supérieure de la complexité de la résolution géométrique à une borne inférieure à la Liouville

Supposons désormais que les polynômes $f_1, \ldots, f_n$ soient de degré et hauteur au plus d et h respectivement, et qu'ils soient donnés par un calcul d'évaluation de taille L et profondeur non scalaire ℓ.
D'après le théorème de Bézout, le degré δ du système est majoré par d^n ; mais il peut fort bien dépasser le degré affine de la variété. Il domine en tout cas le degré total des polynômes apparaissant dans les résolutions successives.
De manière analogue, d'un théorème de Bézout arithmétique sous la forme donnée par exemple dans [8], [9] nous déduisons que la hauteur η du système est majorée par $h\ell d^{An}$, où A est une constante universelle.

Des résultats maintenant classiques font apparaître pour obtenir la résolution géométrique, même en ne comptant que les opérations arithmétiques, une complexité polynomiale en $\binom{nd}{n}$ (voir par exemple [2]). Ceci n'est pas satisfaisant en pratique pour des problèmes où d est petit et n est très grand, rencontrés si l'on tente de simplifier la système via des plongements de Veronese ou typiques par exemple de l'optimisation non linéaire. D'où l'intérêt du :

Théorème 1 : [6] *Il existe une machine de Turing qui, de manière probabiliste mais avec une erreur bornée, à partir de toute une suite régulière (réduite) $f_1, \ldots, f_n$ de $\mathbf{Q}[x_1, \ldots, x_n]$ de taille dominée par le quintuplet (n, d, L, ℓ, h), rend une résolution géométrique en temps (nombre de bits manipulés) polynomial en $ndhL\delta\eta$, si l'on utilise la représentation par calcul d'évaluation pour stocker certains des polynômes apparaissant dans les calculs intermédiaires et les sorties. De plus, si l'on étend cette représentation en évaluation aux entiers, la dépendance en η disparait.*

Ce résultat est à comparer avec les travaux de Shub - Smale [11] qu'il améliore grandement. A la différence de ceux-ci, il constitue un véritable pré-calcul symbolique à la résolution purement numérique des systèmes d'équations polynomiales, et ceci sans restriction sur la généralité du système intersection complète. Qui plus est, le modèle booléen de la machine de Turing n'est entaché d'aucune suspicion contrairement au modèle de Blum-Shub-Smale destiné à représenter le calcul numérique en flottants. Maintenant, toute approximation des points de notre variété rendue par un ordinateur est rationnelle. Si on la code traditionnellement par une suite de bits, une *estimation à la Liouville* est une borne inférieure du nombre de bits nécessaires. Un pas dans cette direction est le :

Théorème 2 : [6] *Supposons qu'aucun point de la variété ne soit rationnel. Il existe une constante universelle C telle que tout dénominateur q de toute approximation rationnelle à ε près vérifie :*

$$\log_2 q \geq \frac{-\log_2 \varepsilon}{(nd\delta)^C} - (h + \eta).$$

Cette borne inférieure est obtenue via un théorème des zéros effectif à partir du résultat de complexité sur la résolution géométrique, via une formule de trace intrinsèque. Elle généralise le résultat classique de Liouville à une variable. Si l'on considère qu'un ordre typique d'approximation ε nécessaire pour distinguer les points de la variété est 2^{-d^n}, elle implique en particulier que pour des systèmes ayant de petits degré et hauteur, la longueur de la sortie des méthodes numériques de résolution est exponentielle en le nombre de variables. Ceci porte à penser que le codage binaire traditionnel n'est donc pas adapté à un traitement efficace de l'approximation diophantienne, donc de la résolution numérique des systèmes d'équations polynomiales.

Remerciements : Les auteurs remercient les projets espagnol et français : GDR CNRS 1026 MEDICIS et PB93–0472–C02–02 pour leur soutien.

Références bibliographiques

[1] **Fitchas N., Giusti M. et Smietanski F., 1995**. Sur la complexité du théorème des zéros, dans Proc. Second International Conference in the Caribbean (1993), M. Florenzano et al. (eds.), *Approximation & Optimization II* 8, Peter Lang Verlag, p. 274–329.

[2] **Giusti M. et Heintz J., 1993**. La détermination des points isolés et de la dimension d'une variété algébrique peut se faire en temps polynomial, dans *Computational Algebraic Geometry and Commutative Algebra*, D. Eisenbud and L. Robbiano (eds.), Symposia Matematica, vol. XXXIV, Cambridge University Press, p. 216–256.

[3] **Giusti M., Heintz J. and Sabia J, 1993**. On the efficiency of effective Nullstellensätze, *Computational Complexity* **3**, 56–95.

[4] **Giusti M., Heintz J., Morais J.E. and Pardo L.M., 1995**. When polynomial equation systems can be "solved" fast?, Proc. AAECC-11, *LNCS* 948, Springer Verlag, 205–231.

[5] **Giusti M., Heintz J., Morais J.E., Morgenstern J. et Pardo L.M.** Straight-line programs in geometric elimination theory, à paraître dans *J. Pure and Applied Algebra.*

[6] **Giusti M., Haegele K., Heintz J., Montaña J.L., Morais J.E. and Pardo L.M.** Lower bounds for diophantine approximations, à paraître dans Proc. MEGA'96, *J. Pure and Applied Algebra.*

[7] **Heintz J. and Morgenstern J., 1993**. On the intrinsic complexity of elimination theory, *J. of Complexity* 9, p. 471–498.

[8] **Krick T. and Pardo L.M., 1996**. A Computational Method for Diophantine Approximation, Algorithms in Algebraic Geometry and Applications, Proc. MEGA'94, L.Gonzalez Vega and T. Recio eds., *Progress in Mathematics* 143, Birkhäuser Verlag, p. 193–253.

[9] **Krick T. et Pardo L.M., 1994**. Une approche informatique pour l'approximation diophantienne, *C. R. de l'Acad. des Sci. Paris*, 318, Série I, no. 5, p. 407–412.

[10] **Pardo L.M., 1995**. How lower and upper complexity bounds meet in elimination theory, Proc. AAECC-11, *Lecture Notes in Computer Science* 948, Springer Verlag, p. 33–69.

[11] **Shub M. and Smale S., 1994**. Complexity of Bézout's theorem V : Polynomial time, *Theoretical Comp. Sci.* 133, p. 141–164.