Sharper Complexity Bounds for Zero-dimensional Gröbner Bases and Polynomial System Solving

Amir Hashemi, Daniel Lazard

INRIA

INSTITUT NATIONAL DE RECHERCHE EN INFORMATIQUE ET EN AUTOMATIQUE

Sharper Complexity Bounds for Zero–dimensional Gröbner Bases and Polynomial System Solving

Amir Hashemi — Daniel Lazard

N° 5491

February 2005

Thème SYM

N 0249-6399 ISRN INRIA/RR--5491--FR+ENG

Sharper Complexity Bounds for Zero–dimensional Gröbner Bases and Polynomial System Solving

Amir Hashemi , Daniel Lazard

Thème SYM — Systèmes symboliques
Projet SALSA

Rapport de recherche n° 5491 — February 2005 — 11 pages

Abstract: In this paper, we improve the bound of complexity of the algorithms on polynomial ideals having complexities polynomial in d^n where d is the maximal degree of input polynomials and n the number of variables.

Instead of this bound, we present the more accurate bound $\max\{S, D^n\}$ where S is the size of the input polynomials in dense representation, and D is the arithmetic mean value of the degrees of input polynomials.

Key-words: Gröbner basis, complexity, polynomial system solving

Unité de recherche INRIA Rocquencourt

Une meilleure borne de complexité pour les bases de Gröbner des idéaux zéro–dimensionnels et la résolution des systèmes polynomiaux

Résumé : Dans cet article, nous améliorons la borne de complexité des algorithmes concernant les idéaux polynomiaux et connus pour être polynomiaux en d^n où d est le degré maximal des polynômes d'entrée et n le nombre des variables.

Nous remplaçons d^n par $\max\{S, D^n\}$, où S est la taille de l'entrée pour la représentation dense des polynômes et D la moyenne arithmétique des degrés des polynômes d'entrée.

Mots-clés : Base de Gröbner, complexité, résolution des systèmes polynomiaux

1 Introduction

In the literature, there are several algorithms on polynomial ideals which have a good complexity, i.e. which are polynomial in d^n where d is the maximal degree of input polynomials and n the number of variables. Some of these algorithms deal with polynomial system solving, for example [Laz81] which shows that one can solve the projectively zero–dimensional systems within this complexity. Also it was shown in [Lak90] and [LL91] that any zero–dimensional system can be solved within this complexity. Other good algorithms deal with the computation of Gröbner bases: Lazard [Laz83] has proved that the reduced Gröbner basis with respect to the degree reverse lexicographic ordering of either a projectively zero–dimensional ideal, or an ideal generated by a homogeneous regular sequence in generic coordinates, can be computed within this complexity. For these algorithms, the complexity which is polynomial in d^n is the *bit complexity* as well as the complexity in the number of field operations. Also, Lakshman [Lak91] has shown that any reduced Gröbner basis of a zero–dimensional ideal may be computed within this complexity in *the number of field operations*. For this, he uses a variant of [FGLM93] where an algorithm for transforming a Gröbner basis of a zero–dimensional ideal with respect to any ordering into a Gröbner basis with respect to another ordering is given, which has a similar complexity.

The aim of this paper is to show that in all these algorithms, we may replace d^n by $\max\{S, D^n\} < nh(eD)^n$ where S is the size of the input polynomials in dense representation (defined below), h is the maximal size of the coefficients of the input polynomials, D is the arithmetic mean value of the degrees of input polynomials and $e = 2.71828\cdots$ is the usual Euler constant.

The best which may be hoped for the complexity of an algorithm on polynomials ideals, is to be polynomial in the maximum of the sizes of the input and of the output. It is well known that Bézout's bound (the product of the degrees of the input polynomials) is generically a good measure of the size of the output. Therefore, the best which may be hoped is to be polynomial in $\max\{S, G^n\}$ where G is the geometric mean value of the degrees. Thus, our bound differs from the optimum as far as the arithmetic mean differs from the geometric mean.

Our bound is *exponentially better* than the previous ones: if the degree are not all equals, it is clear that $D^n \ll d^n$. Moreover, if most of polynomials are linear, D may be as close from 1 as one will; D^n may even be constant in n if the number of non linear polynomials remains constant when n increases. On the other hand, we have $S < nh(d+1)^n = O(nhd^n)$, but S is usually much lower. Especially, if $d \sim n$ we have $S = O(nh2^n) \ll nhd^n$.

Now, we give the structure of this paper. In Section 2, we give the general notations and the model of complexity which is used. In Section 3, we prove the combinatorial lemmas which are used in the next sections. Section 4 is devoted to the elimination of the linear polynomials from the input system and its complexity. In Section 5, we show that each step of the above algorithms has a complexity bounded by $\max\{S, D^n\}$. Finally, in Section 6, we show that both quantities S and D^n which appear in our bound of complexity are necessary, as each one may be the dominant one.

2 General notations and model of complexity

In this section, we precise the general notations and the model of complexity used through this paper.

We work with polynomials over a *computable field* K on which the linear algebra over K has a polynomial complexity. This is especially the case if K is finite or is the field of the rational numbers.

In all this paper, we will consider k polynomials $f_1, \dots, f_k \in K[x_1, \dots, x_n]$ or k homogeneous polynomials $F_1, \dots, F_k \in K[x_0, \dots, x_n]$. We denote by $d_1, \dots, d_k$ their total degrees, and if there are ℓ non linear polynomials, we suppose without lost of generality that $d_2 \geq \cdots \geq d_\ell \geq d_1 \geq 2 > d_{\ell+1} = \cdots = d_k = 1$. If $k < n$, we set $d_{k+1} = \cdots = d_n = 1$. Let $D = (d_1 + \cdots + d_n)/n$ and h be the maximal size of the coefficients of the f_i or of the F_i.

For bounding the complexity of the algorithms under consideration, we need to precise our measure of complexity.

We define the size of a polynomial f_i or F_i as its size in the *dense representation* i.e. in the representation where all the monomials of degree at most d_i (resp. d_i) in non–homogeneous case (resp. homogeneous case) are written, even if they have a zero coefficient, (see [vzGG03] p. 467 for example). Thus, the size of f_i or F_i is $nc\binom{n+d_i}{n}$ where c is the average size of the coefficients (the factor n comes from the representation of the vector of the exponents of the variables), and the size of the input S is the sum of the sizes of the f_i or of the F_i.

It is worthwhile to remark that the number k of input polynomials as well as the maximum size of their coefficients h are bounded by S and therefore need not to appear explicitly in the bounds of complexity.

Note also that, except for [Lak91] algorithm, we consider only the *bit complexity* which takes into account the size of the coefficients. The *arithmetic complexity* may easily be deduced by putting to 1 the size of the coefficients.

3 Combinatorial lemmas

In this section, we prove the combinatorial lemmas, which will be used in the next sections.

We use the well-known fact that the geometric mean is less than the arithmetic mean, i.e.

$$(d_1 \cdots d_n)^{1/n} \leq (d_1 + \cdots + d_n)/n$$

which implies that

Lemma 3.1 $d_1 \cdots d_n \leq D^n$.

The following lemma is the basis of the occurrence of D^n in our complexity bounds.

Lemma 3.2 *The number of monomials of degree at most $\delta = n(D-1)+1$ in n variables and the number of monomials of degree δ in $n+1$ variables are bounded above by $(eD)^n$ for D and $n \geq 1$.*

Proof The number of monomials of degree δ in $n+1$ variables as well as the number of monomials of degree $\leq \delta$ in n variables are equal to $\binom{n+\delta}{n}$ (see [CLO98] p. 106 for example). By definition of the binomial coefficients, we have

$$\binom{n+\delta}{n} = \binom{nD+1}{n} = \frac{D^n}{n!}\prod_{i=1}^{n}\left(n + \frac{2-i}{D}\right).$$

As $n + \frac{2-i}{D} \leq n$ for $i \geq 2$ and $\left(n - \frac{1}{D}\right)\left(n + \frac{1}{D}\right) < n^2$, we have for $n > 2$

$$\binom{n+\delta}{n} < \frac{n^n}{n!}D^n,$$

which implies that $\binom{n+\delta}{n} < (eD)^n$ by Stirling's formula. For $n \leq 2$ the result is easily proved directly. □

The following lemma is the key for eliminating the linear polynomials and keeping D^n in our bounds where $D \sim 1$.

Lemma 3.3 *With the notations of Section 2, suppose that* $m < n$ *and* $d_{m+1} = \cdots = d_n = 1$. *Let* $E = (d_1 + \cdots + d_m)/m$. *We have* $m(E-1) = n(D-1)$ *and*

$$E^m \leq D^n.$$

Proof The first assertion is immediate. The second follows from the following lemma. □

Lemma 3.4 *Let* m, n, E *and* D *be four positive real numbers such that* $1 \leq E < D$, $m < n$ *and* $m(E-1) = n(D-1)$. *Then*

$$E^m \leq D^n.$$

Proof Let $T = m(E-1) = n(D-1)$, and f be the function

$$f(t) = t\log\left(\frac{T+t}{t}\right).$$

Then, $E^m = e^{f(m)}$ and $D^n = e^{f(n)}$. We claim that the function f is increasing for $t > 0$: The second derivative $-\frac{T^2}{t(t+T)^2}$ of f is negative for $t > 0$. Thus, the first derivative f' of f is decreasing. We have also that $\lim_{t\to\infty} f'(t) = \lim_{t\to\infty}\left(\log(\frac{t+T}{t}) - \frac{T}{t+T}\right) = 0$. Thus, $f'(t) > 0$ for $t > 0$, which implies that f is an increasing function for $t > 0$. Therefore,

$$E^m = e^{f(m)} \leq e^{f(n)} = D^n.$$

□

4 Eliminating the linear polynomials

We will see that the complexity of the algorithms that we consider is polynomial in the quantity $(eD)^n$ introduced in Lemma 3.2. If $D \geq 2$ this is also polynomial in D^n, as $eD < D^{2.5}$, but this is not true if D is close to 1. If $D < 2$ we are sure that some polynomials are linear. Using them to eliminate some variables decreases the complexity (by Lemma 3.3), but we have to take into account the complexity of this elimination.

Proposition 4.1 *Let $I = \langle f_1, \ldots, f_k \rangle$ be an ideal in the ring $K[x_1, \ldots, x_n]$ and $<$ be a monomial ordering. Suppose that $f_{\ell+1}, \ldots, f_k$ are linear polynomials for some $\ell < n$. Let G_1 be the reduced Gröbner basis of $\langle f_{\ell+1}, \ldots, f_k \rangle$ and $\bar{f}_i = normalform(f_i, G_1)$ for $i = 1, \ldots, \ell$ (cf. [CLO97]). Let G_2 be the reduced Gröbner basis of $\langle \bar{f}_1, \ldots, \bar{f}_\ell \rangle$.*

The leading terms in G_1 are single variables. Denote them by $\{Y_{m+1}, \ldots, Y_n\} \subset \{x_1, \ldots, x_n\}$, and let $Y_1, \ldots, Y_m$ be the remaining variables. We have:

(1) *$\bar{f}_1, \ldots, \bar{f}_\ell$ depend only on $Y_1, \ldots, Y_m$;*

(2) *$G_1 \cup G_2$ is the reduced Gröbner basis of I with respect to $<$;*

(3) *any common zero $y_1, \ldots, y_n$ of the f_i's is deduced from a common zero $y_1, \ldots, y_m$ of $\bar{f}_1, \ldots, \bar{f}_\ell$ by expressing $y_{m+1}, \ldots, y_n$ as linear functions of $y_1, \ldots, y_m$, using G_1.*

Proof It is well known [Laz83] that for linear polynomials, Gröbner basis computation is equivalent with Gaussian reduction, and that in this case the normal form computation is equivalent with the linear elimination of the leading variables.

To prove the second assertion, it suffices to prove that any $f \in I$ reduces to 0 by $G_1 \cup G_2$. Reducing f by G_1 we get $\bar{f}$ which belongs to $\bar{f}_1, \ldots, \bar{f}_\ell$ and thus reduces to 0 by G_2.

The third assertion comes from the facts that G_2 depends only on $Y_1, \ldots, Y_m$ (by the first item), and that $Y_{m+1}, \ldots, Y_n$ are the linear functions of $Y_1, \ldots, Y_m$ (because $Y_{m+1}, \ldots, Y_n$ are the leading terms of G_1). □

Proposition 4.2 *In the preceding proposition:*

(1) *the complexity of the computation of G_1 is polynomial in cn where c is the maximal size of the coefficients of $f_{\ell+1}, \ldots, f_k$;*

(2) *the complexity of the computation of $\bar{f}_i$ for $i = 1, \ldots, \ell$ is polynomial in c (above coefficient size) and the size of f_i;*

(3) *for each solution, the complexity of the computation of $y_{m+1}, \ldots, y_n$ is polynomial in n, c and the size of $(y_1, \ldots, y_m)$.*

Proof Assertion (1) follows from the equivalence between the computation of G_1 and Gaussian elimination.

To prove the second assertion, let Γ be a matrix whose rows are indexed by the monomials of degree d_i and whose columns are the representation of f_i and of all products of the elements of G_1 by the monomials of degree $d_i - 1$. Then, a Gaussian elimination on the columns of Γ eliminates $Y_{m+1}, \ldots, Y_n$ from f_i, and gives the coefficients of $\bar{f}_i$. The number of columns of Γ is bounded by $1 + n\binom{n+d_i-1}{n} < \binom{n+d_i}{n}$, the number of rows of Γ. Also, the coefficients of G_1 have a size which is polynomial in cn. Thus, the size of Γ is bounded by cn times the square of the size of f_i which shows the assertion (2).

Assertion (3) follows also from the complexity of the linear algebra. □

5 Arithmetic mean versus maximum of the degrees

In this section, we state precisely the complexity bound resulting of the previous results.

5.1 Bézout bounded complexities

Most algorithms dealing with zero–dimensional ideals split in several steps. Some of them have their complexity bounded in term of the degree of the ideal, i.e. the dimension of the vector space of the quotient of the ring by the ideal. In fact, these steps consist mainly in linear algebra in this vector space. This is the case of the algorithm of [FGLM93] for changing of monomial ordering and of the reconstruction of the Gröbner basis of the ideal from those of its associated prime ideals in [Lak91] (note that this algorithm is a variant of [FGLM93]).

As Bézout's theorem asserts that the degree of the ideal is bounded by $d_1 \cdots d_n$, Lemma 3.1 and the bounds of complexity given in [FGLM93] and [Lak91] imply immediately the following. For this, let h be the maximal size of the coefficients of the input polynomials of the algorithm of [FGLM93] or of [Lak91].

Theorem 5.1 *With the notations of Section 2, the complexity of the algorithm of [FGLM93] for transforming a Gröbner basis of a zero–dimensional ideal with respect to an ordering into a Gröbner basis with respect to another ordering, and of [Lak91] for computing a Gröbner basis of a zero–dimensional ideal from Gröbner bases of its associated prime ideals is polynomial in nhD^n.*

5.2 Macaulay bounded complexities

In the algorithms under consideration, the critical step consists in linear algebra on polynomials having the Macaulay's bound $\sum_{i=1}^{n}(d_i - 1) + 1$ as degree. Since the paper [Laz81] the size of the matrices, which are involved, is usually bounded by d^n where $d = d_2$ is the maximum of the degrees of the input polynomials. The results of Section 3 show immediately that $(eD)^n$ is a sharper bound.

If $D \geq 2$ we have $e < D^{1.5}$, which induce a complexity which is polynomial in D^n. However, if D is close to 1 the factor e^n may not be included in D^n. Fortunately, if $D < 2$ some of the input polynomials are linear and we may use them to reduce the number of variables. The results of Section 4 show immediately that the complexity of the elimination is polynomial in the size of the input polynomials, S while the linear algebra after elimination has a complexity which is polynomial in D^n by Lemma 3.2 and 3.3.

More precisely, we have:

Theorem 5.2 *With the notations of Section 2, the bit complexities of the following algorithms are polynomial in*

$$\max\{S, D^n\}.$$

(1) *Algorithms of [Laz81] for solving a zero–dimensional homogeneous system and of [Laz83] for computing the reduced Gröbner basis of a zero–dimensional homogeneous ideal or of an homogeneous ideal generated by a regular sequence in generic coordinates.*

(2) *Algorithm of [LL91] for solving a non–homogeneous zero–dimensional system, and for computing the reduced Gröbner bases of the associated prime ideals of a zero–dimensional ideal.*

Similarly the algorithm of [Lak91], for computing any reduced Gröbner basis of a zero–dimensional ideal, needs a number of fields operations which is bounded above by $\max\{S, D^n\}$.

Proof These algorithms split in different steps. We estimate first the complexity of the steps which are common to all of them before considering each algorithm separately.

The first step consists in eliminating the linear polynomial by Proposition 4.1. This step has a complexity polynomial in S and produces polynomials whose coefficients size is polynomial in $nh < S$ (see Proposition 4.2). Especially, for polynomials with integers coefficients this size is bounded by $n(h + 1/2 \log n)$ by Hadamard inequality.

With the notations of Proposition 4.1, let $f_1, \dots, f_k$ (resp. $F_1, \dots, F_k$ in homogeneous case) be the input polynomials, let $Y_1, \dots, Y_m$ be the variables which remain after the elimination, and let $\bar{f}_1, \dots, \bar{f}_\ell$ (resp. $\bar{F}_1, \dots, \bar{F}_\ell$) be the normal forms of the non linear input polynomials with respect to the reduced Gröbner basis of the linear input polynomials. By item (2) of Proposition 4.2 the complexity of the computation of these normal forms is polynomial in S and the size of the resulting coefficients is also polynomial in S.

By Proposition 4.1, solving the initial problem (Gröbner basis computation or solving the system) is reduced to a similar problem on $\bar{f}_1, \dots, \bar{f}_\ell$ (resp. $\bar{F}_1, \dots, \bar{F}_\ell$ in homogeneous case), where the size of the coefficients is polynomial in S.

As all of the algorithms under consideration use the *Sylvester* matrix of a homogeneous ideal in m variables in degree $\sum_{i=1}^{n}(d_i - 1) + 1$, we recall its definition here. This is a matrix whose rows are indexed by the monomials of this degree in m variables and whose columns are the representation of the generators of the ideal. If $E \geq 2$ is the arithmetic mean of $d_1, \dots, d_m$ then the size of the Sylvester matrix is bounded by $(eE)^m \leq E^{2.5m} < D^{2.5n}$ by Lemmas 3.2 and 3.3.

In the second step of the algorithm of [Laz81], the author has considered the Sylvester matrix in the variables $Y_1, \dots, Y_m$ in degree $\sum_{i=1}^{n}(d_i - 1) + 1$ for the ideal $\langle \bar{F}_1, \dots, \bar{F}_\ell, L\rangle$ in which $L = U_1Y_1 + \cdots + U_mY_m$ where the U_i's are new indeterminates.

As it was shown in [Laz81], a Gaussian elimination on this matrix allows to solve the system. The size of this matrix is polynomial in D^n by Lemma 3.2, and its coefficients size is polynomial in S. Therefore, this Gaussian elimination as well as the whole algorithm have a bit complexity polynomial in $\max\{S, D^n\}$ and the resulting coefficients are similarly bounded.

For computing the reduced Gröbner basis of $\langle \bar{F}_1, \dots, \bar{F}_\ell \rangle$ for the degree reverse lexicographic ordering by the algorithm of [Laz83], the second step consists in a Gaussian elimination on the Sylvester matrix in degree $\sum_{i=1}^{n} (d_i - 1) + 1$ of this ideal. Thus the complexity of this step as well as the coefficients size of the result are polynomial in $\max\{S, D^n\}$. For the other monomial orderings (zero–dimensional case) this step is followed by the algorithm of [FGLM93] which is again a linear algebra computation on matrices of a similar size.

Now, consider the algorithm of [LL91] for solving the system $\bar{f}_1 = \cdots = \bar{f}_\ell = 0$. Following its first step, we may suppose that $\bar{f}_1, \dots, \bar{f}_m$ is a regular sequence. The second step consists in homogenizing and deforming these polynomials. More precisely, the authors consider $G_i = F'_i + s x_i^{d_i}$ where F'_i is the homogenization of $\bar{f}_i$ with respect to x_0, and s is a new indeterminate. The third step consists in applying [Laz81] algorithm over $K(s)$ for computing matrices $M_0, \dots, M_m$ with coefficients in $K(s)$. This step and the output size are also bounded by $\max\{S, D^n\}$. The fourth step consists in a linear change of variables which amounts to replace the M_i by linear combination of them. The next step consists in computing the characteristic polynomials of $2n$ quotients of these matrices, followed by gcd computations on these polynomials in order to provide the solutions as polynomial function of the roots of a univariate polynomial (RUR or shape lemma representation). If there were more than n input polynomials, the remaining ones would be introduced at this step in a way similar to [FGLM93].

Thus, the factorization of the univariate polynomials and [FGLM93] algorithm allow to compute the Gröbner bases of the radical as well as of the associated primes. All these steps involve linear algebra, gcd computation or factorization of univariate polynomials over data whose size is polynomial in $\max\{S, D^n\}$. It follows that the whole algorithm has the asserted complexity.

Consider finally the algorithm of [Lak91] for computing the reduced Gröbner basis of the zero–dimensional ideal $\bar{I} = \langle \bar{f}_1, \dots, \bar{f}_\ell \rangle \subset K[Y_1, \dots, Y_m]$. It works in two steps. In the first step, the output of [LL91] algorithm is used for computing the Gröbner bases of the primary components of $\bar{I}$. In the second step, they are put together to obtain the reduced Gröbner basis of $\bar{I}$. It was shown in this paper that the number of field operations for the first step is polynomial in S and $m\delta_i$ where δ_i is the degree of the primary component under consideration. As $\sum \delta_i$ is the degree of the ideal, it is lower than the Bézout bound. Thus, the complexity of this step is bounded, as before. The same is true for the last step, because it is a variant of [FGLM93]. □

Remark 1 It could be astonishing that the best known bound of binary complexity is not the same to solve a zero–dimensional system and to compute the corresponding Gröbner basis. We are able to solve this problem by producing an algorithm for computing the Gröbner basis of a non–homogeneous zero–dimensional system with a bit complexity which is polynomial in $\max\{S, D^n\}$. This is the object of another paper, in preparation.

6 Discussion

In this section, we show both quantities which appear in our bound of complexity may be the dominant one.

Recalling that our bound of complexity is polynomial in $\max\{S, D^n\}$, we have to compare these two quantities. For this purpose, it is worthwhile to suppose that the size of coefficients is small and therefore to set $h = 1$.

The dominant factor of this complexity bound may be either S or D^n, even if usually the latter is much larger than the former, especially when $D \geq 2$.

In fact, if $D \geq 2$, we have $d_i \leq nD - n + 1$ for any i, then $S \leq n\binom{n+nD-n+1}{n}$ which is less than $n(eD)^n$ by Lemma 3.2. But the latter is less than D^{3n} because we have $e < D^{1.45}$ and

$$n \leq 2^{0.55n} \leq D^{0.55n}$$

for any positive integer n. This shows that in this case, D^n is the dominant factor.

If moreover, the d_i are all equal to n, then $\frac{D^n}{S} \geq \frac{n^n}{n2^ne^n}$ increases more than exponentially with n, and $S \ll D^n$ in this case.

Finally, let us consider a family indexed by n of systems of c polynomials of degree d in n variables and $n - c$ linear polynomials. We have $D^n \leq (1 + cd/n)^n < e^{cd}$ which is independent from n while S increases exponentially with n. Thus $D^n \ll S$ in this case.

References

[CLO97] David Cox, John Little, and Donal O'Shea. *Ideals, varieties, and algorithms.* Undergraduate Texts in Mathematics. Springer-Verlag, New York, second edition, 1997. An introduction to computational algebraic geometry and commutative algebra.

[CLO98] David Cox, John Little, and Donal O'Shea. *Using algebraic geometry*, volume 185 of *Graduate Texts in Mathematics.* Springer-Verlag, New York, 1998.

[FGLM93] J. C. Faugère, P. Gianni, D. Lazard, and T. Mora. Efficient computation of zero-dimensional Gröbner bases by change of ordering. *J. Symbolic Comput.*, 16(4):329–344, 1993.

[Lak90] Y. N. Lakshman. On the complexity of computing a Gröbner basis for the radical of a zero dimensional ideal. In *Proc. of 22nd ACM Symposium on Theory of computing (STOC)*, pages 555–563. 1990.

[Lak91] Y. N. Lakshman. A single exponential bound on the complexity of computing Gröbner bases of zero-dimensional ideals. In *Effective methods in algebraic geometry (Castiglioncello, 1990)*, volume 94 of *Progr. Math.*, pages 227–234. Birkhäuser Boston, Boston, MA, 1991.

[Laz81] Daniel Lazard. Résolution des systèmes d'équations algébriques. *Theoret. Comput. Sci.*, 15(1):77–110, 1981.

[Laz83] D. Lazard. Gröbner bases, Gaussian elimination and resolution of systems of algebraic equations. In *Computer algebra (London, 1983)*, volume 162 of *Lecture Notes in Comput. Sci.*, pages 146–156. Springer, Berlin, 1983.

[LL91] Y. N. Lakshman and Daniel Lazard. On the complexity of zero-dimensional algebraic systems. In *Effective methods in algebraic geometry (Castiglioncello, 1990)*, volume 94 of *Progr. Math.*, pages 217–225. Birkhäuser Boston, Boston, MA, 1991.

[vzGG03] Joachim von zur Gathen and Jürgen Gerhard. *Modern computer algebra.* Cambridge University Press, Cambridge, second edition, 2003.

Unité de recherche INRIA Rocquencourt
Domaine de Voluceau - Rocquencourt - BP 105 - 78153 Le Chesnay Cedex (France)

Unité de recherche INRIA Futurs : Parc Club Orsay Université - ZAC des Vignes
4, rue Jacques Monod - 91893 ORSAY Cedex (France)
Unité de recherche INRIA Lorraine : LORIA, Technopôle de Nancy-Brabois - Campus scientifique
615, rue du Jardin Botanique - BP 101 - 54602 Villers-lès-Nancy Cedex (France)
Unité de recherche INRIA Rennes : IRISA, Campus universitaire de Beaulieu - 35042 Rennes Cedex (France)
Unité de recherche INRIA Rhône-Alpes : 655, avenue de l'Europe - 38334 Montbonnot Saint-Ismier (France)
Unité de recherche INRIA Sophia Antipolis : 2004, route des Lucioles - BP 93 - 06902 Sophia Antipolis Cedex (France)

Éditeur
INRIA - Domaine de Voluceau - Rocquencourt, BP 105 - 78153 Le Chesnay Cedex (France)
http://www.inria.fr
ISSN 0249-6399