INSTITUT NATIONAL DE RECHERCHE EN INFORMATIQUE ET EN AUTOMATIQUE

Real Solving for positive dimensional systems

Ph. Aubry — F. Rouillier — M. Safey El Din

N° 3992

Septembre 2000

THÈME 2

Rapport de recherche

ISSN 0249-6399 ISRN INRIA/RR--3992--FR+ENG

Real Solving for positive dimensional systems

Ph. Aubry [*] , F. Rouillier [†] , M. Safey El Din [‡]

Thème 2 — Génie logiciel
et calcul symbolique
Projets PolKA

Rapport de recherche n° 3992 — Septembre 2000 — 20 pages

Abstract: Finding one point on each semi-algebraically connected component of a real algebraic variety, or at least deciding if such a variety is empty or not, is a fundamental problem of computational real algebraic geometry. Even though numerous studies have been done on the subject, only a few number of efficient implementations exists. In this paper, we propose a new efficient and practical algorithm for computing such points. By studying the critical points of the restriction to the variety of the distance function to one well chosen point, we show how to provide a set of zero-dimensional systems whose zeroes contain at least one point on each semi-algebraically connected component of the studied variety, without any assumption neither on the variety (smoothness or compactness for example) nor on the system of equations that define it. Once such a result is computed, one can then apply, for each computed zero-dimensional system, any symbolic or numerical algorithm for counting or approximating the solutions. We have made experiments using a set of pure exact methods. The practical efficiency of our method is due to the fact that we do not apply any infinitesimal deformations, conversely to the existing methods based on similar strategy.

Key-words: Real roots, real algebraic variety, Connected components, Polynomial systems, Grobner bases, Triangular sets

[*] Équipe de calcul formel, Laboratoire d'Informatique de Paris 6
[†] Projet PolKA, LORIA, INRIA, Nancy
[‡] Équipe de calcul formel, Laboratoire d'Informatique de Paris 6

Unité de recherche INRIA Lorraine
LORIA, Technopôle de Nancy-Brabois, Campus scientifique,
615, rue du Jardin Botanique, BP 101, 54602 Villers-Lès-Nancy (France)
Téléphone : +33 3 83 59 30 00 — Télécopie : +33 3 83 27 83 19

Résolution réelle des systèmes polynomiaux en dimension positive

Résumé : Trouver au moins un point par composante semi-algébriquement connexe d'une variété algébrique réelle, ou décider du vide d'une telle variété est un des problèmes algorithmiques fondamentaux de la géométrie algébriques réelle effective. Malgré le nombre important d'études existantes sur ce sujet, il existe très peu d'implantations efficaces résolvant ce problème de manière satisfaisante. Dans cet article, nous proposons un nouvel algorithme, efficace en pratique qui calcule au moins un point par composante connexe d'une variété algébrique réelle. En étudiant les points critiques de la restriction de la fonction distance à une telle variété, on montre comment calculer des systèmes zéro-dimensionnels dont les solutions contiennent au moins un point par composante connexe de la variété que l'on veut étudier, sans faire la moindre hypothèse de compacité, de régularité, ou de généricité. Une fois ces systèmes zéro-dimensionnels obtenus, on peut alors leur appliquer un algorithme (symbolique ou numérique) pour compter et isoler leurs solutions réelles. Dans nos tests, nous n'utilisons que des méthodes symboliques. L'efficacité de notre méthode tient dans le fait que nous ne faisons aucune déformation infinitésimale, contrairement aux autres méthodes basées sur des stratégies similaires.

Mots-clés : Racines réelles, variété algébrique réelle, Composantes connexes, Systèmes polynomiaux, Bases de Grobner, Ensembles triangulaires

1 Introduction

The problem of finding one point on each semi-algebraically connected component of a real algebraic variety V, or at least deciding if V is empty, appears in several problems in computational algebraic geometry.

The most popular algorithm which solves this problem is Collins' Cylindrical Algebraic Decomposition (see [11]). This algorithm is based on variable elimination, one after the other, and decides the truth of a first order formula. Thus, it solves more general problems than the one in which we are interested. Note also that it is polynomial in the degree and the number of polynomials and doubly exponential in the number of variables. In practice, this theoretical complexity can be observed for many examples, and so, the problem size which can be solved with the CAD algorithm and its variants is limited.

In [17], Grigoriev and Vorobjov propose an algorithm for deciding the emptiness of a semi-algebraic set with a single exponential complexity in the number of variables. In this method as well as in most of its variants (see [25, 10, 19, 6, 6, 7, 30]), the key idea is to apply deformations so that the projection critical points with respect to one coordinate define a finite set that meets every semi-algebraic connected component of the deformed variety. In [6, 7, 30] the authors take, in addition, sums of squares in order to work with smooth and compact real algebraic sets defined by a unique polynomial equation. The final result is then obtained by taking the limits of the points when the infinitesimals tend to zero.

In the previous methods, the problem is reduced to the resolution of zero-dimensional systems. But even though the transformations keep a good theoretical complexity (see [6]), the use of at least two infinitesimals (deformations) and a degree growth (sum of squares) prevent these algorithms from being efficient in practice.

In [5], the authors provide an algorithm, based on straight-line programs, with a good theoretical complexity, when the variety V is smooth, compact and given by a regular sequence of polynomials, so that, in practice, one would have to face, at least, the same problems than in [6] (smoothness and compactness) for providing an algorithm that works in every situation.

An algorithm for deciding the emptiness of semi-algebraic sets is presented in [12] and leaded to better practical results. It avoids taking the sums of the squares of the equations, and deals with the singularities by using the fact that the singular locus of V is a sub-variety of V with strictly smaller dimension following [8]. Nevertheless, the authors keep on using the projection function. Thus, their algorithm requires at least one infinitesimal deformation or the introduction of a new variable for dealing with non compact varieties.

The particular case of a variety defined by a single equation is studied in [28]. Following a classical idea of Seidenberg [32], the algorithm computes the critical points of the distance function to a point instead of the critical points of coordinates functions. The authors recall that the set formed by the critical points of the distance function to a point meets each connected component of V. They show that this set is finite when the point is well chosen (they give a strategy for choosing it) and V has at most a finite number of singularities, so that an infinitesimal deformation is needed only when the variety has an infinite number of singular points.

In this paper, we also compute the critical points of the distance function to some point A, but our algorithms handle systems of several polynomial equations and need not taking the sum of the squares of the polynomials. In our approach, the case of hypersurfaces defined by a unique equation thus becomes only a particular case, and is not considered as a favorable case to reach by various tricks.

Like in [28], we define an algebraic set $\mathcal{C}(V, A)$ that contains these critical points and a sub-algebraic variety of V. Moreover, it meets every semi-algebraically connected component of V. Our main result consists in proving that a good point A may be chosen so that $\mathcal{C}(V, A)$ is the disjoint union of a finite set of points and a sub-algebraic variety W of V with smaller dimension than V. We are thus leaded to compute the isolated points of $\mathcal{C}(V, A)$ and to study, in the same way, the sub-variety W. We therefore obtain an algorithm without any infinitesimal deformation whose proof is simply based on the fact that the dimension of the studied varieties strictly decreases at each step.

Let us recall that [18] concludes "theoritical analyses based on the big O notation are too coarse for comparing decision algorithms over the reals", after showing that the methods of [17] and [25] are not usable in practice. This remark is enforced by some experiments (see [31]) about the more recent algorithms given in [7, 20, 30]. Our first algorithm (Section 2), designed only for an overview of the method and its generality, does not improve the single exponential complexity of some previous methods [6, 7, 17, 19, 20, 25, 30]. We do not know the complexity for computing the used equi-dimensional decompositions of an ideal and consequently of the theoritical complexity of our main algorithm (Algorithm 3 in Section 3). Our goal is to obtain an algorithm for real solving of positive dimensional algebraic systems that provides efficiently in practice at least one point by semi-algebraically connected component.

The paper is structured as follows. Section 2 is devoted to the definition and the study of the algebraic set $\mathcal{C}(V, A)$ mentioned above. We give the explicit construction of a set of zero-dimensional systems whose real roots meet every semi-algebraic connected component of $V \bigcap R^n$. Section 3 points out and solves the limitations (number of determinants) of this generic algorithm (Algorithm 1). We show how to use the theory of triangular sets to optimize the computations (Algorithm 2). Section 4 studies some improvements in particular cases (such as Noether position). Section 5 presents some practical experiments which illustrate the practical behaviour of our algorithms. It shows the interest of our approach and justifies our choices.

Acknowledgments : We would like to thank J.-C. Faugère, D. Lazard and M.-F. Roy for their helpful comments, advises and supports and H. Hong who provided us the CAD implementation used for the tests.

2 The basic idea

In the whole paper, K is an ordered field, R is its real closure and C its algebraic closure. For $S \subset K[X_1, \ldots, X_n]$, we denote by $V(S)$ the K-variety formed by the zeros of S in C^n and by $\langle S \rangle$ the ideal generated by S in $K[X_1, \ldots, X_n]$.

Let V be a K-variety. On a computational point of view, V is actually known through a finite set of polynomials $\{P_1, \ldots, P_s\}$ such that $V = V(P_1, \ldots, P_s)$. In this section, our purpose is to avoid strong requirements on these polynomials and to design an algorithm as general as possible. We are interested in the critical points of the distance function to some point A which define at least one point on each semi-algebraically connected component of $V \bigcap R^n$. The set of critical points is finite when A is well chosen. Therefore, we naturally intend to compute a zero-dimensional system whose zeros contain these critical points. However, the critical points may not be represented simply by an algebraic system. We thus introduce a subvariety of V, denoted by $\mathcal{C}(V, A)$, that contains these points.

Notation 2.1 *Let $S = \{P_1, \ldots, P_s\}$ be a set of polynomials in $K[X_1, \ldots, X_n]$ such that $V = V(S)$ is a variety of dimension d. Given any point $A \in C^n$, we define the following algebraic set :*

$$\mathcal{C}(V, A) \;=\; \{M \in V, rank(\overrightarrow{grad}_M(P_1), \ldots, \overrightarrow{grad}_M(P_s), \overrightarrow{AM}) \leq n - d\}.$$

Building the set $\mathcal{C}(V, A)$ is interesting only if it is smaller than V and if it intersects each semi-algebraically connected components. We therefore want its dimension to be strictly inferior to d when $d > 0$. It is not generally true when the ideal $\langle P_1, \ldots, P_s \rangle$ is not assumed to be radical as in the following example :

Let $n = 2$ and $V = V(X_2^2)$. For any point $A = (a_1, a_2)$, the set $\mathcal{C}(V, A)$ is formed by the points of V such that $2X_1X_2 = 0$, and thus $\mathcal{C}(V, A) = V$.

A similar problem may be caused by a bad choice of the point A. This is illustrated by the simple example where V is a sphere and A is its center. But assuming that $\langle P_1, \ldots, P_s \rangle$ is radical, it becomes possible to detect such a bad choice, and we show in Theorem 2.3 that a good point may be found after a finite number of tests.

Another drawback appears with general radical ideals. When the irreducible components of V have different dimensions, the critical points of the distance function that do not lie on a component of dimension d may not be in $\mathcal{C}(V, A)$. It is the case, in particular, for any nonsingular point M of V that lies on such a component since $rank(\overrightarrow{grad}_M(P_1), \ldots, \overrightarrow{grad}_M(P_s), \overrightarrow{AM}) > n - d$.

For instance, let V be the variety defined by the radical ideal generated by $P_1 = (X_1^2 + X_2^2 - 1)(X_1 - 2)$ and $P_2 = (X_1 - 2)X_3$. The set $V \cap \mathbb{R}$ is the union of the plane with equation $X_1 = 2$ and the circle defined by the equations $X_1^2 + X_2^2 - 1 = 0$ and $X_3 = 0$. Every point of the circle is nonsingular and satisfies

$$rank\left(\begin{bmatrix} 3\,X_1{}^2 - 4\,X_1 + X_2{}^2 - 1 & 2\,X_2\,X_1 - 4\,X_2 & 0 \\ X_3 & 0 & X_1 - 2 \end{bmatrix}\right) = 2\,.$$

It is obvious if $X_2 \neq 0$ since $X_1 \neq 2$.And if $X_2 = 0$ then $3X_1^2 - 4X_1 + X_2^2 - 1 = 2 - 4X_1$ cannot vanish. Hence, $\mathcal{C}(V, A)$ does not meet every semi-algebraically connected component of $V \cap \mathbb{R}$.

The critical points of the distance function to A lies on $\mathcal{C}(V, A)$ but what about the other points in $\mathcal{C}(V, A)$? Let $M \in V$. If the dimension of the vector space generated by $\overrightarrow{grad}_M(P_1), \ldots, \overrightarrow{grad}_M(P_s)$ is strictly smaller than $n - d$, then $M \in \mathcal{C}(V, A)$. This happens when M is a singular point of an irreducible component of V with dimension d.

But since V is not necessary supposed to be irreducible, the subvariety $\mathcal{C}(V, A)$ contains also each point of V that lies on an irreducible component of dimension $d' < d$. We thus introduce the following notation, which defines a set that may contains strictly the singular locus of V ([13]) :

Notation 2.2 *Let V be an algebraic variety of dimension d and $\{P_1, \ldots, P_s\} \subset K[X_1, \ldots, X_n]$ such that $I(V) = \langle P_1, \ldots, P_s \rangle$. We denote by* $\mathrm{Sing}(V)$ *the variety*

$$\mathrm{Sing}(V) \;=\; \{M \in V \mid \mathrm{rank}(\overrightarrow{grad}_M(P_1), \ldots, \overrightarrow{grad}_M(P_s)) < n - d\} \;.$$

Theorem 2.3 *Let V be an* ***equidimensional*** *algebraic variety of dimension $d > 0$ and $\{P_1, \ldots, P_s\} \subset K[X_1, \ldots, X_n]$ such that $I(V) = \langle P_1, \ldots, P_s \rangle$.*
If D is a positive integer large enough, there exists at least one point A in $\{1 \ldots D\}^n$ such that :

1. *$\mathcal{C}(V, A)$ meets every semi-algebraically connected component of $V \bigcap R^n$,*
2. *$\mathcal{C}(V, A) = \mathrm{Sing}(V) \bigcup V_0$, where V_0 is a finite set of points in C^n.*

Moreover, $dim(\mathcal{C}(V, A)) < dim(V)$.

Proof : Let A be any point in C^n and $\mathcal{D}$ be a semi-algebraically connected component of $V \bigcap R^n$. There exists $M \in \mathcal{D}$ such that M is at minimal distance from A. If $M \in \mathrm{Sing}(V)$ then M is obviously in $\mathcal{C}(V, A)$. Now, suppose that $M \notin \mathrm{Sing}(V)$ and let $\mathcal{S}(A, r)$ be the sphere of center A and radius $r = d(A, M)$. Since M is at minimal distance to A, the varieties $\mathcal{S}$ and V are tangent at M and then $\overrightarrow{AM} \in \mathrm{Vect}(\overrightarrow{grad}_M(P_1), \ldots, \overrightarrow{grad}_M(P_s))$. Since V is equi-dimensional, it follows that $M \in \mathcal{C}(V, A)$ and assertion 1 is proved.

Let $Q_1, \ldots, Q_n$ be polynomials in $K[X_1, \ldots, X_n, \lambda_1, \ldots, \lambda_s]$ defined by $Q_j = \sum_{i=1,\ldots,s} \lambda_i \frac{\partial P_i}{\partial X_j} - X_j$, and let $\mathcal{H}$ be the subset of C^{n+s} defined by $\mathcal{H} = \{(M, \lambda_1, \ldots, \lambda_s) \in C^{n+s} \mid M \in V \setminus \mathrm{Sing}(V)\}$.
Consider the application

$$\begin{array}{cccc} F : & \mathcal{H} & \longrightarrow & C^n \\ & (M, \lambda_1, \ldots, \lambda_s) & \longmapsto & (Q_1(M, \lambda_1, \ldots, \lambda_s), \ldots, Q_n(M, \lambda_1, \ldots, \lambda_s)) \end{array}$$

If $Jac(P_1, \ldots, P_s, Q_1 + b_1, \ldots, Q_n + b_n)$ denotes the determinant of the Jacobian matrix associated to the polynomials $P_1, \ldots, P_s, Q_1 + b_1, \ldots, Q_n + b_n$, the critical values of F are

the points $B = (b_1, \dots, b_n)$ of C^n such that $V(Q_1 + b_1, \dots, Q_n + b_n, \mathrm{Jac}(P_1, \dots, P_s, Q_1 + b_1, \dots, Q_n + b_n)) \neq \emptyset$.
From Sard's theorem over C [24] and the transfer principle [9] it follows that

$$\mathcal{B} = \{B = (b_1, \dots, b_n) \in C^n \mid \\ \mathcal{H} \cap V(Q_1 + b_1, \dots, Q_n + b_n, \mathrm{Jac}(P_1, \dots, P_s, Q_1 + b_1, \dots, Q_n + b_n)) \neq \emptyset\}$$

is a constructible set of dimension $< n$ of C^n.
Since $\mathcal{B}$ is a constructible set of dimension $< n$, one can choose $A = (a_1, \dots, a_n) \in \{0, \dots, D\}^n$ with D large enough, and such that $A \notin \mathcal{B}$. In such case,

$$\mathcal{H} \cap V(Q_1 + a_1, \dots, Q_n + a_n, \mathrm{Jac}(P_1, \dots, P_s, Q_1 + a_1, \dots, Q_n + a_n)) = \emptyset$$

and thus the points of $\mathcal{H} \cap V(Q_1 + a_1, \dots, Q_n + a_n)$ are isolated and non singular. Let π be the projection defined by :

$$\begin{array}{cccc} \pi : & C^{n+s} & \longrightarrow & C^n \\ & (x_1, \dots, x_n, \ell_1, \dots, \ell_s) & \longmapsto & (x_1, \dots, x_n) \end{array}.$$

Since $\mathcal{C}(V, A) = \mathrm{Sing}(V) \cup \pi(\mathcal{H} \cap V(Q_1 + a_1, \dots, Q_n + a_n))$, $\mathcal{C}(V, A) = \mathrm{Sing}(V) \cup V_0$, where V_0 is finite set of points. From [13], $\mathrm{Sing}(V)$ is the union of algebraic varieties whose dimensions are strictly inferior to the dimension of V. ■

Remark 2.4 *From the proof of Theorem 2.3, a point A taken at random satisfies* $\dim(\mathcal{C}(V, A)) < \dim(V)$ *with a probability one.*

Suppose that $V \subset C^n$ and $A \in R^n$ fit the conditions of theorem 2.3 and that V_1 is a finite set of points that meets each semi-algebraically connected component of $\mathrm{Sing}(V) \bigcap R^n$. Since $\mathrm{Sing}(V) \subset \mathcal{C}(V, A)$, then $V_1 \bigcup V_0$ is a finite set of points that meets each semi-algebraically connected component of $\mathcal{C}(V, A) \bigcap R^n$. Since $\mathcal{C}(V, A)$ meets each semi-algebraically connected component of $V \bigcap R^n$ and $\mathcal{C}(V, A) \subset V$, then $V_1 \bigcup V_0$ meets each semi-algebraically connected component of $V \bigcap R^n$. The set V_1 can be obtained by applying theorem 2.3 to each equi-dimensional component of $\mathrm{Sing}(V)$.
The algorithm we propose consists in applying inductively the above process by performing at each step equi-dimensional decompositions of the intermediate varieties. At the end, we obtain a set of zero-dimensional components containing at least one point in each semi-algebraically connected component of $V \bigcap R^n$.

Notation 2.5 *For $B \in C^n$ and $\mathcal{Q} = \{Q_1, \dots Q_s\} \subset K[X_1, \dots, X_n]$, we define the matrix*

$$\mathcal{M}_B(\mathcal{Q}) = \left[\left[\frac{\partial Q_j}{\partial X_i} \right]_{(i=1\dots n, j=1\dots s)} \middle| \overrightarrow{BM} \right].$$

For $d \in \mathbb{N}$, $0 \leq d < n$, we denote by $\Delta_{B,d}(\mathcal{Q})$ the set of all the minors of order $(n - d + 1, n - d + 1)$ of the matrix $\mathcal{M}_B(\mathcal{Q})$.

According to the results above, the basic routines needed to implement an algorithm that computes this set of zero-dimensional components may be the following :

- **EquiDim** : takes as input a polynomial system S of equations and returns a list of polynomials $\mathcal{P}_d, \ldots, \mathcal{P}_0$ generating equi-dimensional radical ideals (for example Gröbner bases) such that $V(S) = V(\mathcal{P}_d) \bigcup \ldots \bigcup V(\mathcal{P}_0)$,
- **Dim** : takes as input a finite set of generators of an ideal and computes the dimension of the associated variety,
- **Minors** : takes as input a finite set of polynomials $\mathcal{Q}$, an integer d and a point $A \in C^n$ (in fact in K^n), and computes $\Delta_{A,d}(\mathcal{Q})$.

Algorithm 1

- **Input :** A polynomial system S of equations in $K[X_1, \ldots, X_n]$.
- **Output :** An empty list if $V(S) \bigcap R^n = \emptyset$, else a list of zero-dimensional systems whose roots contain at least one point in each semi-algebraically connected component of $V(S) \bigcap R^n$.

1. list := EquiDim(S), result := [],
2. Choose $A \notin V(S)$.
3. while list $\neq \emptyset$ do
   - S := first(list), and remove S from list, set $d := \mathrm{Dim}(S)$,
   - if $d = 0$ then result := result $\bigcup S$,
   - else
     - (*) $Q := \mathrm{Minors}(S, d, A) \bigcup S$ and set $u := \mathrm{Dim}(Q)$
     - if $u = d$ choose another point $A \notin V(S)$ and go to step (*).
     - $d := u$; list := list $\bigcup$ EquiDim(Q),
4. return result.

Note that the required subroutines of our algorithm are weaker than the ones of the algorithm described in [12] since we do not need to perform an irreducible decomposition.

3 Towards an efficient algorithm

Let $\mathcal{G} \subset K[X_1, \ldots, X_n$ be a Gröbner basis containing s polynomials and generating a radical equi-dimensional ideal of dimension d. According to the results above, the number of determinants which are computed by **Algorithm 1** is

$$\begin{pmatrix} s \\ n-d \end{pmatrix} \begin{pmatrix} n \\ n-d+1 \end{pmatrix}.$$

Hence such a combinatorial factor becomes limiting as soon as significant problems are considered. In this section, we show how to reduce the size and the number of deterimants we need to compute by using the properties of some specific polynomial triangular sets.

Let $\mathcal{G} \subset K[X_1,\ldots,X_n]$ be a **reduced lexicographical Gröbner basis** generating a radical equi-dimensional ideal of dimension d $K[X_1,\ldots,X_n]$ for the ordering $X_1 < \ldots < X_n$. For $p \in K[X_1,\ldots,X_n]$, we denote by $\mathrm{mvar}(p)$ (and we call main variable of p) the greatest variable appearing in p.

Definition 3.1 *[4, 23, 3] A set of polynomials $\mathcal{T} = (t_{d+1},\ldots,t_n) \subset K[X_1,\ldots,X_n]$ is a said to be a triangular set of polynomials if and only if*

$$\forall (t_i,t_j) \in \mathcal{T}\times\mathcal{T} \mid t_i \neq t_j \quad \mathrm{mvar}(t_i) \neq \mathrm{mvar}(t_j).$$

Let $\mathcal{T} = (t_{d+1},\ldots,t_n)$ be a set of polynomials extracted from $\mathcal{G}$ such that $\forall g \in \mathcal{G}$, $\exists i \in \{d+1,\ldots,n\}$ that satisfies ([23, 4, 3]) :

- (i) $\mathrm{mvar}(t_i) = \mathrm{mvar}(g)$,
- (ii) $\deg(t_i, \mathrm{mvar}(t_i)) \leq \deg(g, \mathrm{mvar}(t_i))$.
- (iii) for all such $g \in \mathcal{G}$ the leading monomial of g is inferior to the leading monomial of t_i for the lexicographical ordering.

The set $\mathcal{T}$ is obviously a triangular set of polynomials. In the following, we denote by **ExtractTriangular** a subroutine taking as input a reduced lexicographical Gröbner base and returning a triangular set as described above. We also denote by :

- h_i the leading coefficient of t_i (when it is seen as a univariate polynomial in its main variable) and $\mathcal{H}(\mathcal{T}) = \{h_{d+1},\ldots,h_n\}$.
- $W(\mathcal{T}) = \{M \in V(\mathcal{T}) \setminus V(\Pi_{i=d+1}^{n} h_i)\}$,
- $\mathrm{sat}(\mathcal{T}) = \{p \in K[X_1,\ldots,X_n] \mid \exists m \in N, \exists h \in \langle\mathcal{H}(\mathcal{T})\rangle, h^m p \in \langle\mathcal{T}\rangle\}$ the saturated ideal associated to the triangular set $\mathcal{T}$.

Without lost of generality, we suppose in the following that $\forall i \in \{d+1,\ldots,n\}$, $\mathrm{mvar}(t_i) = X_i$.

Definition 3.2 *[4, 23, 3]*

- *A triangular set $\mathcal{T} = (t_{d+1},\ldots,t_n) \subset K[X_1,\ldots,X_n]$ is said to be regular if and only if for all $i \in \{d+1,\ldots,n\}$, the initial h_i does not belong to any prime ideal associated to $\mathrm{sat}(t_{d+1},\ldots,t_{i-1}) \bigcap K[X_1,\ldots,X_{i-1}]$.*
- *A regular triangular set is said to be separable if and only if for all $i \in \{d+1,\ldots,n\}$, the partial derivative $\frac{\partial t_i}{\partial X_i}$ does not belong to any prime ideal associated to $\mathrm{sat}(t_{d+1},\ldots,t_i) \bigcap K[X_1,\ldots,X_i]$.*

In the following, we suppose that :

- the triangular set $\mathcal{T}$ extracted from $\mathcal{G}$ as described above is regular and separable,

- $\mathrm{sat}(\mathcal{T}) = \langle \mathcal{G} \rangle$.

Let $A = (a_1, \ldots a_n)$, $d = \dim(V(\mathcal{G}))$, and consider, for $j = 1 \ldots d$, the restricted list of minors of order $(n - d + 1)$ extracted from $\Delta_{A,d}(\mathcal{T})$:

$$\Gamma_A(\mathcal{T}) = \{det(\mathcal{M}_A^{(j)}), j = 1 \ldots d\}$$

where

$$\mathcal{M}_A^{(j)} = \left[\begin{array}{c|c} \left[\frac{\partial t_j}{\partial X_i}\right]_{j=d+1\ldots n} & X_i - a_i \\ \hline \mathcal{U}_{\mathcal{T}} = \left[\frac{\partial t_j}{\partial X_i}\right]_{j=d+1\ldots n}^{i=d+1\ldots n} & \begin{array}{c} X_{d+1} - a_{d+1} \\ \vdots \\ X_n - a_n \end{array} \end{array}\right]$$

Without lost of generality, we may suppose that $\mathrm{mvar}(t_i) = X_i$, so that the minors in $\Gamma_A(\mathcal{T})$ are easy to compute since $\mathcal{U}_{\mathcal{T}}$ is upper triangular. Our goal is now to show that we can replace, in our algorithm, the computation of $\Delta_{A,d}(\mathcal{G})$ by the computation of $\Gamma_A(\mathcal{T})$, and thus decrease the number and the cost of the computations:

Proposition 3.3 *Let us define* $\mathcal{D}(V(\mathcal{G}), A) = V(\mathcal{G}) \bigcap V(\Gamma_A(\mathcal{T}))$, $d = \dim(\mathcal{G})$ *and* $Sep(\mathcal{T}) = \prod_{i=d+1}^{n} \frac{\partial t_i}{\partial X_i}$. *If* $A \in C^n$ *such that* $\dim(\mathcal{C}(V(\mathcal{G}), A)) < \dim(V(\mathcal{G}))$, *then, according to the notations of theorem 2.3, we have :*

- $\mathcal{C}(V(\mathcal{G}), A) \subset \mathcal{D}(V(\mathcal{G}), A)$,
- $(\mathcal{D}(V(\mathcal{G}), A) \setminus V(Sep(\mathcal{T}))) \subset V_0$,
- $dim\ (\mathcal{D}(V(\mathcal{G}), A) \bigcap V(Sep(\mathcal{T}))) < \dim(V(\mathcal{G}))$.

In particular, $\dim(\mathcal{D}(V(\mathcal{G}), A)) < \dim(V(\mathcal{G}))$ *and* $\mathcal{D}(V(\mathcal{G}), A)$ *meets every semi-algebraically connected component of* $V(\mathcal{G})$.

Proof :

- Since $\mathcal{T} \subset \mathcal{G}$, $\Gamma_A(\mathcal{T}) \subset \Delta_{A,d}(\mathcal{T}) \subset \Delta_{A,d}(\mathcal{G})$, then :

  $$\mathcal{C}(V(\mathcal{G}), A) = V(\mathcal{G}) \bigcap V(\Delta_{A,d}(\mathcal{G})) \subset V(\mathcal{G}) \bigcap V(\Delta_{A,d}(\mathcal{T})) \subset V(\mathcal{G}) \bigcap V(\Gamma_A(\mathcal{T})).$$

- Let $M \in \mathcal{D}(V(\mathcal{G}), A) \setminus V(Sep(\mathcal{T}))$. We have $det(\mathcal{U}_{\mathcal{T}}(M)) \neq 0$ so that

  $$\mathrm{rank}(\overrightarrow{grad}_M(t_{d+1}), \ldots, \overrightarrow{grad}_M(t_n)) \geq n - d,$$

  and consequently $\mathrm{rank}(\overrightarrow{grad}_M(g_1), \ldots, \overrightarrow{grad}_M(g_s)) \geq n - d$. On one hand, each minor in $\Gamma_A(\mathcal{T})(M)$ equals 0, and so $\mathrm{rank}(\overrightarrow{grad}_M(t_{d+1}), \ldots, \overrightarrow{grad}_M(t_n), \overrightarrow{AM}) = n - d$. On the other hand, $dim(V(\mathcal{G})) = d$, so that $\mathrm{rank}(\overrightarrow{grad}_N(g_1), \ldots, \overrightarrow{grad}_N(g_s)) \leq n - d\ ,\ \forall N \in V(\mathcal{G})$ and thus $M \notin Sing(V(\mathcal{G}))$. Moreover, the vector spaces $Vect(\overrightarrow{grad}_M(g_1), \ldots, \overrightarrow{grad}_M(g_s))$ and $Vect(\overrightarrow{grad}_M(t_{d+1}), \ldots, \overrightarrow{grad}_M(t_n))$ coincide which shows that $M \in V_0 = \mathcal{C}(V(\mathcal{G}), A) \setminus Sing(V(\mathcal{G}))$.

- From Definition 3.2, $\dim(V(Sep(\mathcal{T}))\bigcap V(\mathcal{G})) < \dim(V(\mathcal{G}))$.

■

The full algorithm induced by Proposition 3.3 requires a stronger subroutine than an equi-dimensional decomposition subroutine. Indeed, it is not always possible to extract a regular separable triangular set $\mathcal{T}$ from a reduced lexicographical Gröbner basis $\mathcal{G}$ such that $\mathrm{sat}(\mathcal{T}) = \langle\mathcal{G}\rangle$ (consider for example the example $\langle xy, xz\rangle$ for $x < yz$). We denote by **Lex-TriSetEquiDim** : a subroutine taking as input a polynomial system of equations S and returning a set of reduced lexicographical Gröbner basis $\mathcal{G}_1, \ldots, \mathcal{G}_m$ generating radical equi-dimensional ideals and such that for all $i \in \{1, \ldots, m\}$

- $\mathcal{T}_i := \mathrm{ExtractTriangular}(\mathcal{G}_i)$ is a regular separable triangular set,
- $\mathrm{sat}(\mathcal{T}_i) = \langle\mathcal{G}_i\rangle$.

Such a decomposition can be obtained, for example, by first performing a Kalkbrener's decomposition into regular and separable triangular sets [21, 3] and then computing a Gr'obner base of the saturated ideals associated to each component.

We obtain the following algorithm :

Algorithm 2

- **Input :** A polynomial system S of equations in $K[X_1, \ldots, X_n]$.
- **Output :** A list of zero-dimensional systems whose roots contain at least one point in each semi-algebraically connected component of $V(S)\bigcap R^n$.

1. list := LexTriSetEquiDim(S), result := [],
2. Choose $A \notin V(S)$.
3. while list $\neq \emptyset$ do
   - S := first(list), and remove S from list, set $d := \mathrm{Dim}(S)$,
   - if $d = 0$ then result := result $\bigcup S$,
   - else
     - $\mathcal{T} := \mathit{ExtractTriangular}(S)$.
     - (*) $Q := \Gamma_A(\mathcal{T})\bigcup S$ and set $u := \mathrm{Dim}(Q)$
     - if $u = d$ choose another point $A \notin S$ and go to step (*).
     - $d := u$; list := list $\bigcup$ LexTriSetEquiDim(Q),
4. return result.

Remark 3.4 *Let $\mathcal{G}$ be a lexicographical reduced Gröbner base generating a prime ideal. In [4], the authors show that a regular triangular set $\mathcal{T}$ can always be extracted from $\mathcal{G}$ such that $\mathrm{sat}(\mathcal{T}) = \langle\mathcal{G}\rangle$. Thus, a prime decomposition can be performed instead of the* **LexTriSetEquiDim** *subroutine in the above algorithm. The advantages of a such strategy are that :*

- *computations are more splitted and the degree of the ideals that we study and the determinants we compute decrease,*
- *the study of singular points which lie on the intersection of irreducible components is avoided, and hence the computation times and the size of the output of the algorithm decrease.*

4 Optimizations in *generic* cases

Let $\mathcal{G} \subset K[X_1, \ldots, X_n]$ be a lexicographical reduced Gröbner basis and $\mathcal{T} = (t_{d+1}, \ldots, t_n)$ a regular separable triangular set extracted from $\mathcal{G}$. Without lost of generality, we assume in this section that $\mathrm{mvar}(t_i) = X_i$ for each $i \in \{d+1, \ldots, n\}$. In the following, for $S \subset K[X_1, \ldots, X_n]$, we denote by S_k the subset of S such that $S_k = \{p \in S \mid \mathrm{mvar}(p) \leq X_k\}$.

Definition 4.1 *$\mathcal{T}$ is said to be in quasi-generic position if there exists $k \in \{d+1, \ldots, n\}$ such that*

$$\forall i > k \quad \deg(t_i, X_i) = 1.$$

We denote by k the index of quasi-generic position of $\mathcal{T}_k$.

In the following, we show how to take advantage of quasi-generic position to decide emptiness of $V(\mathcal{G}) \bigcap R^n$ by performing computations in $K[X_1, \ldots, X_k]$ which decreases the size of the computed determinants.

If $\mathcal{T}$ is in quasi-generic position, we may suppose, without lost of generality, that $t_j = h_j X_j + q_j$ with h_j , $q_j \in K[X_1, \ldots, X_k]$, $\forall j = k+1 \ldots n$.
If $V(\mathcal{G}_k) = \emptyset$ then $V(\mathcal{G}) = \emptyset$.
Suppose that $V(\mathcal{G}_k) \bigcap R^k \neq \emptyset$ and let $M \in \mathcal{D}(V(\mathcal{G}_k, A_k)) \bigcap R^k$ for any $A_k \in R^k$.

- If $M = (x_1, \ldots, x_k) \notin V(h_{k+1})$, there exists a unique value $y \in R$ such that $M' = (x_1, \ldots, x_k, y) \in V(\mathcal{T}_{k+1})$. Moreover, if $M \notin V(\prod_{j=d+1}^{k+1} h_j)$ then, $M' \in V(\mathcal{G}_{k+1}) \bigcap R^{k+1}$.
- Suppose $M \in V(h_{k+1})$ and $M \notin Sing(V(\mathcal{G}_k))$. Since $dim(V(h_{k+1}) \bigcap V(\mathcal{G}_k)) < dim(V(\mathcal{G}_k))$ and since M is a regular point of $V(\mathcal{G}_k)$, there exists a neighborhood $U \subset V(\mathcal{G}_k) \bigcap R^k$ containing a point N such that $h_{k+1}(N) \neq 0$ and so, according to the preceeding item, $N \in V(\mathcal{G}_{k+1}) \bigcap R^{k+1}$.

Since the cases where $\mathcal{D}(V(\mathcal{G}_k), A) \subset \mathrm{Sing}(V(\mathcal{G}_k)) \cap V(\prod_{i=d+1}^{n} h_i) \bigcap R^k$ are rare, we propose a specific algorithm, based on **Algorithm 3** and optimized to decide the emptiness.
In the following, we denote by $\Delta(\mathcal{G}_k)$ of all the minors of order $k-d$ of the Jacobian matrix associated to $\mathcal{G}_k$ and we define new external functions :

- **ZeroDimTest :** takes as input a zero-dimensional system S and returns *true* if $V(S) \bigcap R^n = \emptyset$, else it returns *false*.

- **cleaningStep :** takes as input a zero-dimensional system S and a polynomial p, and returns a list of real solutions of S which do not vanish p. This can be done by gcd computations on univariate polynomials if we solve zero-dimensional systems by computing Rational Univariate Representations (see [26, 27]).

Algorithm 4

- **Input :** A polynomial system S of equations in $K[X_1, \ldots, X_n]$.
- **Output :** *true* if $V(S) \bigcap R^n = \emptyset$, else it returns *false*.

1. list := LexTriSetEquiDim(S), result := true,
2. Choose $A \notin V(S)$.
3. while list $\neq \emptyset$ do
   - (*)S := first(list), and remove S from list, set d := Dim(S),
   - if $d = 0$ and if ZeroDimTest(S) = false then return *false*,
   - $\mathcal{T} := ExtractTriangular(S)$.
   - if $\mathcal{T}$ is in quasi-generic position then
     - newlist := Algorithm3($\Gamma_A(\mathcal{T}_k) \cup S_k$), where k is the index of quasi-generic position of $\mathcal{T}_k$,
     - for S' in newlist, remove S' if ZeroDimTest(S') = true,
     - if newlist = $\emptyset$ then go to step (*),
     - for S' in newlist, remove S' from newlist and if cleaningStep($S', \Delta(\mathcal{G}_k)$) $\neq \emptyset$ then return *false*,
     - for S' in newlist, remove S' from newlist and if cleaningStep($S', \prod_{i=d+1}^{n} h_i$) $\neq \emptyset$ then return *false*,
   - (**) $Q = \Gamma_A(\mathcal{T}) \bigcup S$ and set $u = \mathrm{Dim}(Q)$
   - if $u = d$ choose another point $A \notin S$ and go to step (**).
   - $d := u$; list := list $\bigcup$ LexTriSetEquiDim(Q),
4. return result.

Remark 4.2 *One can guarantee that the method described above computes at least one point in each connected component if* $\forall i \in \{k+1, \ldots, n\}, \quad h_i \in K$*, which occurs, for example, when the system is in Noether position. In other cases, this can not be guaranteed (consider the example :* $t_2 = y - x$ *and* $t_3 = xz - 1$ *with* $z > y > x$ *and take* $A = (0, 1)$*).*

5 Experiments

This section is devoted to present some tests performed with an experimental implementation of our algorithms.

5.1 Software and basic algorithms

The equi-dimensional decompositions have been implemented using lexicographical Gröbner basis and techniques of split from [23, 3]. It takes as input a polynomial system S and returns a list of lexicographical Gröbner basis $\mathcal{G}_1, \ldots, \mathcal{G}_\ell$ such that :

- $V(S) = V(\mathcal{G}_1) \bigcup \ldots \bigcup V(\mathcal{G}_\ell)$,
- each lexicographical Gröbner base generates a radical equi-dimensional ideal,
- for all $i \in \{1, \ldots, \ell\}$ a regular and separable triangular set $\mathcal{T}_i$ can be extracted from $\mathcal{G}_i$ such that $\mathrm{sat}(\mathcal{T}_i) = \langle \mathcal{G}_i \rangle$.

A full description of the algorithm can be found in [31]. It has been implemented using Gb (software devoted to computations of Gröbner basis, implemented in C++ by J.-C. Faugère) and Maple.

The resolution of zero-dimensional systems (counting/isolating of the real roots) has been done using **ZDS** algorithm (Rational Univariate Representation + Isolation of the Real Roots) which uses Gb and RS (software implemented by F. Rouillier). In particular, all the computations have been done using exclusively exact computations.

The other parts of the algorithms were implemented in Maple.

In order to show the efficiency of our algorithms, we have applied the CAD algorithm on each example. Remember that this method is more general than ours. In particular, it is currently the only efficient method able to compute, in practice, at least one point on every semi-algebraically connected component of a semi-algebraic variety. The implementation we used (QEPCAD) is built upon the SACLIB library and has been provided by Hoon Hong.

5.2 The methodology

The polynomial systems used for our experiments come from various sources and most of them can be found in the FRISCO Test-Suite (see [16]). A larger list is available on the web page [1].
We may point out that the examples *F633*, *F744* and *F855* come from an industrial application (design of filter banks - see [15]).
All the computations have been performed on a PC Pentium II 400 MHz with 512 Mo of RAM (machine of the UMS MEDICIS [2]). The timings are given in seconds.
We chose to stop the computations systematically after 12 hours. Also, the symbol ∞ in the timing tables means in fact *stopped after 12 hours.*
It happens that the CAD fails when the number of cells becomes too large. In such cases, we put *failed(n)*, where n denotes a lower bound of the number of cells, in the tables.

5.3 Algorithm 2 / Algorithm 3

The goal of these tests is to show how the use of triangular sets decreases the computation times.

The following table contains the timings for the computation of all the zero-dimensional systems (outputs of Algorithm 2 and Algorithm 3) but excludes the computation times related to their resolution. In the columns *Algorithm 2* and *Algorithm 3* the first number is the cumulative computation time of the equi-dimensional decompositions, while the second one is the cumulative computation time of the determinants. If one of these both columns contains "?", it means that the preceeding step (either an equi-dimensional decomposition computation, or a determinant computation) has not ended.

System	Dimension/Degree	Nb Vars	Algorithm 2		Algorithm 3	
Vermeer	1,26	5	0.01	0	0.01	0
Wang	1,114	13	0.12	0	0.12	0
Euler	3,2	10	0.01	0	0.01	0
Neural	1,24	4	0.43	0	0.43	0
Butcher	3,3	8	1.7	0	1.7	0
Buchberger	4,6	8	0	0	0	0
DiscPb	2,3	4	0.02	0	0.02	0
Donati	1,10	4	0.04	26	0.04	0
Hairer2	2,25	13	?	∞	∞	?
Prodecco	2,2	5	284	26	284	0
F633	2,32	10	?	∞	∞	?
F744	1,40	12	24.06	∞	24.06	0.02
F855	1,52	14	5654	∞	5654	173

Table 1 : computation times for Algorithm 2 and Algorithm 3

One can remark that the construction of the zero-dimensional systems is a limiting step in Algorithm 2 and not in Algorithm 3. In Algorithm 3, we compute only a subset of the set of the determinants needed by algorithm 2.

5.4 Algorithm 3 / CAD

5.4.1 Size of the output

In the following table, we give the number of points computed by **Algorithm 3** (sum of the degrees of the zero-dimensional systems) and by the CAD implementation **QEPCAD** on the examples of table 1 for which at least one of these methods ends . When **QEPCAD** is stopped after 12 hours, we put ∞ in the table. If the computation failed because the number of cells is too large, we put failed(n), where n is the lower bound of number of cells that **QEPCAD** has predicted.

System	**Algorithm 3 + ZDS**	**QEPCAD**
Vermeer	84	65976
Wang	132	∞
Euler	10	failed(872043)
Neural	133	205
Butcher	15	∞
Buchberger	32	failed(991324)
DiscPb	28	∞
Donati	61	10

Table 2 : comparison between (Algorithm 3 + ZDS) and QEPCAD

One can observe that these results are coherent with what we expected : the size of the output of CAD is a lot bigger than the size of the output of our algorithms.

We can also remark that none of the methods solved the examples *Hairer2*, *Prodecco*, *F633*, *F744* and *F855*, even if *Algorithm 3* provided all the zero-dimensional systems. These systems were too large for the computation of a Gröbner basis by Gb.

5.4.2 Computation times

One of our motivations was to provide an algorithm whose output is reasonable with the hope to get significantly better computation times, compared to existing implementations that computes at least the same thing, even if the methods used have not, theoretically, a better complexity in terms of computation times.
The next table shows that both algorithms **Algorithm 2 + ZDS** and **Algorithm 3 + ZDS** have a better behavior, in practice, than **QEPCAD** :

System	Algorithm 2 + ZDS	Algorithm 3 + ZDS	QEPCAD
Vermeer	62.36	3.32	43
Wang	1.37	1.37	∞
Euler	0.01	0.01	failed(*872043*)
Neural	1.02	1.02	0.9
Butcher	1.7	1.7	∞
Buchberger	< 0.01	< 0.01	failed(*991324*)
DiscPb	0.2	0.2	∞
Donati	11609	10	0.6

Table 3 : Computation times for Algorithm 2 +ZDS, Algorithm 3 +ZDS and QEPCAD.

5.5 Algorithm 4

The last table shows the progress induced by **Algorithm 4**. According to remark 4.2, **Algorithm 4** computes one point on each connected component in favorable cases, and allows to decide if a variety is empty or not in any case. The examples for which **Algorithm 4** gives at least one point on each semi-algebraically connected component are marked by *.

System	Algorithm 2 + ZDS	Algorithm 3 + ZDS	Algorithm 4	QEPCAD
Vermeer	62.36	3.32	<0.01	43
Wang	1.37	1.37	0.13	∞
Euler	0.01	0.01	<0.01*	failed(*872043*)
Neural	1.02	1.02	0.44*	0.9
Butcher	1.7	1.7	1.7*	∞
Buchberger	< 0.01	<0.01	<0.01*	failed(*991324*)
DiscPb	0.2	0.2	0.02	∞
Donati	11609	10	0.04	0.6
Hairer2	∞	∞	23.03	failed(*872043*)
Prodecco	∞	∞	286	∞
F633	∞	∞	5700	∞
F744	∞	∞	40	∞
F855	∞	∞	5664	∞

Table 4 : Computation times of Algorithm 2 + ZDS, Algorithm 3 + ZDS and QEPCAD.

In terms of computations, the difference between **Algorithm 3** and **Algorithm 4** is the number and size of the intermediate determinants. One can see that the zero-dimensional systems provided by **Algorithm 4** are much more simple to solve.

The cases where **Algorithm 4** computes one point on each semi-algebraic component are few. This means, in particular, that, in our test list, the systems in Noether position are few, and so justifies a large part of our study whose goal is to provide an algorithm that works in every situation.

6 Conclusions

We have provided an efficient algorithm (Algorithm 3) in practice that allows to compute one point on each semi-algebraically connected component of a real algebraic variety, without assumption neither on the variety (smoothness, compactness) nor on the system of polynomial equations that define it.

We proposed an optimization (Algorithm 4) for deciding the emptiness of the variety in any cases or for computing at least one point on each connected component in *generic* cases (see Remark 4.2). According to our experiments, we noticed that in practice, these conditions of genericity (for example the Noether position) are too strong, which prevents **Algorithm 4** for computing at least one point on each semi-algebraically connected component.

Moreover, we should obtain much better timings in a near future. For example, we try a recent prototype, due to J.C. Faugère, of an algorithm for computing prime decompositions that speeds up our algorithms : with this implementation, *Algorithm 4* can decide emptiness for *F633* in 7.2 sec. and *F855* in 26 sec.

According to other experiments we made, additional assumptions on the variety (smoothness, compactness) or on the system of equations that defines it (Noether position, radical, prime, etc ...) speeds up strongly the method. For example, if we suppose the real algebraic set to be compact, then, we can replace the distance function by any projection with respect to one coordinate X_i. In practice, it is sufficient to replace $\overrightarrow{AM}$ by the vector $\overrightarrow{u_i}$ whose coordinates are null except the i-th.

The theoretical complexity of our method strongly depends on the complexity of the used decompositions (equi-dimensional or prime). So note that there is no precise result about it. We can just give an upper doubly exponential in the number of variables bound (since we compute lexicographical Gröbner bases).

We plan to extend our work to the case of semi-algebraic sets, generalizing our main results or simply applying well known transformations (see for example [30]) that comes to study real algebraic varieties.

References

[1] P. Aubry, F. Rouillier, M. Safey El Din, *Practical comparisons,* available on the Web, `http://calfor.lip6.fr/~safey/benchs.html`

[2] UMS MEDICIS, `http://medicis.polytechnique.fr/`.

[3] P. Aubry, *Ensembles triangulaires de polynômes et résolution de systèmes algébriques. Implantation en Axiom*, Doctoral Thesis, University of Paris VI, 1999.

[4] P. Aubry, D. Lazard, M. Moreno Maza, *On the theories of triangular sets,* in Journal of Symbolic Computation, 1999.

[5] B. Bank, M. Giusti, J. Heintz, and M. MBakop *Polar Varieties and Efficient Real Elimination*, in Mathematische Zeitschrift, 2000.

[6] S. Basu, R. Pollack, M.-F. Roy, *A New Algorithm to Find a Point in Every Cell Defined by a Family of Polynomials,* in Quantifier Elimination and Cylindrical Algebraic Decomposition, Texts and Monographs in Symbolic Computation, B. Caviness and J. Johnson, Eds. 341-349, Springer-Verlag, Wien, New York (1998).

[7] S. Basu, R. Pollack, M.-F. Roy, *On the combinatorial and algebraic complexity of Quantifier elimination.* J. Assoc. Comput. Machin., 43, 1002–1045, (1996).

[8] E. Becker, R. Neuhaus, *Computation of Real Radicals for polynomial ideals*, in Computational Algebraic Geometry, Progress in Math., vol. 109, 1-20, Birkhaüser, 1993.

[9] J. Bochnak, M. Coste, M.-F. Roy, *Real algebraic geometry,* Springer-Verlag (1999).

[10] J. Canny , *A toolkit for nonlinear algebra,* Goldberg, Ken (ed.) et al., Algorithmic foundations of robotics, Proceedings of the workshop on the algorithmic foundations of robotics, WAFR '94, held in San Francisco, CA, USA, 17-19 February, 1994. Wellesley, MA: A.K. Peters.

[11] G. E. Collins, *Quantifier elimination for real closed fields by cylindrical algebraic decomposition,* Springer, Lecture Notes in Computer Science 33, 515- 532, (1975).

[12] P. Conti, C. Traverso, *Algorithms for the real radical,* unpublished manuscript

[13] D. Cox, J. Little, D. O'Shea, *Ideals, Varieties, and Algorithms,* Springer-Verlag (1991)

[14] J.C. Faugère, *FGb*, available on the web, http://www-calfor.lip6.fr/~jcf.

[15] J.C. Faugère and F. Moreau de Saint Martin and F .Rouillier *Design of regular nonseparable bidimensional wavelets using Groebner bases techniques*, in IEEE SP Transactions Special Issue on Theory and Applications of Filter Banks and Wavelets (1997).

[16] The FRISCO Test-suite, available on the web http://www-sop.inria.fr/saga/POL

[17] D. Grigor'ev, N. Vorobjov, *Solving Systems of Polynomial Inequalities in Subexponential Time,* J. Symbolic Comput., 5:37–64, (1988).

[18] H. Hong, Comparison of several decision algorithms for the existential theory of the reals, Research report, RISC, (1991).

[19] J. Heintz, M.-F. Roy, P. Solernó , *On the Complexity of Semi-Algebraic Sets,* Proc. IFIP 89, San Francisco. North-Holland 293-298 (1989).

[20] J. Heintz, M.-F. Roy, P. Solernó , *On the theoritical and practical complexity of the existential theory of the reals,* Comput. J. 36, No. 5, 427-431 (1993).

[21] M. Kalkbrener, *Three contributions to elimination theory,* Doctoral thesis, 1991.

[22] D. Lazard, *A new method for solving algebraic systems of positiove dimension,* in Discrete Applied Mathematics, 1991.

[23] M. Moreno Maza, *Calculs de Pgcd au-dessus des Tours d'Extensions Simples et Résolution des Systèmes d'Equations Algébriques,* Doctoral Thesis, University of Paris VI, 1997.

[24] D. Mumford *Algebraic Geometry I, Complex projective varieties*, Berlin, Heildelberg, New York : Springer Verlag (1976).

[25] J. Renegar *On the computational complexity and geometry of the first order theory of the reals,* J. of Symbolic Comput.13(3):255-352, (1992).

[26] F. Rouillier, *Algorithmes efficaces pour l'étude des zéros réels des systèmes polynomiaux,* Doctoral Thesis, University of Rennes I (1996).

[27] F. Rouillier, *Solving Zero-Dimensional Systems through the Rational Univariate Representation,* AAECC Journal.9 : 433-461 (1999).

[28] F. ROUILLIER, M.-F. ROY, M. SAFEY EL DIN, *Finding at least one point in each connected component of a real algebraic set defined by a single equation,* to appear in Journal of Complexity, 1999.

[29] F. ROUILLIER, M. SAFEY EL DIN, *Some Benchmarks for RSDF Algorithm,* available on the Web http://posso.lip6.fr/ safey/benchs.html

[30] M.-F. ROY, *Basic algorithms in real algebraic geometry: from Sturm theorem to the existential theory of reals,* Lectures on Real Geometry in memoriam of Mario Raimondo, Expositions in Mathematics 23, 1- 67. Berlin, New York: de Gruyter (1996).

[31] M. SAFEY EL DIN, *Résolution réelle des systèmes polynomiaux en dimension positive,* Doctoral Thesis, in preparation, University Paris 6.

[32] A. SEIDENBERG, *A new decision method for elementary algebra,* Annals of Mathematics, 60:365–374, (1954).

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 - 38330 Montbonnot-St-Martin (France)
Unité de recherche INRIA Rocquencourt : Domaine de Voluceau - Rocquencourt - BP 105 - 78153 Le Chesnay Cedex (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