On the combinatorial and algebraic complexity of Quantifier Elimination

S. Basu *

Department of Computer Science, Courant Institute New York, NY 10012

R. Pollack †

Department of Mathematics, Courant Institute New York, NY 10012

M.-F. Roy ‡

Department of Mathematics, Université de Rennes Campus de Beaulieu 35042 Rennes cedex FRANCE

Abstract

In this paper we give a new algorithm for performing quantifier elimination from first order formulae over real closed fields. This algorithm improves the complexity of the asymptotically fastest algorithm for this problem, known to this date. A new feature of our algorithm is that the role of the algebraic part (the dependence on the degrees of the input polynomials) and the combinatorial part (the dependence on the number of polynomials) are separated, making possible our improved complexity bound. Another new feature is that the degrees of the polynomials in the equivalent quantifier-free formula that we output, are independent of the number of input polynomials. As special cases of this algorithm, we obtain new and improved algorithms for deciding a sentence in the first order theory over real closed fields, and also for solving the existential problem in the first order theory over real closed fields.

Using the theory developed in this paper, we also give an improved bound on the radius of a ball centered at the origin, which is guaranteed to intersect every connected component of the sign partition induced by a family of polynomials. We also use our methods to obtain algorithms for solving certain decision problems in real and complex geometry which improves the complexity of the currently known algorithms for these problems.

*Supported in part by NSF grant CCR-9122103

†Supported in part by NSF grant CCR-9122103, and NSA grant MDA904-92-H-3075.

‡Supported in part by the project ESPRIT-BRA 6846POSSO

1 Introduction

1.1 The Problem

We are given a family $\mathcal{P} = \{P_1, \ldots, P_s\}$ of s polynomials in $k + l$ variables, $X_1, \ldots, X_k, Y_1, \ldots, Y_l$. The degrees of the polynomials are bounded by d, and the coefficients are drawn from an ordered domain D, contained in a real closed field R. We are given a first-order formula of the form,

$$(Q_\omega X^{[\omega]}) \ldots (Q_1 X^{[1]}) F(P_1, \ldots, P_s),$$

(henceforth, denoted $\Phi(Y)$) where, $Q_i \in \{\forall, \exists\}$, $Q_i \neq Q_{i+1}$, $Y = (Y_1, \ldots, Y_l)$ is a block of l free variables, $X^{[i]}$ is a block of k_i variables, $\sum_{1 \le i \le \omega} k_i = k$, and $F(P_1, \ldots, P_s)$ is a quantifier-free Boolean formula with atomic predicates of the form,

$$P_i(Y, X^{[\omega]}, \ldots, X^{[1]}) \, \epsilon \, 0, 1 \le i \le s,$$

where $\epsilon \in \{>, <, =\}$.

The quantifier elimination problem is to construct a quantifier-free Boolean formula, $\Psi(Y)$, such that for any $y \in R^l$, $\Phi(y)$ is true if and only if $\Psi(y)$ is true.

There are two special cases of this problem which are interesting in their own right. First, if $l = 0$, so that there are no free variables, then the above problem is to decide whether a given first order sentence over the reals, is true or not. This is known as the *General Decision Problem* for the first order theory of the reals. Secondly, if $l = 0$ and $\omega = 1$, so that there are no free variables and only one block of quantified variables, (without loss of generality, we can assume that the quantifier is the existential one), then the problem is to decide whether a given set of equalities and inequalities involving the input polynomials, has a solution.

This problem is known as the *Decision Problem for the Existential Theory* of the reals.

The above problems have wide applications. For instance, an algorithm for the general decision problem, allows us to decide any sentence in elementary geometry. Another application is in the area of robotics, where the configuration space of a robot can often be described algebraically, and a decision procedure for the first order theory of the reals, allows one to answer many questions about the configuration space. There are various other applications in fields like mathematical programming, computational geometry, computer vision, solid modeling etc.

We now give a brief sketch of the history of these problems.

1.2 History

The first result is due to Tarski who proved that there is an algorithm deciding any sentence of the first order theory of the reals. The complexity of the decision procedure for the first order theory of the reals, given by Tarski [29] (see also ([28]) was not elementary recursive. The first algorithm, with a reasonable worst-case time bound was given by Collins ([10]). His algorithm had a worst case running time doubly exponential in the number of variables. Grigor'ev and Vorobjov ([15]) gave an algorithm to solve the decision problem for the existential theory of the reals. Their algorithm has a time complexity which is singly exponential in the number of variables. Grigor'ev ([13]), extended this result to the general decision problem and achieved doubly exponential complexity in the number of blocks. It should be noted that for a fixed value of ω, this is only singly exponential in the number of variables. However, all these algorithms have the drawback that they require that the coefficients of the input polynomials be integers rather than belonging to an arbitrary ordered ring contained in a real closed field.

Heintz, Roy and Solerno ([16]) gave an algorithm to decide the existential problem in a singly exponential number of arithmetic operations over any ring and the quantifier elimination problem in a number of operations, doubly exponential in the number of blocks. Renegar's ([25]) algorithms which solve the general decision problem in time $(sd)^{\Pi O(k_i)}$, and the quantifier elimination problem in time $(sd)^{(l+1)\Pi O(k_i)}$ were the best to date. The degrees of the polynomials in the formula output by his quantifier elimination algorithm, are bounded by $(sd)^{\Pi O(k_i)}$.

We prove the following theorem, in which we improve the complexity of Renegar's algorithm by achieving a separation of the combinatorial part, (the dependence on s), and the algebraic part (the dependence on d), and improving the combinatorial complexity. Moreover, the degrees of the polynomials in the quantifier-free formula that we output, do not depend on s. The algebraic complexity of our algorithm is the same as that of Renegar's algorithm and we use many of his techniques as well.

Theorem 1 *(The Quantifier Elimination Problem) Given a family*

$$\mathcal{P} = \{P_1, \ldots, P_s\} \subset D[X_1, \ldots, X_k, Y_1, \ldots, Y_l]$$

of polynomials whose degrees are bounded by d, as well as a first-order formula,

$$\Phi(Y) = (Q_\omega X^{[\omega]}) \ldots (Q_1 X^{[1]}) F(P_1, \ldots, P_s),$$

there exists an equivalent quantifier-free formula,

$$\Psi(Y) = \vee_{i=1}^{I} \wedge_{j=1}^{J_i} (P_{ij}(Y)\ \epsilon_{ij}\ 0),$$

where $P_{ij}(Y)$ are polynomials in the variables Y, $\epsilon_{ij} \in \{>, <, =\}$

$$I \leq s^{(l+1)\Pi_i(k_i+1)} d^{(l+1)\Pi_i O(k_i)},$$

$$J_i \leq s^{\Pi_i(k_i+1)} d^{\Pi_i O(k_i)},$$

and the degrees of the polynomials $P_{ij}(Y)$ are bounded by $d^{\Pi_i O(k_i)}$. Moreover, there is a well-behaved *(see below) algorithm that computes $\Psi(Y)$ using $s^{(l+1)\Pi(k_i+1)} d^{(l+1)\Pi O(k_i)}$ arithmetic operations in D.*

As special cases of the above theorem, we immediately derive the following two corollaries.

Corollary 1 *(The General Decision Problem) Given a first order formula of the form,*

$$(Q_\omega X^{[\omega]})(Q_{\omega-1} X^{[\omega-1]}) \ldots (Q_1 X^{[1]}) F(P_1, \ldots, P_s),$$

where, $Q_i \in \{\forall, \exists\}, Q_i \neq Q_{i+1}$, $X^{[i]}$ is a block of k_i variables, $\sum_{1 \leq i \leq \omega} k_i = k$, and $F(P_1, \ldots P_s)$ is a quantifier-free Boolean formula, with atoms of the form $P_i(X^{[\omega]}, \ldots, X^{[1]})\ \epsilon\ 0$, where $\epsilon \in \{>, <, =\}$, there exists a well-behaved algorithm to decide the truth of the formula which uses $s^{\Pi(k_i+1)} d^{\Pi O(k_i)}$ arithmetic operations in D.

If $\omega = 1$, we have the following corollary.

Corollary 2 *(The Decision Problem for the Existential Theory) Given a first order formula of the form,*

$$\exists (X_1, \ldots, X_k) F(P_1, \ldots, P_s)$$

where $F(P_1,\ldots,P_s)$ is a quantifier-free Boolean formula, with atoms of the form $P_i(X_1,\ldots,X_k)\ \epsilon_j\ 0$, where $\epsilon_j \in \{>,<,=\}$, there exists a well-behaved algorithm to decide the truth of the formula which uses $s^{k+1}d^{O(k)}$ arithmetic operations in D.

We say that an algorithm is *well-behaved* if it satisfies the following conditions.

- If $D = \mathbf{Z}$, the maximum bit-size of the output and of any intermediate computations is obtained by multiplying the number of arithmetic operations by the bit-size of the inputs.
- If the cost of evaluating the Boolean formula given the truth values of its atoms is L, and the algorithm performs $O(N)$ arithmetic operations (additions and multiplications in the ring D), then the total cost of performing the Boolean evaluations is $O(NL)$. Thus the cost of evaluating the Boolean formula enters as a multiplicative factor to the total number of arithmetic operations performed by the algorithm. Note that all the known algorithms for the decision problem, and the quantifier elimination problem, which do not assume any particular structure on the input formulae, have this property (see [25]).
- It is well parallelizable i.e. it can be described by an arithmetic network whose depth is a polynomial in the log of the sequential complexity (see [16]).

Since our algorithm is well-behaved, we define the complexity of our algorithm to be the number of arithmetic operations in the ring D, keeping in mind that the cost of and the size of the integer coefficients (if we are counting the number of bit operations) enter as multiplicative factors.

2 Outline of our Algorithm

In the remaining part of this paper, we give an overall description of our algorithm. We will skip many details, and refer the reader to the full paper [4], where we present the algorithm in greater detail.

In our algorithms, we construct points whose coordinates belong to certain real closed non-archimedean extensions of R. Typically, these are fields of Puiseux series, in one or more infinitesimals, with coefficients in R. We denote by $R\langle\epsilon\rangle$, the field of Puiseux series in ϵ with coefficients in R. (Here and elsewhere, when we refer to a variable as an infinitesimal with respect to the real closed field R we will mean that it is positive and smaller than any positive element of R.)

The coordinates of the points that we compute might belong to $R\langle\epsilon\rangle$. However, these coordinates are always algebraic over $D[\epsilon]$, and hence are roots of some polynomial in $D[\epsilon][t]$. We say that such points are *defined* over $D[\epsilon]$.

We denote by $eval_\epsilon$ the map from the valuation ring of $R\langle\epsilon\rangle$ to R, obtained by evaluating the Puiseux series at $\epsilon = 0$. We refer the reader to [3] for a complete list of properties of Puiseux series that we utilize.

Whenever we compute a point $x = (x_1,\ldots,x_k)$ what we actually compute is :

1. A univariate polynomial $f(t)$.
2. A root, say α of f which is characterized by f and the sign vector

   $$(\mathrm{sign}(f(\alpha)), \mathrm{sign}(f'(\alpha)), \ldots, \mathrm{sign}(f^{\{degf\}-1}(\alpha))).$$

   This sign vector is called the *Thom encoding* of α.
3. $k+1$ polynomials $g_0(t),\ldots,g_k(t)$, such that $x_i = \dfrac{g_i(\alpha)}{g_0(\alpha)}, 1 \le i \le k$.

We call the tuple $(f(t), g_0(t), \ldots, g_k(t))$ a *univariate representation* of the point x, and say that the point is *associated* to this univariate representation.

Moreover, if the polynomials have coefficients which are themselves polynomials in some variables, $Y_1,\ldots,Y_m$, which we call *parameters*, then we output *parametrized univariate representations* of the form $(f(Y,t), g_0(Y,t), \ldots, g_k(Y,t))$. For any $y \in R^m$, the coordinates of the points associated to a parametrized univariate representation are obtained by evaluating $\dfrac{g_i(y,\alpha)}{g_0(y,\alpha)}, 1 \le i \le k$, at the real roots, α, of the polynomial $f(y,t)$, where the parameters have been *specialized* to y.

The *sign condition* of a set $\mathcal{P} = \{P_1,\ldots,P_s\} \subset D[X_1,\ldots,X_k]$ at a point x is the sign vector $\sigma \in \{-1,0,+1\}^s$ such that

$$(\mathrm{sign}(P_1(x)),\ldots,\mathrm{sign}(P_s(x))) = \sigma.$$

The *realization space* of a sign condition σ in R is the set

$$\{x \in R^k | (\mathrm{sign}(P_1(x)),\ldots,\mathrm{sign}(P_s(x))) = \sigma\}.$$

We will abuse language and use σ to denote a sign condition as well as its realization space. Thus, a sign condition σ is called *non-empty* (with respect to $\mathcal{P}$) if its realization space is non-empty.

Given a polynomial Q with coefficients in a real closed field R, we will denote by $Z(Q)$ the set of real zeros of Q.

Our algorithm builds on the following basic subroutine, which we call the *Parametrized Sample Points Subroutine* (see [4] or[2]). Given a set of s polynomials

$$\mathcal{P} = \{P_1, \ldots, P_s\} \subset D[Y_1, \ldots, Y_m, X_1, \ldots, X_k],$$

the *Parametrized Sample Points Subroutine* computes a set of parametrized univariate representations with parameters $Y = (Y_1, \ldots, Y_m)$, of size $s^k d^{O(k)}$, with the following property: for every $y \in R^m$ the union of the sets of points associated to the parametrized univariate representations computed by the subroutine intersect every connected component of every realizable sign-condition (henceforth, referred to as the *cells*) of the set $\mathcal{P}(y)$. Moreover, the *Parametrized Sample Points Subroutine* uses $s^k d^{O(mk+k)}$ arithmetic operations in D and the degrees of the polynomials in the output are bounded by $d^{O(k)}$.

2.1 The Parametrized Sample Points Subroutine

We first consider a set of polynomials $\mathcal{P}$ without any parameters. The following basic proposition, which we prove in [3], makes it possible to reduce the problem of computing a point in every cell of $\mathcal{P}$, to finding points in every connected component of certain algebraic sets.

Proposition 1 *Let $\mathcal{P} \subset D[X_1, \ldots, X_k]$ be a finite set of polynomials, and C be a connected component of a non-empty sign condition of the form $P_1 = \ldots = P_\ell = 0, P_{\ell+1} > 0, \ldots, P_s > 0$, then we can find an algebraic set $\mathcal{V}$ in $(R\langle\epsilon\rangle)^k$ defined by equations $P_1 = \ldots = P_\ell = P_{i_1} - \epsilon = \ldots P_{i_m} - \epsilon = 0$, such that a connected component of $\mathcal{V}$ is contained in $C_{R\langle\epsilon\rangle}$ and $eval_\epsilon(\mathcal{V}) \neq \emptyset$. (Here, ϵ is an infinitesimally small positive variable, and $R\langle\epsilon\rangle$ is the field of Puiseux series in ϵ with coefficients in R.)*

Notice, that the above proposition implies that in order to find points in every cell of $\mathcal{P}$, it suffices to compute points on the connected components of certain algebraic sets, that is sets defined by polynomial equalities, over a real closed extension, $R\langle\epsilon\rangle$ of R.

Given a family of polynomials $\mathcal{P}$, we denote by $\bar{\mathcal{P}}$ the family of polynomials $\cup_{P\in\mathcal{P}}\{P, P+\epsilon, P-\epsilon\}$.

One easy but nevertheless crucial, observation is that in the case of real closed fields the common zeros of a set of polynomials are also the zeros of the single polynomial which is the sum of the squares of the polynomials. Hence, any algebraic set that occurs, which is defined as the common zeros of a collection of polynomials whose degrees are bounded by d, is defined by a single polynomial of degree at most $2d$. This is the key that makes the degrees of the polynomials in the output formulae independent of s.

The number of such algebraic sets, we need to consider can be exponentially large in s. However, if the set of polynomials $\bar{\mathcal{P}}$ satisfies the requirement, (which we call the *general position requirement*), that no $k+1$ polynomials in $\bar{\mathcal{P}}$ have a common real zero, then it is sufficient to consider at most $(O(s))^k$ algebraic sets. Each such algebraic set is defined by $l \le k$ polynomial equations whose degrees are bounded by d and hence by a single polynomial equation whose degree is bounded by $2d$. It is possible to compute points in every connected component of such an algebraic set using $d^{O(k)}$ arithmetic operations (this is implicit in [9], [25]). We follow [3] and compute these points using the *Cell Representatives Subroutine*.

We briefly outline the *Cell Representatives Subroutine*.

Given a polynomial $Q \in R[X_1, \ldots, X_k]$ we define the *total degree of Q in X_i* as the maximal total degree of the monomials in Q containing the variable X_i.

The input to the *Cell Representatives Subroutine* is a polynomial $Q \in R[X_1, \ldots, X_k]$, whose total degrees in $X_1, \ldots, X_k$, are $d_1, \ldots, d_k$, (without loss of generality we can assume that $d_1 \ge \cdots \ge d_k$,) and its output is a point in every connected component of $Z(Q)$.

In the first step we replace k by $k+1$, the polynomial Q by

$$Q^2 + ({X_1}^2 + \cdots + {X_{k+1}}^2 - \Omega^2)^2$$

where Ω is an infinitely large element and X_{k+1} is an extra variable, and the field R by $R\langle\frac{1}{\Omega}\rangle$. We then use a new infinitesimal ζ to perturb the polynomial Q and define

$$Q_1 = (1-\zeta)Q + \zeta(X_1^{2(d_1+1)} + \cdots + X_k^{2(d_k+1)} - k(\Omega+1)^{2(d_1+1)}).$$

This makes the perturbed variety, $Z(Q_1)$, smooth, and the number of critical points of the projection map, from $Z(Q_1)$ onto the first co-ordinate, finite. We then compute a set of points intersecting every connected component of this perturbed variety, $Z(Q_1)$, by computing these critical points. These points are the real solutions to the equations,

$$Q_1 = \frac{\partial Q_1}{\partial X_2} = \ldots = \frac{\partial Q_1}{\partial X_k} = 0.$$

The following Proposition, whose proof appears in [3], proves the claimed properties of $Z(Q_1)$, and also provides an easy way to do computations in the quotient ring of the ideal generated by

$$Q_1, \frac{\partial Q_1}{\partial X_2}, \ldots, \frac{\partial Q_1}{\partial X_k}.$$

Proposition 2 *Let Q, and Q_1 be as above. Then the following is true:*

1. *The algebraic set $Z(Q_1) \subset (R\langle\zeta\rangle)^{k+1}$ is bounded and smooth.*

2. *The set $K \subset Z(Q_1)$ of critical points of $Z(Q_1)$ (over $\bar{R}\langle\zeta\rangle$) of the projection map onto the X_1 coordinate, is finite (where $\bar{R} = R[i]$, is the algebraic closure of R).*

   The polynomials,

   $$Q_1, \frac{\partial Q_1}{\partial X_2}, \ldots, \frac{\partial Q_1}{\partial X_k}$$

   whose zero set defines K, form a Gröbner basis for the degree lexicographical ordering with $X_1 > \ldots > X_k$.

3. *Let K' be the set of real critical points of this projection map (with coordinates in $R\langle\zeta\rangle$). For every connected component, C of the algebraic set $Z(Q)$ there exists a point $p \in K'$, such that projection on the first k coordinates of $eval_\zeta(p)$ belongs to C.*

In order to compute the real solutions to the system of equations,

$$Q_1 = \frac{\partial Q_1}{\partial X_2} = \ldots \frac{\partial Q_1}{\partial X_k} = 0,$$

we do a Chow polynomial computation, following [1], and output a set of univariate representations consisting of $(k+2)$-tuples of univariate polynomials, $(f(t), g_0(t), \ldots, g_k(t))$, such that the complex zeroes of the above system are among the points obtained by evaluating the rational functions $(\frac{g_1(t)}{g_0(t)}, \ldots, \frac{g_k(t)}{g_0(t)})$ at the roots of the univariate polynomial $f(t)$. The degrees of the polynomials $f(t), g_i(t)$ are bounded by $(O(d))^k$. We then let ζ go to zero and Proposition 2, ensures that we recover points in every connected component of the variety $Z(Q)$.

Finally we replace Ω by a sufficiently large element of R (see [4]).

We remark that the *Cell Representatives Subroutine* is completely uniform, except the last step where we let $\zeta \to 0$, and that the dimension of the quotient ring, is independent of the particular coefficients of Q. Thus, even if the coefficients of Q, are themselves polynomials in some parameters Y, the algebraic computations would be the same.

In order, to recover points on the variety by letting ζ go to zero in the parametrized case, we do the following:

For each parametrized univariate representation,

$$(f(Y,t), g_0(Y,t), \ldots, g_k(Y,t))$$

computed, we express each polynomial in powers of ζ.

Let,

$$f(Y,t) = \sum_j f(j)(Y,t)\zeta^j,$$

and

$$g_i(Y,t) = \sum_j g_i(j)(Y,t)\zeta^j,\ 0 \leq i \leq k.$$

We output the set of all non-zero tuples of the form

$$(f(j)(Y,t), g_0(j)(Y,t), \ldots, g_k(j)(Y,t))$$

for all the values of j occuring as powers of ζ in the expressions for f and the g_i's. The proof of correctness of this procedure can be found in the full paper [4].

The input set of polynomials $\mathcal{P}$, in general will not satisfy the general position requirement. In order to overcome this difficulty, and still ensure that every connected component of the original set will be intersected, we introduce two additional infinitesimals $\delta \gg \delta'$, and replace each polynomial in $\mathcal{P}$, by four other perturbed polynomials, in such a way that this enlarged family is in general position. More precisely, we follow [25] and define $H_i = 1 + \sum_{1 \leq j \leq k} i^j X_j^{d'}$, for $1 \leq i \leq s$, where d' is the least even integer greater than d, the maximum degree of the polynomials in $\mathcal{P}$. Then it is true that,

Proposition 3 *The polynomials $(1-\delta)P_i + \delta H_i$ are in general position.*

We replace the original set of polynomials by the set,

$$\mathcal{P}^* = \cup_{i=1,\ldots,s}\{(1-\delta')P_i - \delta H_{4i-3}, (1-\delta')P_i + \delta H_{4i-2},$$

$$(1-\delta')P_i - \delta'\delta H_{4i-1}, (1-\delta')P_i + \delta'\delta H_{4i}\}.$$

It can be shown that this enlarged set is in general position as well as the family of polynomials $\bar{\mathcal{P}}^*$.

We compute points in every cell of the enlarged set of polynomials $\mathcal{P}^*$, by considering all possible $l \leq k$

tuples of polynomials in the set $\bar{\mathcal{P}}^*$, $(Q_{i_1}, \ldots, Q_{i_\ell})$, and applying the *Cell Representatives Subroutine* to the polynomial,

$$Q = Q_{i_1,e_1}^2 + \ldots + Q_{i_\ell,e_\ell}^2 + (X_1^2 + \ldots + X_k^2 + X_{k+1}^2 - \Omega^2)^2,$$

where X_{k+1} is an extra variable, and Ω is infinitely large, omitting the first step.

Once we have computed points in every cell of this enlarged set, we let δ' go to 0, and, using the following two propositions, recover points in every cell of $\mathcal{P}$, defined over $D[\delta]$.

Proposition 4 *If $S' \subset (R\langle\delta'\rangle)^k$ is a semi-algebraic set defined over $D[\delta']$ and $S = \mathrm{eval}_{\delta'}(S')$, then S is a semi-algebraic set. Moreover, if S' is connected then S is connected.*

Proposition 5 *Let $\mathcal{P} \subset R[X_1, \ldots, X_k]$ be any set of s polynomials, $P_1, \ldots, P_s$ and let C be a connected component of the sign condition*

$$P_1 = P_2 = \ldots = P_l = 0$$
$$P_{l+1} > 0, \ldots, P_s > 0.$$

Then there exists a connected component C' in $R\langle\delta, \delta'\rangle^k$, satisfying the following inequalities:

$$-\delta'\delta H_{4i} < (1-\delta')P_i < \delta'\delta H_i 4i - 1,\ 1 \le i \le l,$$

$$(1-\delta')P_i > \delta H_{4i-3},\ l+1 \le i \le s,$$

such that $\mathrm{eval}_{\delta'}(C')$ *is contained in the extension of C to $R\langle\delta\rangle$.*

The crucial property of the method that we outlined, is the fact that all its steps are uniform, i.e. there is no branching which depends on the coefficients of the input polynomials. Once again we must be careful about the last step of our subroutine where we let δ', ϵ go to zero, in the parametrized case. However, as in the case of the *Cell Representatives Subroutine* it is possible to do this uniformly without increasing the asymptotic complexity of the subroutine. Thus if the coefficients of our input polynomials are themselves polynomials in some other variables, say $Y = (Y_1, \ldots, Y_m)$, we can output a set of *Parametrized Univariate Representations* depending on Y and t, which if the parameters Y are specialized to say $y \in R^m$, will have its associated points intersecting every cell of the input polynomials, with the parameters specialized to y. The univariate representations computed will then be defined over $D[\delta]$. Moreover, the number of univariate representations produced is bounded by $s^k d^{O(k)}$, and the degrees of the polynomials f, g_i in the univariate representations, are bounded by $(O(d))^k$, since these do not depend on the number of parameters at all. However, since we have to do arithmetic over the ring, $D[Y, V]$, where V denotes the constant number of additional variables introduced by the *Parametrized Sample Points Subroutine* and the *Cell Representatives Subroutine*, the number of arithmetic operations over D, is $s^k d^{O(km+k)}$.

2.2 Quantifier Elimination Algorithm

We now give an outline of our quantifier elimination algorithm, using the subroutines described above.

Given the input set of polynomials, $\mathcal{P} = \{P_1, \ldots, P_s\}$ in $k+l$ variables, $X^{[1]}, \ldots, X^{[\omega]}, Y$, we first call the *Parametrized Sample Points Subroutine* treating the variables $X^{[2]}, \ldots, X^{[\omega]}, Y$ as parameters. The output is a set of parametrized tuples,

$$(f(\bar{X}_1, \delta_1, t_1), g_0(\bar{X}_1, \delta_1, t_1), \ldots, g_{k_1}(\bar{X}_1, \delta_1, t_1)),$$

where $\bar{X}_i$ denotes the variables $X^{[i+1]}, \ldots, X^{[\omega]}, Y$. Thus for a fixed $\bar{x}_1 \in R^{k+l-k_1}$, the points associated to the specialized output tuples,

$$(f(\bar{x}_1, \delta_1, t_1), g_0(\bar{x}_1, \delta_1, t_1), \ldots, g_{k_1}(\bar{x}_1, \delta_1, t_1)),$$

intersects every cell of $\mathcal{P}(\bar{x}_1)(X^{[1]})$, in $(R\langle\delta_1\rangle)^{k_1}$.

The sign conditions of the polynomials $\mathcal{P}(\bar{x}_1)(X^{[1]})$ at these points is denoted by, $SIGN_1(\bar{x}_1)$. Note that $X^{[1]}$ is the innermost block of quantified variables, and we eliminate it, by partitioning R^{k+l-k_1} in such a way, that over each cell of the partition, the set $SIGN_1(\bar{x}_1)$ remains invariant.

We obtain such a partition by computing a set of polynomials $\mathcal{Q}_1 \subset D[\bar{X}_1]$, such that the set $SIGN_1(\bar{x}_1)$ remains invariant over each cell of $\mathcal{Q}_1$.

We make use of the classical theory of subresultant coefficient sequences (see [21], or [22]) and compute the subresultant coefficient sequences, of the following pairs of polynomials.

For every tuple

$$u = (f(\bar{X}_1, \delta_1, t_1), g_0(\bar{X}_1, \delta_1, t_1), \ldots,$$
$$g_{k_1}(\bar{X}_1, \delta_1, t_1)),$$

let

$$f(\bar{X}_1, \delta_1, t_1) = \sum_{0 \le i \le \deg(f)} f_i(\bar{X}_i, \delta_1) t_1^i,$$

and

$$\bar{f}_j(\bar{X}_1, \delta_1, t_1) = \sum_{0 \le i \le j} f_i(\bar{X}_i, \delta_1) t_1^i,$$

$$0 \leq j \leq \deg(f).$$

For every tuple, $u = (f, g_0, \ldots, g_{k_1})$, and every $P \in \mathcal{P}$, we compute the subresultant coefficient sequences of, $\bar{f}_j(\bar{X}_1, \delta_1, t_1), \bar{f}_j'(\bar{X}_1, \delta_1, t_1)$, as well as those of

$$\bar{f}_j(\bar{X}_1, \delta_1, t_1), P_u(\bar{X}_1, \delta_1, t_1),$$

for all j such that $0 \leq j \leq \deg(f)$.

Here $\bar{f}_j{}'$ refers to the derivative with respect to t_1, $\deg(f)$ is the degree of the polynomial f with respect to t_1, and P_u denotes the polynomial obtained by substituting, $(g_0, g_1, \ldots, g_{k_1})$ for $(X_0, X_1, \ldots, X_{k_1})$ in the polynomial, $X_0^{d'} P(\frac{X_1}{X_0}, \ldots, \frac{X_{k_1}}{X_0}, \bar{X}_1)$, where d' is the least even number greater than the degree of P.

We consider all the subresultant coefficients generated this way, as well as the polynomials $f_i(\bar{X}_i, \delta_1)$, and express all these polynomials in powers of δ_1. We then collect together all these coefficients into the set $\mathcal{Q}_1$. Note, that the size of $\mathcal{Q}_1$ is $s^{k_1+1} d^{O(k_1)}$ which follows from the bound on the degrees of the polynomials f and the number of tuples we generate. Using the well known properties of subresultant coefficients, it can be easily shown that as $\bar{x}_1$ varies over each cell of $\mathcal{Q}_1$, the set $SIGN_1(\bar{x}_1)$ remains invariant, as claimed.

Next, we eliminate the block $X^{[2]}$. More precisely, we compute a set of polynomials $\mathcal{Q}_2 \subset D[\bar{X}_2]$, with the property that the k_2 dimensional fibre over each point in one cell of $\mathcal{Q}_2$, meets exactly the same cells of $\mathcal{Q}_1$. This in turn implies that the set,

$$SIGN_2(\bar{x}_2) = \{SIGN_1(x^{[2]}, \bar{x}_2) | x^{[2]} \in R^{k_2}\}$$

remains invariant as $\bar{x}_2$ varies over each cell of $\mathcal{Q}_2$. Moreover, we compute parametrized points in each of these cells. We continue this process of elimination, and at the end of the i^{th} stage, we have a set of polynomials $\mathcal{Q}_i$, such that the set

$$SIGN_i(\bar{x}_i) = \{SIGN_{i-1}(x^{[i]}, \bar{x}_i) | x^{[i]} \in R^{k_i}\}$$

remains invariant, as $\bar{x}_i$ varies over each cell of $\mathcal{Q}_i$, where $SIGN_i(\bar{x}_i)$ is formally defined below. We also have a set of parametrized points, which for a fixed $\bar{x}_i$ meets every cell of $\mathcal{Q}_{i-1}$, met by the fibre over $\bar{x}_i$. Moreover, an easy inductive argument shows that the sizes of the sets $\mathcal{Q}_i$, and the set of parametrized points is bounded by $s^{\Pi_{1 \leq j \leq i}(k_j+1)} d^{\Pi_{1 \leq j \leq i} O(k_j)}$.

We formalize the notion of the nested sets of sign conditions described above, and define what we call the *total list of signs*.

The Total List of Signs: Let $\mathcal{P}$ be a family of s polynomials, $\{P_1, \ldots, P_s\}$, in k variables, $(X_1, \ldots, X_k)$, and let Π denote the partition of the set of variables $(X_1, \ldots, X_k)$ into blocks, $X^{[1]}, \ldots, X^{[\omega]}$, where the block $X^{[i]}$ is of size $k_i, 1 \leq i \leq \omega$. For $x^{[\omega]} \in R^{k_\omega}, \ldots, x^{[1]} \in R^{k_1}$, let

$$\mathrm{SIGN}_{\Pi,0}(\mathcal{P})(x^{[\omega]}, \ldots, x^{[1]}) = (\mathrm{sign}(P_1(x^{[\omega]}, \ldots, x^{[1]})),$$
$$\ldots, \mathrm{sign}(P_s(x^{[\omega]}, \ldots, x^{[1]}))).$$

For $x^{[i+1]} \in R^{k_{i+1}}, \ldots, x^{[\omega]} \in R^{k_\omega}$, and for all i, $0 < i \leq \omega$, we recursively define,

$$\mathrm{SIGN}_{\Pi,i}(\mathcal{P})(x^{[\omega]}, \ldots, x^{[i+1]}) =$$
$$\{\mathrm{SIGN}_{\Pi,i-1}(\mathcal{P})(x^{[\omega]}, \ldots, x^{[i+1]}, x^{[i]}) | x^{[i]} \in R^{k_i}\}.$$

Finally, we define the *total list of signs of* $\mathcal{P}$ with respect to the partition Π of the variables into the blocks, $(X^{[\omega]}, \ldots, X^{[1]})$, as

$$\mathrm{SIGN}_\Pi(\mathcal{P}) = \mathrm{SIGN}_{\Pi,\omega}(\mathcal{P}).$$

A given formula,

$$(Q_\omega X^{[\omega]})(Q_{\omega-1} X^{[\omega-1]}) \ldots (Q_1 X^{[1]}) F(P_1, \ldots, P_s),$$

can be decided easily from the set $\mathrm{SIGN}_\Pi(\mathcal{P})$. The number of evaluations of the Boolean formula F that is required is no more than the size of the set $\mathrm{SIGN}_\Pi(\mathcal{P})$.

We prove a bound on the size of the set $\mathrm{SIGN}_\Pi(\mathcal{P})$, in the following theorem which is the key for proving theorem 1.

Theorem 2 *Let Π denote a partition of the set of variables $(X_1, \ldots, X_k)$ into blocks, $X^{[1]}, \ldots, X^{[\omega]}$, where the block $X^{[i]}$ is of size $k_i, 1 \leq i \leq \omega$. Then the size of the set $\mathrm{SIGN}_\Pi(\mathcal{P})$ is bounded by $s^{\Pi(k_i+1)} d^{\Pi O(k_i)}$. Moreover, there exists a well-behaved algorithm which computes this set using $s^{\Pi(k_i+1)} d^{\Pi O(k_i)}$ number of arithmetic operations in D.*

The proof of the theorem follows from the correctness of our algorithm for doing block elimination described previously.

In order to construct the quantifier-free formula, we repeat the block elimination phase ω times. The output of the last iteration, is a set of polynomials $\mathcal{Q}_\omega(Y)$. We compute sample points in every cell of $\mathcal{Q}_\omega(Y)$, and for each such $y \in R^l$, we compute the set $SIGN_{\Pi,\omega}(y)$.

The size of the set of sample points is bounded by $s^{(l+1)\Pi(k_i+1)} d^{(l+1)\Pi O(k_i)}$.

The given formula $\Phi(y)$ can be decided from the set $SIGN_{\Pi,\omega}(y)$. We denote by T, the set of points y computed in the last step, for which $\Phi(y)$ is true. For each $y \in T$, we construct a formula $\Psi_y(Y)$, such that for any $z \in R^l$, $\Psi(z)$ is true iff the sign vector of the family of polynomials $\mathcal{P}(y)$ and $\mathcal{P}(z)$ are the same at the parametrized points generated during the elimination phase, with the parameters specialized to y and z respectively.

The quantifier free formula can then be written as $\Psi(Y) = \vee_{y \in T} \Psi_y(Y)$. In order to construct the formulae Ψ_y, we make use of the multivariate version ([24]) of the sign determination algorithm due to Ben Or, Kozen and Reif, in an inverse fashion. We call this the *Inverse Sign Determination Subroutine*. Roughly, the idea is the following.

2.3 The Inverse Sign Determination Subroutine

We assume that the reader is familiar with the *Sign Determination Algorithm* ([27],[24], inspired by[5]).

Given a multivariate polynomial h, and another system of polynomials, T, having only a finite number of real zeros, denoted by $Z(T)$, we define the *Sturm query* of h with respect to T by

$$SQ(T,h) = |\{x \in Z(T),\ h(x) > 0\}|$$
$$-|\{x \in Z(T),\ h(x) < 0\}|.$$

The *Sign Determination Algorithm* consists of a combinatorial part reducing Sign Determination to computing various Sturm queries and an algebraic part for computing the Sturm queries.

Given a family of s polynomials in k variables, and another system of polynomials T having N complex zeros, the multivariate *Sign Determination Algorithm* computes the list of sign conditions of the given family of polynomials, at the real zeros of T using $O(sN^2)$ Sturm queries. If we know the signs of the family of polynomials, at the real zeros of T, then we can simulate the *Sign Determination Algorithm*, since the branches that occur only depend on this information and not on the particular coefficients of the polynomials. However, this means that we know the particular Sturm queries made and also the answers to these queries. The Sturm query computation, (following [24]) involves computing the sign variations of the coefficients of the characteristic polynomial of a matrix. The matrix is of size $N \times N$ and its characteristic polynomial can be computed using $N^{O(1)}$ arithmetic operations in D. The details can be found in [24] or in the full paper [4].

We compute these coefficients, which are polynomials in Y, and also compute the set of realizable sign conditions for these polynomials. We output those sign conditions which give the right answer to the Sturm query as a disjunction of conjunctive Boolean formulae. The conjunction of these formulae over all the Sturm queries required by the *Sign Determination Algorithm*, gives us the formula Ψ_y. Actually, the polynomials in the atoms of Ψ_y contain the extra variables δ_i, introduced by the *Parametrized Sample Points Subroutine*. However, by expressing the polynomials in powers of these variables, and making use of the ordering amongst them, we can replace each inequality, by an equivalent larger formula involving only Y, and this does not affect the size of the output formula asymptotically. Note, that the number of Sturm queries required by the *Sign Determination Algorithm* is bounded by $sN^{O(1)}$, where N is the number of zeros of the input system of polynomial equations. In our application, $N = d^{\Pi O(k_i)}$. Moreover, each Sturm query computation produces $d^{\Pi O(k_i)}$ number of polynomials in Y. Together, these bounds imply the bounds in Theorem 1.

3 A Few Applications

We now give a few important consequences of the techniques developed above.

3.1 Small Ball

First we consider the problem of bounding, the radius of a ball centered at the origin which is guaranteed to intersect every cell of a finite family of polynomials, $\mathcal{P} = \{P_1, \ldots, P_s\} \subset Z[X_1, \ldots, X_k]$, whose degrees are bounded by d.

We have the following theorem :

Theorem 3 *Given a family $\mathcal{P}$ of s polynomials of degree d in k variables with coefficients in Z of bit length at most τ, then there exists a ball of radius $2^{O(\tau \times d^{O(k)})}$ intersecting every cell of $\mathcal{P}$.*

Corollary 3 *Given a family $\mathcal{P}$ of s polynomials of degree d in k variables with coefficients in Z of bit length at most τ such that*

$$C = \{x \in R^k \mid P_i(x) > 0\}$$

is a non-empty sign condition, there exists a rational number $\frac{a}{b}$ with a and b of bit length $\tau \times d^{O(k)}$ in every connected component of C.

This result improves the bound in [19], where the dependence on k was doubly exponential as well as the bound in [14] which depended on s.

3.2 Real and Complex Decision Problems

As applications of our techniques, we give improved algorithms for several problems in real and complex geometry. We can prove the following theorems :

Theorem 4 *Let $\mathcal{V} = Z(\mathcal{Q})$ be a variety where $\mathcal{Q}$ is a finite set of m polynomials in $D[X_1,\ldots X_k]$ of degree at most d. There is an algorithm whose input is $Q = \sum_{\{i=1,\ldots,m\}} Q_i^2$, which determines if the real dimension of the variety $\mathcal{V}$ is 0 in time $d^{O(k)}$.*

If the real dimension of the variety $\mathcal{V}$ is 0, the algorithm also outputs a univariate representation of the real points in $\mathcal{V}$ in $d^{O(k)}$ arithmetic operations in D. Moreover, let $\mathcal{P} = \{P_1,\ldots,P_s\}$ be a family of polynomials in k variables each of degree at most d and each with coefficients in an ordered domain D contained in a real closed field R. The signs of all the polynomials at the points of $\mathcal{V}$ can be computed in $sd^{O(k)}$ arithmetic operations in D.

Proposition 6 *Let V be an algebraic variety V defined as the zero set of m polynomials of degree d in k variables with coefficients in D. Then the coordinates of the isolated points of V are zeroes of polynomials of degrees $O(d)^k$. Moreover if $D = Z$ and the bit size of coefficients is bounded by τ, these points are contained in a ball of radius $\frac{a}{b}$ with a and b in Z of bit size $(\tau + log(m))d^{O(k)}$.*

Our techniques can be applied to decision problems in complex geometry improving slightly the results of [12] and [20], since we achieve the same complexity with deterministic algorithms, rather than with probabilistic ones.

Proposition 7 *Given a set $\mathcal{P}$ m polynomials of degree d in k variables with coefficients in $D[i]$, we can decide in $md^{O(k)}$ arithmetical operations over D whether the set of zeros of $\mathcal{P}$ is empty in $R[i]^k$.*

Proposition 8 *Given a set $\mathcal{P}$ of m polynomials of degree d in k variables with coefficients in $D[i]$, we can decide in $md^{O(k)}$ arithmetical operations over D whether the set of zeroes of $\mathcal{P}$ is zero-dimensional in $R[i]^k$.*

Proposition 9 *Given an algebraic variety V defined as the zero set of of a family $\mathcal{P}$ of m polynomials of degree d in k variables with coefficients in $D[i]$, the real and imaginary part of the coordinates of the isolated points of V are zeroes of polynomials of degrees $O(d)^k$. Moreover if $D = Z$ and the bit size of coefficients of the polynomials in $\mathcal{P}$ is bounded by τ, these points are contained in a ball of radius $\frac{a}{b}$ with a and b in Z of bit size $(\tau + log(m))d^{O(k)}$ in $R^{2k} = R[i]^k$.*

References

[1] M. E. Alonso, E. Becker, M.-F. Roy. T. Wormann *Zeros, Multiplicities and Idempotents for Zero Dimensional Systems* to appear in MEGA 94.

[2] 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", B. Caviness and J. Johnson Eds., Springer–Verlag, to appear.

[3] S. Basu, R. Pollack, M.-F. Roy *On the number of cells defined by a family of polynomials on a variety.* " Algorithmic foundations of robotics", K.Y. Goldberg, D. Halperin, J.-C. Latombe, R.H. Wilson, Eds., A.K. Peters, Boston, to appear.

[4] S. Basu, R. Pollack, M.-F. Roy *On the combinatorial and algebraic complexity of Quantifier Elimination.* Preprint.

[5] M. Ben-Or, D. Kozen , J. Reif The complexity of elementary algebra and geometry. *J. of Computer and Systems Sciences*, 32:251– 264, (1986).

[6] J. Bochnak, M. Coste, M.-F. Roy *Géométrie algébrique réelle.* Springer-Verlag (1987).

[7] B. Buchberger *Gröbner bases: an algorithmic method in polynomial ideal theory.* Recent trends in multidimensional systems theory, Reider ed. Bose, (1985).

[8] J. Canny *Some Algebraic and Geometric Computations in PSPACE,* Proc. Twentieth ACM Symp. on Theory of Computing, 460-467, (1988).

[9] J. Canny, *Improved algorithms for sign determination and existential quantifier elimination,* the Computer Journal, 36:409–418, (1993).

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

[11] M. Coste, M.-F. Roy *Thom's lemma, the coding of real algebraic numbers and the topology of semi-algebraic sets.* J. of Symbolic Computation 5 121- 129 (1988).

[12] M. Giusti, J. Heintz La détermination des points isolés d'une variété algébrique peut se faire en temps polynomial Proc. Int. Meeting on Commutative Algebra, Cortona (1991) .

[13] D. Grigor'ev, *The Complexity of deciding Tarski algebra* Journal of Symbolic Computation 5 (1988), 65-108.

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

[15] D. Grigor'ev, N. Vorobjov *Counting connected components of a semi-algebraic set in subexponential time.* Comput. Complexity, 2 :133-186 (1992).

[16] J. Heintz, M.-F. Roy, P. Solernò *Sur la complexité du principe de Tarski-Seidenberg.* Bull. Soc. Math. France 118 101-126 (1990).

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

[18] J. Heintz, M.-F. Roy, and P. Solernó *Single Exponential Path Finding in Semi-Algebraic sets: Part II: The general case*, Algebraic geometry and its applications, C. Bajaj editor, Springer Verlag (1993) 467-481.

[19] P. Koiran A Weak Version of the Blum, Shub & Smale model Proc. 34th IEEE Symp. on Foundations of Computer Science 486-495 (1993).

[20] T. Krick, L. M Pardo A Computational Method for Diophantine Approximation To appear in MEGA 94 .

[21] R. Loos *Generalized polynomial remainder sequences*, Computer Algebra : Symbolic and Algebraic Computation, B. Buchberger, G. E. Collins, R. Loos editors. Springer–Verlag, (1983) 115-138.

[22] B. Mishra *Algorithmic Algebra*, Springer-Verlag, (1993).

[23] R. Pollack, M.-F. Roy *On the number of cells defined by a set of polynomials*, C. R. Acad. Sci. Paris, 316:573–577, (1993).

[24] P. Pedersen, M.-F. Roy, A. Szpirglas *Counting real zeroes in the multivariate case*, Computational algebraic geometry, Eyssette et Galligo ed. Progress in Mathematics 109, 203-224, Birkhauser (1993).

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

[26] J. Renegar *Recent progress on the complexity of the decision problem for the reals*, Discrete and Computational Geometry: Papers from the DIMACS Special Year. DIMACS series in Discrete Mathematics and Theoretical Computer Science, AMS-ACM. 6:287–308, (1991).

[27] M.-F. Roy, A. Szpirglas *Complexity of computations with real algebraic numbers*, J. of Symbolic Comput., 10:39–51, (1990).

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

[29] A. Tarski *A Decision method for elementary algebra and geometry*, University of California Press (1951).

[30] B. L. Van Der Waerden *Modern Algebra, Volume II*, F. Ungar Publishing Co. (1950).