Verified Error Bounds for Real Solutions of Positive-dimensional Polynomial Systems *

Zhengfeng Yang
Shanghai Key Laboratory of Trustworthy Computing
East China Normal University, Shanghai 200062, China
zfyang@sei.ecnu.edu.cn

Lihong Zhi
Key Laboratory of Mathematics Mechanization
Chinese Academy of Sciences
Beijing 100190, China
lzhi@mmrc.iss.ac.cn

Yijun Zhu
Shanghai Key Laboratory of Trustworthy Computing
East China Normal University, Shanghai 200062, China
yjzhu@ecnu.cn

ABSTRACT

In this paper, we propose two algorithms for verifying the existence of real solutions of positive-dimensional polynomial systems. The first one is based on the critical point method and the homotopy continuation method. It targets for verifying the existence of real roots on each connected component of an algebraic variety $V \cap \mathbb{R}^n$ defined by polynomial equations. The second one is based on the low-rank moment matrix completion method and aims for verifying the existence of at least one real roots on $V \cap \mathbb{R}^n$. Combined both algorithms with the verification algorithms for zero-dimensional polynomial systems, we are able to find verified real solutions of positive-dimensional polynomial systems very efficiently for a large set of examples.

Categories and Subject Descriptors: G.4 [Mathematics of computing]: Mathematical Software; I.1.2 [Symbolic and Algebraic Manipulation]: Algorithms;
General Terms: Algorithms, experimentation
Keywords: positive-dimensional polynomial systems, real solutions, verification, error bounds.

1. INTRODUCTION

Let $F(\mathbf{x}) = [f_1, \ldots, f_m]^T$ be a polynomial system in $\mathbb{Q}[\mathbf{x}] = \mathbb{Q}[x_1, \ldots, x_n]$, and $V \subset \mathbb{C}^n$ be the algebraic variety defined by:

$$f_1(x_1, \ldots, x_n) = \cdots = f_m(x_1, \ldots, x_n) = 0. \tag{1}$$

We are interested in verifying the existence of real solutions on $V \cap \mathbb{R}^n$.

Suppose $I = \langle f_1, \ldots, f_m \rangle$ is a radical ideal and V is equidimensional, i.e., the irreducible components of V have same dimensions, then a point $\hat{\mathbf{x}} \in V$ is called a *regular point* of V, or V is called smooth at $\hat{\mathbf{x}}$ if and only if the rank of the Jacobian matrix $F_{\mathbf{x}}(\hat{\mathbf{x}})$ satisfies

$$\dim V = n - \operatorname{rank}(F_{\mathbf{x}}(\hat{\mathbf{x}})). \tag{2}$$

The set V_{reg} of regular points of V is called the *regular locus* of V. A point $\hat{\mathbf{x}}$ is called *singular* at V if and only if

$$\operatorname{rank}(F_{\mathbf{x}}(\hat{\mathbf{x}})) < n - \dim V. \tag{3}$$

The set $V_{sing} := V \backslash V_{reg}$ is called the *singular locus* of V. If all points on V are regular, then V is called *smooth*.

Remark 1 If $I = \langle f_1, \ldots, f_m \rangle$ is not a radical ideal, then a point $\hat{\mathbf{x}} \in V$ is called a *regular point* of V if and only if the rank of the Jacobian matrix $G_{\mathbf{x}}(\hat{\mathbf{x}})$ satisfies $\dim V = n - \operatorname{rank}(G_{\mathbf{x}}(\hat{\mathbf{x}}))$, where $G(\mathbf{x}) = [g_1, \ldots, g_s]^T$ is a polynomial basis of $\sqrt{I}$.

Computing real roots of a polynomial system is a fundamental problem of computational real algebraic geometry. There are symbolic methods based on Cylindrical Algebraic Decomposition [12] and critical point methods [5, 8, 15, 16, 19, 31, 38] for finding real points on the variety $V \cap \mathbb{R}^n$. Algorithms proposed in [1, 3, 4, 33, 37] find at least one real point on each connected component of $V \cap \mathbb{R}^n$. Recent work for computing verified real roots based on homotopy methods include certified homotopy-tracking method in [6], certifying solutions to polynomial systems using Smale's α-theorem [18].

A square zero-dimensional polynomial system . Suppose $F(\mathbf{x})$ is a square and zero-dimensional polynomial system, i.e., $m = n$. Standard verification methods for nonlinear square systems are based on the following theorem [21, 30, 34].

Theorem 1 *Let $F(\mathbf{x}) : \mathbb{R}^n \to \mathbb{R}^n$ be a polynomial system, and $\tilde{\mathbf{x}} \in \mathbb{R}^n$. Let $\mathbb{IR}$ be the set of real intervals, and $\mathbb{IR}^n$ and $\mathbb{IR}^{n\times n}$ be the set of real interval vectors and real interval matrices, respectively. Given $\mathbf{X} \in \mathbb{IR}^n$ with $\mathbf{0} \in \mathbf{X}$ and $M \in \mathbb{IR}^{n\times n}$ satisfies $\nabla f_i(\tilde{\mathbf{x}} + \mathbf{X}) \subseteq M_{i,:}$, for $i = 1, \ldots, n$. Denote by I_n the $n \times n$ identity matrix and assume*

$$-F_{\mathbf{x}}^{-1}(\tilde{\mathbf{x}})F(\tilde{\mathbf{x}}) + (I_n - F_{\mathbf{x}}^{-1}(\tilde{\mathbf{x}})M)\mathbf{X} \subseteq \operatorname{int}(\mathbf{X}), \tag{4}$$

where $F_{\mathbf{x}}(\tilde{\mathbf{x}})$ is the Jacobian matrix of $F(\mathbf{x})$ at $\tilde{\mathbf{x}}$. Then there is a unique $\hat{\mathbf{x}} \in \mathbf{X}$ with $F(\hat{\mathbf{x}}) = 0$. Moreover, every matrix

*This material is based on work supported by a NKBRPC 2011CB302400, the Chinese National Natural Science Foundation under Grants 91118001, 60821002/F02, 60911130369(Zhi). This material is supported in part by the National Natural Science Foundation of China under Grants 91118007 and 61021004(Yang, Zhu).

ISSAC'13, June 26–29, 2013, Boston, Massachusetts, USA.

$\widetilde{M} \in M$ *is nonsingular. In particular, the Jacobian matrix* $F_{\mathbf{x}}(\hat{\mathbf{x}})$ *is nonsingular.*

The non-singularity of the Jacobian matrix $F_{\mathbf{x}}(\hat{\mathbf{x}})$ restricts the application of Theorem 1 to regular solutions of a square polynomial system. If $F_{\mathbf{x}}(\hat{\mathbf{x}})$ is singular and $\mathbf{x}$ is an isolated singular solution of $F(\mathbf{x})$, in [26, 27, 36], by adding smoothing parameters properly to $F(\mathbf{x})$, an extended regular and square polynomial system is generated for computing verified error bounds, such that a slightly perturbed polynomial system of $F(\mathbf{x})$ is guaranteed to possess an isolated singular solution within the computed bounds. The method in [29] can also be used to verify the isolated singular solutions.

Remark 2 There are two functions verifynlss and verifynlss2 in the INTLAB package implemented by Rump in Matlab [35]. The procedure verifynlss can be used to verify the existence of a simple root of a square and regular zero-dimensional polynomial system and verifynlss2 can be used to verify the existence of a double root of a slightly perturbed polynomial system of $F(\mathbf{x})$. If the polynomial system $F(\mathbf{x})$ has an isolated singular root with multiplicity larger than 2, then the function viss designed in [26, 27] and implemented by Li and Zhu in Matlab can be applied to obtain verified error bounds such that a slightly perturbed polynomial system of $F(\mathbf{x})$ is guaranteed to possess an isolated singular solution within the computed bounds.

An overdetermined zero-dimensional polynomial system . Suppose $F(\mathbf{x})$ is an overdetermined zero-dimensional polynomial system, i.e., $m > n$. A natural procedure for obtaining a square polynomial system from $F(\mathbf{x})$ is to pick up a full rank random matrix $A \in \mathbb{Q}^{n\times m}$ and form a square polynomial system $A \cdot F(\mathbf{x})$. According to [42, Theorem 13.5.1], we have the following theorem.

Theorem 2 *There is a nonempty Zariski open subset* $\mathcal{A} \in \mathbb{C}^{n\times m}$ *such that for every* $A \in \mathcal{A}$*, a solution of* $F(\mathbf{x})$ *is regular if and only if it is a nonsingular solution of the square system* $A \cdot F(\mathbf{x})$*. Moreover, if* $F(\mathbf{x})$ *is a zero-dimensional system, then* $A \cdot F(\mathbf{x})$ *is also a zero-dimensional system.*

According to Theorem 2, we can apply Theorem 1 to regular solutions of the square polynomial system $A\cdot F(\mathbf{x})$ and check whether the verified solution of $A \cdot F(\mathbf{x})$ is a solution of $F(\mathbf{x})$ by computing the residual of $F(\hat{\mathbf{x}})$ as an additional test, see also [18, Lemma 3.1]. If $F(\hat{\mathbf{x}})$ is small, with high probability, the verified real solution of $A \cdot F(\mathbf{x})$ is a real solution of $F(\mathbf{x})$.

A positive-dimensional polynomial system . Suppose $F(\mathbf{x})$ is a positive-dimensional polynomial system. It is clear that an underdetermined system $F(\mathbf{x})$ is a positive-dimensional system whose dimension is at least $n - m \geq 1$. A square polynomial system and an overdetermined system can also be positive-dimensional. In [9, 10], the authors transformed an underdetermined system into a regular square system by choosing m independent variables and setting $n - m$ remaining variables to be anchors, then they used a Krawczyk-type interval operator to verify the existence of the solutions of the transformed regular and square system. It is very impressive that they can verify a solution of a polynomial system with more than 10000 variables and 20000 equations with degrees as high as 100. More general methods using linear slices to reduce the underdetermined system to a square system were proposed in [40, 41, 42]. We notice that it is very important to choose independent variables and initial values for the dependent variables or linear slices. Especially, we might have a big chance to miss the real points because of the bad choice for values of some variables.

Example 1 Consider the polynomial *Vor2*, which appears in a problem studying Voronoi Diagram of three lines in $\mathbb{R}^3$ [13]. *Vor2* is a polynomial in five variables with degree 18. It has an infinite number of real solutions. Let us set four variables as rational numbers chosen in the range $[-\frac{3000}{1000}, \frac{3000}{1000}]$, e.g.

$$\hat{x}_2 = \frac{177}{500},\ \hat{x}_3 = \frac{423}{1000},\ \hat{x}_4 = \frac{209}{1000},\ \hat{x}_5 = \frac{143}{50},$$

the univariate polynomial $V(x_1) = \mathit{Vor2}\,(x_1, \hat{x}_2, \hat{x}_3, \hat{x}_4, \hat{x}_5) \in \mathbb{Q}[x_1]$ has no real solutions.

Remark 3 If there is only one polynomial $f(x_1, \ldots, x_n)$ and the degree of f with respect to the variable x_i is odd, the univariate polynomial $f(\hat{x}_1, \ldots, \hat{x}_{i-1}, x_i, \hat{x}_{i+1}, \ldots, \hat{x}_n)$ will always have a real root $\hat{x}_i \in \mathbb{R}$ for arbitrary fixed values $\hat{x}_j \in \mathbb{Q}, 1 \leq j \leq n, j \neq i$. Hence, it is easy to verify that $(\hat{x}_1, \ldots, \hat{x}_{i-1}, \hat{x}_i, \hat{x}_{i+1}, \ldots, \hat{x}_n)$ is the real root of $f(\mathbf{x})$.

The main task of this paper is to construct a square and zero-dimensional polynomial system for computing verified real solutions of positive-dimensional polynomial systems. Let $I = \langle f_1, \ldots, f_m\rangle$ and V be an algebraic variety defined by $\{f_1 = 0, \ldots, f_m = 0\}$. We propose below different strategies for computing verified solutions on $V \cap \mathbb{R}^n$.

1. If the ideal I is radical and contains regular real solutions, we propose two algorithms for computing verified real solutions on $V \cap \mathbb{R}^n$:

   a. We use theoretical results developed in real algebraic geometry for finding one point on each connected component of $V \cap \mathbb{R}^n$ to construct a square and regular zero-dimensional polynomial system [1, 3, 4, 33, 37], then use the homotopy continuation solver HOM4PS-2.0 [24] to find its approximate real solutions. Finally, we apply verifynlss in the INTLAB package [35] to verify the existence of real solutions in the neighborhood of the computed approximate real solutions on connected components of $V \cap \mathbb{R}^n$.

   b. We compute an approximate real solution $\tilde{\mathbf{x}}$ of $F(\mathbf{x})$ by the low-rank moment matrix completion method in [28]. If the Jacobian matrix $F_{\mathbf{x}}(\tilde{\mathbf{x}})$ is singular, we compute a normalized null vector $\mathbf{v}$ of $F_{\mathbf{x}}(\tilde{\mathbf{x}})$ and add new polynomials $\sum_{j=1}^{m} \mathbf{v}_i \frac{\partial f_j(\mathbf{x})}{\partial x_i}$ for $1 \leq i \leq n$ to $F(\mathbf{x})$. Otherwise, we choose a normalized random vector λ and add polynomials $F_{\mathbf{x}}(\mathbf{x})\lambda - F_{\mathbf{x}}(\tilde{\mathbf{x}})\lambda$ to $F(\mathbf{x})$. Finally, we apply verifynlss to verify the existence of a real solution $\hat{\mathbf{x}}$ in the neighborhood of $\tilde{\mathbf{x}}$ on $V \cap \mathbb{R}^n$.

2. If the ideal I is not radical, we add tiny perturbations to the polynomial system $F(\mathbf{x})$ and modify above two algorithms accordingly.

c. The critical variety for the perturbed system is a zero-dimensional polynomial system containing not only regular solutions but also approximate singular solutions. For approximate singular solutions, we apply the verification algorithms verifynlss2 in [35] or viss in [26, 27] to compute verified error bounds, such that a slightly perturbed polynomial system of $F(\mathbf{x})$ possesses a real solution within the computed error bounds.

d. The real solutions computed by the method in [28] can be approximate singular solutions. We need to apply verification algorithms verifynlss2 or viss to compute verified error bounds of a slightly perturbed polynomial system.

Structure of the paper. In Section 2, we introduce theoretical results and methods for computing verified real solutions for positive-dimensional polynomial systems. In Section 3, we present three routines: verifyrealroot0 computes verified real solutions for zero-dimensional polynomial systems; verifyrealrootpc aims for computing verified real solutions on each connected components of $V \cap \mathbb{R}^n$; verifyrealrootpm is designed for computing at least one verified real solution for positive-dimensional polynomial systems. In Section 4, we demonstrate the effectiveness of the algorithms for computing verified real roots of a set of benchmark systems.

2. POSITIVE-DIMENSIONAL POLYNOMIAL SYSTEMS

2.1 The Radical Ideal Case

Let us consider the case where the ideal I generated by $f_1(\mathbf{x}), \ldots, f_m(\mathbf{x})$ is radical and V is of dimension d and contains a regular point in $\mathbb{R}^n$.

The critical point method

Theorem 3 *[4, Lemma 1]Let C be a connected component of the real variety V containing a regular point. Then, with respect to the Euclidean topology, there exists a non-empty open subset U_C of $\mathbb{R}^n \backslash V$ that satisfies the following condition: Let $\mathbf{u}$ be an arbitrary point of U_C and let $\hat{\mathbf{x}}$ be any point of V that minimizes the Euclidean distance to $\mathbf{u}$ with respect to V. Then $\hat{\mathbf{x}}$ is a regular point belonging to C.*

According to Theorem 3, one can compute a regular real sample point on V by computing its critical points of a distance function to a generic point restricted to V. This method was proposed in [1, 32, 33], see also [2, 4, 7] for some recent results when $F(\mathbf{x})$ has real singular solutions. Let us briefly introduce the method in [1].

Definition 1 *[1, Notation 2.4] For an arbitrary point $\mathbf{u} = (u_1, \ldots, u_n) \in \mathbb{R}^n$, let $g = \frac{1}{2}(x_1 - u_1)^2 + \cdots + \frac{1}{2}(x_n - u_n)^2$ and*

$$J_g(F) = \begin{bmatrix} \frac{\partial f_1}{\partial x_1} & \cdots & \frac{\partial f_m}{\partial x_1} & \frac{\partial g}{\partial x_1} \\ \vdots & & \vdots & \vdots \\ \frac{\partial f_1}{\partial x_n} & \cdots & \frac{\partial f_m}{\partial x_n} & \frac{\partial g}{\partial x_n} \end{bmatrix}. \tag{5}$$

We define the algebraic set:

$$C(V, \mathbf{u}) = \{\hat{\mathbf{x}} \in V, \operatorname{rank}(J_g(F(\hat{\mathbf{x}})) \leq n - d\}. \tag{6}$$

Let $\Delta_{\mathbf{u},d}(F)$ be the set of all the minors of order $n - d + 1$ in the matrix $J_g(F)$ such that their last column contains the entries in the last column of $J_g(F)$.

Theorem 4 *[1, Theorem 2.3] Let V be an algebraic variety of dimension d and I be a radical equidimensional ideal. If D is a large enough positive integer, there exists at least one point $\mathbf{u}$ in $\{1, \ldots, D\}^n$ such that:*

1. *$C(V, \mathbf{u})$ meets every semi-algebraically connected component of $V \cap \mathbb{R}^n$;*

2. *$C(V, \mathbf{u}) = V_{sing} \cap V_{0,\mathbf{u}}$, where $V_{0,\mathbf{u}}$ is a finite set of points in $\mathbb{C}^n$ and V_{sing} are singular points on V whose Jacobian matrix have rank less than $n - d$.*

Moreover,

$$\dim(C(V, \mathbf{u})) < \dim(V). \tag{7}$$

According to Theorem 4, for almost all $\mathbf{u}$, the dimension of the algebraic variety $C(V, \mathbf{u})$ of $\Delta_{\mathbf{u},d}(F) \cup F(\mathbf{x})$ will be strict less than the dimension of V. Therefore, inductively, we will obtain a zero-dimensional polynomial system which can be used to verify the existence of regular real solutions on V. As stated in [1, 32], the main bottleneck for the critical points method is the computation of $\Delta_{\mathbf{u},d}$ since the number of elements in $\Delta_{\mathbf{u},d}$ is equal to $\binom{m}{n-d}\binom{n}{n-d+1}$ and the polynomials in $\Delta_{\mathbf{u},d}$ are usually dense and have large coefficients. An alternative way to avoid the computation of the minors is to introduce extra variables $\lambda_0, \ldots, \lambda_{n-d}$ and pick up randomly $n - d$ real numbers $a_0, \ldots, a_{n-d}$ and polynomials in $F(\mathbf{x})$ such as $f_1, \ldots, f_{n-d}$, and replace the minors in $\Delta_{\mathbf{u},d}$ by polynomials defined below

$$p_i = \lambda_0 \frac{\partial g}{\partial x_i} + \lambda_1 \frac{\partial f_1}{\partial x_i} + \ldots + \lambda_{n-d} \frac{\partial f_{n-d}}{\partial x_i}, \quad \text{for } 1 \leq i \leq n,$$
$$p_{n+1} = a_0 \lambda_0 + \ldots + a_{n-d} \lambda_{n-d} - 1.$$

This is the way used in [17, Theorem 5] to generate solution paths leading to real solutions on V using the homotopy continuation method.

If V is compact and smooth, and the variables $x_1, \ldots, x_n$ are in a generic position with respect to $f_1, \ldots, f_m$, then as shown in [3, Theorem 10], one can change the distance function g to a coordinate function $g = x_i, 1 \leq i \leq n$ such that the dimension of the real variety of $\Delta_{\mathbf{u},d}(F) \cup F(\mathbf{x})$ will be zero and contains at least one real point on each connected component of $V \cap \mathbb{R}^n$. Moreover, in [37], Safey El Din and Schost extended the result in [3] to deal with the case where $V \cap \mathbb{R}^n$ is non-compact.

The low-rank moment matrix completion method. Recently, there is also an arising interest in using numerical semidefinite programming (SDP) based method [11, 20, 23] for characterizing and computing the real solutions of polynomial systems. As pointed out in [23], the great benefit of using SDP techniques is that it exploits the real algebraic nature of the problem right from the beginning and avoids the computation of complex components. For example, if $V \cap \mathbb{R}^n$ is zero-dimensional, then the moment-matrix algorithm in [23] can compute all real solutions of $F(\mathbf{x})$ by solving a sequence of SDP problems.

If the polynomial system $F(\mathbf{x})$ has an infinite number of real solutions, then the algorithm in [23] can not be used.

Hence, in [20, 22], they replaced the constant object function in (**??**) by the trace of the moment matrix and showed that their software GloptiPoly is very efficient for finding a partial set of real solutions for a large set of polynomial systems [22, Table 6.3, 6.4]. Since the trace of a semidefinite moment matrix is equal to its nuclear norm defined as the sum of its singular values, the optimization problem (**??**) can be transformed to the following nuclear norm minimization problem:

$$\begin{cases} \min & \|M_t(y)\|_* \\ \text{s. t.} & y_0 = 1, \\ & M_t(y) \succeq 0, \\ & M_{t-d_j}(f_j\, y) = 0, \quad j = 1, \ldots, m, \end{cases} \tag{8}$$

In [28], a new algorithm based on accelerated fixed point continuation method and alternating direction method was presented to solve the minimization problem (8) for finding real solutions of $F(\mathbf{x})$ even when its real variety $V \cap \mathbb{R}^n$ is positive-dimensional. Although the method based on function values and gradient evaluations cannot yield as high accuracy as interior point methods, much larger problems can be solved since no second-order information needs to be computed and stored.

Encouraged by the results shown in [28, Table1] and noted that the main bottleneck for the critical point method is the computation of $\Delta_{\mathbf{u},d}$, we explain below how to avoid the computation of minors by constructing a zero-dimensional polynomial system based on the approximate real solution $\tilde{\mathbf{x}}$ computed by the algorithm MMCRSolver in [28] for verifying the existence of real solutions in $V \cap \mathbb{R}^n$ in the neighborhood of $\tilde{\mathbf{x}}$ when V is positive-dimensional.

Suppose $\tilde{\mathbf{x}}$ is an approximate real root of $F(\mathbf{x})$ computed by MMCRSolver. If the rank of the Jacobian matrix $F_{\mathbf{x}}(\tilde{\mathbf{x}})$ is less than $n - d$, then $\tilde{\mathbf{x}}$ is a singular point on V. Stimulated by the deflation method used in [25] for constructing extended regular polynomial systems, we compute a normalized null vector $\mathbf{v}$ ($|\mathbf{v}|_2 = 1$) of $F_{\mathbf{x}}(\tilde{\mathbf{x}})$ and generate a new polynomial system $\widetilde{F}(\mathbf{x}) = F(\mathbf{x}) \cup F_{\mathbf{x}}(\mathbf{x})\mathbf{v}$. It is clear that $\tilde{\mathbf{x}}$ is a real solution of

$$\begin{cases} F(\mathbf{x}) & = & \mathbf{0}, \\ F_{\mathbf{x}}(\mathbf{x})\mathbf{v} & = & \mathbf{0}. \end{cases} \tag{9}$$

It is possible that $\tilde{\mathbf{x}}$ is still a singular solution of $\widetilde{F}$, then we can perform the similar deflations to the system $\widetilde{F}(\mathbf{x})$ again.

If the approximate solution $\tilde{\mathbf{x}}$ is not a singular point on the variety $V \cap \mathbb{R}^n$, i.e., the rank of the Jacobian matrix $F_{\mathbf{x}}(\tilde{\mathbf{x}})$ is $n - d$, then we choose a normalized random vector λ and construct a new polynomial system $\widetilde{F}(\mathbf{x}) = F(\mathbf{x}) \cup \{F_{\mathbf{x}}(\mathbf{x})\lambda - F_{\mathbf{x}}(\tilde{\mathbf{x}})\lambda\}$. It is clear that $\tilde{\mathbf{x}}$ is a solution of

$$\begin{cases} F(\mathbf{x}) & = & \mathbf{0}, \\ F_{\mathbf{x}}(\mathbf{x})\lambda - F_{\mathbf{x}}(\tilde{\mathbf{x}})\lambda & = & \mathbf{0}. \end{cases} \tag{10}$$

Suppose we obtain a zero-dimensional regular system $\widetilde{F}(\mathbf{x})$ after above steps, then we apply verifynlss [35] to verify the existence of a real solution $\hat{\mathbf{x}}$ in the neighborhood of $\tilde{\mathbf{x}}$ on $V \cap \mathbb{R}^n$. However, it is not guaranteed that the variety of the new polynomial system $\widetilde{F}(\mathbf{x})$ generated above will be a zero-dimensional regular system. In order to obtain a zero-dimensional regular system, we may need to add more random polynomials vanishing at $\tilde{\mathbf{x}}$.

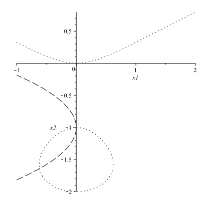
The verification algorithm based on using the null vector of $F_{\mathbf{x}}(\tilde{\mathbf{x}})$ or a random vector to construct new polynomial systems can be more efficient since it avoids the computations of minors or the introduction of new variables. However, since we only use local information about the approximate real root $\tilde{\mathbf{x}}$ of $F(\mathbf{x})$ in order to construct the new extended system, it is limited to verify the existence of a real root $\hat{\mathbf{x}}$ in the neighborhood of $\tilde{\mathbf{x}}$. For some interesting applications, it is enough to verify the existence of one real solution, e.g., for deciding reachability of the infimum of a multivariate polynomial [14], if we can verify the existence of one real solution for $f - f^*$, then we prove that f^* is a minimum which can be attained.

Example 2 [7, Example 4] To illustrate the above method, consider the polynomial $f(x_1, x_2) = x_1^2 - x_2\,(x_2+1)(x_2+2)$.

MMCRSolver yields one approximate real solution

$$\tilde{\mathbf{x}} = [3.671518 \times 10^{-8}, -0.999902]^T.$$

Since the approximate solution $\tilde{\mathbf{x}}$ is not a singular solution of $f(\mathbf{x})$, we choose a random vector $\lambda = [0.715927, -0.328489]^T$ and construct a new square polynomial system by adding one more polynomial g defined by λ and $\tilde{\mathbf{x}}$ in (10). The curves of f and g are displayed in the following figure with dot and dash line styles respectively.



We run the algorithm verifynlss and prove that $f(x_1, x_2)$ has a verified real solution within the inclusion

x_1	x_2
$4.3211387 \times 10^{-8} \pm 2.7 \times 10^{-15}$	$-1 \pm 2.2 \times 10^{-15}$

2.2 The Non-radical Ideal Case

If the ideal I generated by polynomials in $F(\mathbf{x})$ is not radical, as pointed out in [1], the inequality (7) in Theorem 4 is not true. It is difficult to verify the exact existence of real points on singular locus V_{sing} which might have the same dimension as V.

Example 3 [43, 5.15] Consider the system $F(\mathbf{x})$ containing polynomials $f_1 = x_1^3 x_3^2 + x_3, f_2 = x_1^2 x_2 + x_3$.

The ideal I generated by polynomials f_1, f_2 is not radical. The real algebraic variety $V \cap \mathbb{R}^n$ defined by $\{f_1 = 0, f_2 = 0\}$ contains three one-dimensional solutions $V_1 = \{x_1 = 0, x_3 = 0\}, V_2 = \{x_2 = 0, x_3 = 0\}, V_3 = \{x_1^5 x_2 - 1 = 0, x_3 x_1^3 + 1 = 0\}$.

Since the rank of the Jacobian matrix at all points on the variety V_1 is 1, we know that the variety $C(V, \mathbf{u})$ defined in (6) for an arbitrary chosen point $\mathbf{u}$ contains the one-dimensional variety V_1. Hence $\dim(C(V, \mathbf{u})) = \dim(V) = 1$, the inequality (7) in Theorem 4 is not true for this example.

Let us choose $\mathbf{u}$ as

$$\{u_1 = 1, u_2 = 2, u_3 = 3\}.$$

The set $\Delta_{\mathbf{u},1}(F)$ consists of the determinant of $J_g(F)$ defined by (5) in Theorem 4. Applying the homotopy solver HOM4PS-2.0 to the polynomial system $F(\mathbf{x}) \cup \Delta_{\mathbf{u},1}(F)$, we obtain 5 real approximate solutions of $C(V, \mathbf{u})$.

- The real solution $\{x_1 = 0, x_2 = 0, x_3 = 0\}$ is on $V_1 \cap C(V, \mathbf{u})$. It is not an isolated singular solution. Therefore, it can not be verified by verifynlss2 or viss.

  However, it is interesting to notice that there is another real root computed by HOM4PS-2.0 which is very near to V_1. Run the algorithms viss, we are able to compute the verified error bound, such that the slightly perturbed system (within 4.16×10^{-15}) of $F(\mathbf{x})$ has a verified real solution within the inclusion

x_1	x_2	x_3
0	$2 \pm 4.44 \times 10^{-16}$	0

- Applying the algorithm verifynlss to other three approximate real roots computed by HOM4PS-2.0, we obtain:

  – two verified regular real solutions within inclusions on the component V_3,

x_1	x_2	x_3
$1.7 \pm 2 \times 10^{-15}$	$0.07 \pm 5 \times 10^{-15}$	$-0.2 \pm 8 \times 10^{-16}$
$-1.1 \pm 7 \times 10^{-16}$	$-0.57 \pm 1 \times 10^{-15}$	$0.71 \pm 1 \times 10^{-15}$

  – one verified regular real solution within the inclusion on the component V_2,

x_1	x_2	x_3
$1 \pm 4.4440892 \times 10^{-16}$	0	0

If I is not radical, a well-known method to get a smooth algebraic variety is to add one or more infinitesimal deformations to polynomials in $F(\mathbf{x})$ and work over a non-archimedean real closed extension of the ground field [5, 33]. The computation could be quite expensive. Therefore, instead of proving the exact existence of real roots on the variety defined by a non-radical ideal, we perturb the system by a tiny real number and show the existence of real roots of this slightly perturbed polynomial system.

Theorem 5 *[17, Lemma 4] Suppose G consists of $n - d$ polynomials and $V(G)$ is a pure d-dimensional variety. There is a nonempty Zariski open set $Z \subset \mathbb{C}^{n-d}$ such that, for every $z \in Z$, $V(G - z)$ is a smooth algebraic set of dimension d.*

For $m = 1$, it is a well known consequence of Sard theorem, see [33, Lemma 3.5].

Let us add a small perturbation 10^{-25} to f_1 above and run the homotopy solver HOM4PS-2.0 for the perturbed system $\{f_1 + 10^{-25}, f_2\} \cup \Delta_{\mathbf{u},1}(F)$, we also obtain 5 approximate real solutions on $C(V, \mathbf{u})$. The algorithm verifynlss computes inclusions of three real solutions near to V_2 and V_3:

x_1	x_2	x_3
$1 \pm 3 \times 10^{-16}$	0	0
$1.7 \pm 2 \times 10^{-15}$	$0.07 \pm 5 \times 10^{-15}$	$-0.2 \pm 8 \times 10^{-16}$
$-1.1 \pm 7 \times 10^{-16}$	$-0.57 \pm 1 \times 10^{-15}$	$0.71 \pm 1 \times 10^{-15}$

Applying the algorithm viss, we obtain inclusions for another two real solutions near to V_1:

x_1	x_2	x_3
$7.4 \times 10^{-9} \pm 3 \times 10^{-24}$	$1.8 \times 10^{-9} \pm 2 \times 10^{-24}$	0
0	$2 \pm 9 \times 10^{-16}$	0.

Remark 4 Notice here, the perturbed system is smooth, $C(V, \mathbf{u})$ is a zero-dimensional variety. However, it contains approximate singular solutions [29]. Hence, it is necessary to apply the algorithm viss to verify the existence of real singular solutions of a slightly perturbed system. Moreover, since computations in Matlab have limited precisions, with or without tiny perturbations, we may get similar results.

We can also apply MMCRSolver to obtain an approximate real solution of $F(\mathbf{x})$ near to $(0, 0, 0)$. Running the algorithms verifyrealrootpm and verifynlss2, we prove that the slightly perturbed system (within 10^{-58}) of $F(\mathbf{x})$ has a verified real solution within the inclusion

x_1	x_2	x_3
$1.35 \times 10^{-15} \pm 2 \times 10^{-30}$	$6.77 \times 10^{-15} \pm 9 \times 10^{-29}$	0

3. ALGORITHMS FOR COMPUTING VERIFIED SOLUTIONS OF POLYNOMIAL SYSTEMS

Based on discussions in above sections, we present three procedures: verifyrealroot0 is based on the verification algorithms verifynlss, verifynlss2 [35] and viss [26, 27], and computes verified real solutions for zero-dimensional polynomial systems; verifyrealrootpc is based on the critical point method and the homotopy continuation method, and aims for computing at least one verified real solution on each connected component of $V \cap \mathbb{R}^n$; verifyrealrootpm is based on the low-rank moment matrix completion method in [28], and aims for computing at least one verified real solution for positive-dimensional polynomial systems. Before we show the algorithms, we would like to point out that unlike symbolic methods [1, 32, 33, 37], our algorithms can not be used to verify the nonexistence of real solutions on $V \cap \mathbb{R}^n$, i.e., the failure of our algorithms does not mean there exist no real solutions on $V \cap \mathbb{R}^n$.

Remark 5 According to Theorem 4, suppose I is a radical equidimensional ideal, then the variety $C(V, \mathbf{u})$ meets every semi-algebraic connected component of $V \cap \mathbb{R}^n$. Furthermore, applying Theorem 4 recursively, one is able to obtain a zero-dimensional overdetermined system, which contains a regular real root on $V \cap \mathbb{R}^n$ if it is not empty. It should be pointed out that we do not perform the equidimensional decomposition of the ideal $I = \langle f_1, \ldots, f_m \rangle$. The function verifyrealrootpc does not guarantee to verify real roots on each connected component of $V \cap \mathbb{R}^n$ if V is not equidimensional.

Remark 6 If I is not radical, it is still possible to verify the existence of regular real solutions. However, as observed from the Example 3, the variety $C(V, \mathbf{u})$ may have singular locus with the same dimension as the variety V. Hence, we could only verify the existence of a singular real root near to the slightly perturbed polynomial system. Another possibility would be to perturb the polynomial system $F(\mathbf{x})$ at the beginning by a tiny number. We notice in the non-radical case, it is necessary to call verifynlss2 or viss for verifying the existence of a singular solution of a slightly perturbed polynomial system.

verifyrealroot0

Input: A zero-dimensional polynomial system $F(\mathbf{x}) = [f_1, \ldots, f_m]^T$ in $\mathbb{Q}[x_1, \ldots, x_n]$, a given small tolerance $\epsilon \in \mathbb{R}$.
Output: A set L of real root inclusions X.

1. If $m > n$, choose a random matrix $A \in \mathbb{Q}^{n\times m}$; otherwise, set $A = I_n$.
2. Set $\widetilde{F} = A \cdot F(\mathbf{x})$, apply MMCRSolver or HMO4PS-2.0 to obtain approximate real roots of $\widetilde{F}(\mathbf{x})$, denoted by $\tilde{\mathbf{x}}_1, \ldots, \tilde{\mathbf{x}}_k$.
3. Set $L = \{\}$. For $i = 1, \ldots, k$ do:
   (a) If the Jacobian matrix $\widetilde{F}_{\mathbf{x}}(\tilde{\mathbf{x}}_i)$ is regular
   call verifynlss to obtain the real root inclusion X of $\widetilde{F}(\mathbf{x})$, set $b = 0$.
   Otherwise
   call verifynlss2 or viss to obtain the verified error bound b and the real root inclusion X.
   (b) Compute the residue $\tau = F(X)$. If $\tau < \epsilon$ and $b < \epsilon$, set $L = L \cup \{X\}$.
4. Output L.

Figure 1: The Verification Algorithm for Zero-Dimensional Polynomial Systems.

verifyrealrootpc

Input: A positive-dimensional polynomial system $F(\mathbf{x}) = [f_1, \ldots, f_m]^T$ in $\mathbb{Q}[x_1, \ldots, x_n]$, a given small tolerance $\epsilon \in \mathbb{R}$.
Output: A set L of real root inclusions X.

1. Construct a zero-dimensional overdetermined system, denoted by $\widetilde{F}(\mathbf{x})$ via the critical point method.
2. Suppose the number of polynomials in $\widetilde{F}(\mathbf{x})$ is s, choose a random matrix $A = \mathbb{Q}^{n\times s}$ and update $\widetilde{F}(\mathbf{x})$ to be $A \cdot \widetilde{F}(\mathbf{x})$.
3. Apply HOM4PS-2.0 to obtain approximate real roots of $\widetilde{F}(\mathbf{x})$, denoted by $\tilde{\mathbf{x}}_1, \ldots, \tilde{\mathbf{x}}_k$.
4. Run Step 3 of verifyrealroot0 for $\widetilde{F}(\mathbf{x})$ and $\tilde{\mathbf{x}}_1, \ldots, \tilde{\mathbf{x}}_k$.
5. Output L.

Figure 2: The Verification Algorithm for Positive-Dimensional Polynomial Systems Based on the Critical Point Method and the Homotopy Continuation Method

Remark 7 The polynomial system $\widetilde{F}(\mathbf{x})$ generated in Step 2(a) of verifyrealrootpm is not guaranteed to be of zero dimension. If $\widetilde{F}(\mathbf{x})$ is still of positive dimension and Step 2(c) fails for $\tilde{\mathbf{x}}_i$, then we add more polynomials vanishing at $\tilde{\mathbf{x}}_i$ to $\widetilde{F}(\mathbf{x})$.

verifyrealrootpm

Input: A positive-dimensional polynomial system $F(\mathbf{x}) = [f_1, \ldots, f_m]^T$ in $\mathbb{Q}[x_1, \ldots, x_n]$, a given small tolerance $\epsilon \in \mathbb{R}$.
Output: A set L of real root inclusions X.

1. Apply MMCRSolver to obtain approximate real roots of $F(\mathbf{x})$, denoted by $\tilde{\mathbf{x}}_1, \ldots, \tilde{\mathbf{x}}_k$.
2. Set $L = \{\}$. For $i = 1, \ldots, k$ do:
   (a) i. If $F_{\mathbf{x}}(\tilde{\mathbf{x}}_i)$ is singular
   compute a normalized null vector $\mathbf{v}$ of $F_{\mathbf{x}}(\tilde{\mathbf{x}}_i)$
   $\widetilde{F}(\mathbf{x}) = \left\{\sum_{j=1}^{m} \mathbf{v}_i \frac{\partial f_j(\mathbf{x})}{\partial x_i},\ 1 \le i \le n\right\} \cup F(\mathbf{x})$.
   ii. Otherwise
   compute a normalized random vector λ
   $\widetilde{F}(\mathbf{x}) = F(\mathbf{x}) \cup \{F_{\mathbf{x}}(\mathbf{x})\lambda - F_{\mathbf{x}}(\tilde{\mathbf{x}}_i)\lambda\}$
   (b) Choose a random matrix $A = \mathbb{Q}^{n\times(m+n)}$, update $\widetilde{F}$ to be $A \cdot \widetilde{F}$.
   (c) Run Step 3(a)(b) of verifyrealroot0 for $\widetilde{F}(\mathbf{x})$ and $\tilde{\mathbf{x}}_i$.
3. Output L.

Figure 3: The Verification Algorithm for Positive-dimensional Polynomial Systems Based on Low-rank Moment Matrix Completion Method.

Remark 8 In verifyrealrootpc and verifyrealrootpm, if the polynomial system $F(\mathbf{x})$ is underdetermined, i.e., $m < n$, our first choice of the random matrix A will always have the block structure $\begin{bmatrix} I_m & 0 \\ 0 & A_{sub} \end{bmatrix}$, where A_{sub} is chosen randomly. In this case, we do not need to compute the residue. The verified solution of $\widetilde{F}$ will be a verified solution of $F(\mathbf{x})$.

4. EXPERIMENTS

Our algorithms have been implemented in Matlab (2011R) and the performance is reported in the following tables. All examples are run on Intel(R) Core(TM) at 2.6GHz under Windows. We also translate the Maple codes of MMCRSolver [28] and viss [26, 27] into Matlab codes. The codes can be downloaded from http://www.mmrc.iss.ac.cn/~lzhi/Research/hybrid/VerifyRealRoots/

In Table 1, we exhibit the performance of the algorithm verifyrealroot0 for computing verified real solutions of zero-dimensional polynomial systems. All problems are taken from the homepage of Jan Verschelde http://www.math.uic.edu/~jan/. Here *var* and *deg* denote the number of the variables and the highest degree of polynomials; *ctrs* denotes the number of the equations; verifyrealroot0(M) and verifyrealroot0(H) refer to the two methods based on the low-rank moment matrix completion method and the homotopy method respectively for computing approximate roots in verifyrealroot0; *sol* denotes the number of the verified solutions; *time* is given in seconds for computing verified real solutions; whereas *width* denotes the largest of widths of all verified solutions computed by our algorithms.

In Table 2, Table 3 and Table 4, we exhibit the performance of our algorithms on positive dimensional polynomial

problem	var	deg	verifyrealroot0(M)			verifyrealroot0(H)		
			time	*sol*	*width*	*time*	*sol*	*width*
cohn2	4	6	10.8	1	6.3e-29	20.1	3	6.5e-12
cohn3	4	6	24.7	1	2.4e-26	137	5	2.9e-9
comb3000	10	3	1.56	1	2.0e-20	1.38	4	2.7e-20
d1	12	3	52.3	2	1.7e-14	6.24	16	1.8e-14
boon	6	4	27.6	1	5.1e-15	1.98	8	2.9e-15
des22_24	10	2	1.79	1	2.5e-14	1.73	10	1.1e-8
discret3	8	2	51.5	1	1.3e-13	107	102	1.5e-14
geneig	6	3	6.53	2	6.7e-15	4.63	10	2.7e-13
heart	8	4	24.9	2	5.3e-15	1.40	2	4.9e-15
i1	10	3	1.23	1	1.7e-16	11.0	16	9.1e-08
katsura5	6	2	1.35	1	2.2e-16	3.26	12	1.8e-15
kin1	12	3	52.3	2	1.8e-14	5.91	16	1.8e-14
ku10	10	2	37.8	1	4.7e-14	0.96	2	6.7e-14
noon3	3	3	1.88	1	1.6e-16	11.7	8	1.6e-15
noon4	4	3	9.70	1	3.6e-15	30.2	22	3.9e-15
puma	8	2	5.85	2	2.9e-14	3.99	16	1.8e-13
quadfor2	4	4	1.48	2	5.6e-16	0.71	2	2.2e-16
rbp1	6	3	5.59	1	2.6e-15	23.2	4	8.4e-14
redeco5	5	2	0.95	1	8.3e-17	1.07	4	1.3e-15
reimer5	5	6	26.7	3	8.4e-14	5.83	24	3.2e-13

Table 1: Algorithm Performance on Zero-dimensional Polynomial Systems

systems. The symbol $\triangle$ denotes the singular solutions verified by verifynlss2 or viss; the symbol $*$ denotes that the verified solution is a real solution of the original polynomial system with high probability. *curve0-5* are examples from [7]; *ex4* and *ex5* are cited from [33]; *Vor2* is from [13]; the remaining examples are taken from the homepage of Jan Verschelde and the polynomial test suite of D. Bini and B. Mourrain http://www-sop.inria.fr/saga/POL/.

problem	var	ctrs	deg	verifyrealrootpm			verifyrealrootpc		
				time	*sol*	width	*time*	*sol*	*width*
curve1	2	1	6	3.43	1	1.1e-14	4.13	9	5.0e-14
curve2	2	1	12	8.87	1	9.5e-20	160	27, $11_\triangle$	5.2e-10
curve3	2	1	6	2.02	1	8.7e-15	20.0	8	2.1e-14
curve4	2	1	3	1.26	1	8.7e-15	3.96	3	3.3e-15
curve5	2	1	6	6.11	$1_\triangle$	4.9e-16	12.9	4	4.7e-11
ex4	3	1	5	4.72	1	5.0e-14	46.6	10	6.0e-12
ex5	4	1	4	5.13	1	1.4e-26	122	46, $2_\triangle$	2.2e-9
adjmin22e4	6	2	2	9.86	1	9.0e-30	234	14, $22_\triangle$	1.8e-12
butcher	4	2	3	3.41	1	8.9e-15	319	30	1.7e-12
gerdt2	5	3	4	4.82	1	1.6e-15	506	31	1.2e-10

Table 2: Algorithm Performance on Positive-dimensional Polynomial Systems

The algorithm verifyrealrootpc is designed for computing the verified solutions on each connected component on V by adding all minors in $\Delta_{n,d}$. However, it is well-known that polynomials in $\Delta_{n,d}$ are usually dense and have large coefficients. It is difficult for HOM4PS-2.0 to handle large polynomials in Matlab. Therefore, verifyrealrootpc can only find successfully verified real solutions for polynomial systems in Table 2. In order to apply verifyrealrootpc to polynomial systems in Table 3, we use the most possible canonical projections, i.e., fixing as many variables as possible, to construct the zero-dimensional polynomial system. The modified version of verifyrealrootpc is denoted as verifyrealrootpc*. Therefore, in Table 3, both algorithms are aiming only for verifying the existence of at least one real root of polynomial systems.

problem	var	ctrs	deg	verifyrealrootpm			verifyrealrootpc*		
				time	*sol*	width	*time*	*sol*	*width*
vor2	5	1	18	19.9	$1_\triangle$	3.2e-11	587	$1_\triangle$	1.7e-6
curve0	2	1	12	9.28	$3_\triangle$	3.9e-15	10.8	$4_\triangle$	4.4e-16
birkhoff	4	1	10	127	$1_\triangle$	2.2e-26	7.72	7	1.0e-14
adjmin23e5	8	3	2	1.24	1	2.3e-28	1.09	1	7.8e-16
adjmin24e6	10	4	2	1.68	1	4.8e-28	1.46	1	1.1e-15
adjmin25e7	12	5	2	6.19	1	3.7e-27	1.68	1	6.2e-15
adjmin26e8	14	6	2	4.05	1	3.3e-29	2.32	1	3.1e-15
adjmin27e9	16	7	2	3.51	1	1.0e-29	1.98	1	2.7e-15
adjmin28eA	18	8	2	26.6	1	3.9e-29	3.29	1	3.3e-15
adjmin29eB	20	9	2	6.39	1	2.3e-29	9.22	1	4.0e-15
geddes2	5	4	6	18.9	1	5.8e-14	5.43	11	3.6e-11
geddes3	11	2	3	2.58	1	5.5e-28	1.26	1	7.1e-15
geddes4	12	3	3	3.05	1	1.3-27	1.34	1	7.1e-15
hairer1	8	6	3	2.06	1	1.2e-14	1.25	1	5.8e-15
hairer2	9	7	4	244	3	1.3e-12	17.7	6	9.3e-12
lanconelli	8	2	3	5.38	1	6.7e-15	1.48	2	4.9-13
bronestein2	4	3	4	14.7	$1_\triangle$	1.3e-25	3.18	2	3.8e-15
hawesl	5	4	9	16.1	$1_\triangle$	4.5e-19	2.09	1	3.6e-14
raksanyi	8	4	3	2.47	1	1.4e-19	1.69	2	1.2e-15
spatburmel	6	5	2	11.9	1	5.9e-15	3.92	2	1.6e-12

Table 3: Algorithm Performance on Positive-dimensional Polynomial Systems

In Table 4 we show the performance of our algorithms on non-radical polynomial systems cited from [40, 43, 44]. Let *pert.* denote the real number added to the original polynomial system. It should be noticed that only a very limited number of small non-radical polynomial systems are tested above. We are working on providing a more reliable algorithm for certifying real roots of non-radical polynomial systems.

Ex	var	ctrs	deg	verifyrealrootpm				verifyrealrootpc			
				pert.	*time*	*sol*	width	*pert.*	*time*	*sol*	*width*
Ex.1	3	2	4	0	18.9	$1_\triangle$	2.5e-16	0	18.7	1	1.4e-15
				1e-15	1.48	1	5.6e-17	1e-14	8.84	$1, 1_\triangle$	1.6e-15
Ex.2	3	3	2	0	3.33	1*	4.2e-27	0	7.12	3*	2.2e-11
				1e-15	0.99	1	8.5e-21	1e-15	0.48	1	9.9e-31
Ex.3	3	3	2	0	3.90	1*	4.4e-9	0	3.13	1	3.3e-16
				1e-10	22.2	1	8.9e-15	1e-14	2.63	1	8.5e-9
Ex.4	2	2	5	0	3.51	1*	2.3e-19	0	22.9	2*, $1^*_\triangle$	5.6e-13
				1e-20	8.01	1	2.0e-31	1e-15	0.655	1	1.9e-11
Ex.5	3	2	5	0	8.92	$1_\triangle$	4.2e-17	0	32.3	3	2.0e-15
				1e-14	11.5	$1_\triangle$	5.7e-18	1e-15	6.98	5	2.2e-15
Ex.6	2	2	8	0	92.9	$1_\triangle$	6.6e-16	0	15.2	3	2.6e-15
				1e-9	43.6	1	3.6e-12	1e-8	11.3	5	2.9e-12

Table 4: Algorithm Performance on Nonradical Positive-dimensional Polynomial Systems

5. REFERENCES

[1] Philippe Aubry, Fabrice Rouillier, and Mohab Safey El Din. Real solving for positive dimensional systems. *J. Symb. Comput.*, 34(6):543–560, 2002.

[2] Bernd Bank, Marc Giusti, Joos Heintz, Lutz Lehmann, and Luis Miguel Pardo. Algorithms of intrinsic complexity for point searching in compact real singular hypersurfaces. *Foundations of Computational Mathematics*, 12:75–122, 2012.

[3] Bernd Bank, Marc Giusti, Joos Heintz, and Guy Mbakop. Polar varieties and efficient real elimination. *MATHEMATISCHE ZEITSCHRIFT*, 238:115–144, 2000.

[4] Bernd Bank, Marc Giusti, Joos Heintz, Mohab Safey El Din, and Eric Schost. On the geometry of polar varieties. *Applicable Algebra in Engineering, Communication and Computing*, 21:33–83, 2010.

[5] Saugata Basu, Richard Pollack, and Marie-Françoise Roy. On the combinatorial and algebraic complexity of quantifier elimination. *J. ACM*, 43(6):1002–1045, November 1996.

[6] Carlos Beltrán and Anton Leykin. Certified Numerical Homotopy Tracking. *Experimental Mathematics*, 21:69–83, 2012.

[7] Mork Heidi Camilla and Piene Ragni. Polars of real singular plane curves. In *Alorithm in Algebraic Geometry*, pages 99–115. Springer, 2008.

[8] J.F. Canny. Computing roadmaps of general semi-algebraic sets. In HaroldF. Mattson, Teo Mora, and T.R.N. Rao, editors, *Applied Algebra, Algebraic Algorithms and Error-Correcting Codes*, volume 539 of *Lecture Notes in Computer Science*, pages 94–107. Springer Berlin Heidelberg, 1991.

[9] Xiaojun Chen, Andreas Frommer, and Bruno Lang. Computational existence proofs for spherical t-designs. *Numerische Mathematik*, 117:289–305, 2011.

[10] Xiaojun Chen and Robert S. Womersley. Existence of solutions to systems of underdetermined equations and spherical designs. *SIAM Journal on Numerical Analysis*, 44(6):2326–2341, 2006.

[11] G. Chesi, A. Garulli, A. Tesi, and A. Vicino. Characterizing the solution set of polynomial systems in terms of homogeneous forms: an LMI approach. *International Journal of Robust and Nonlinear Control*, 13(13):1239–1257, 2003.

[12] George Collins. Quantifier elimination for real closed fields by cylindrical algebraic decompostion. volume 33 of *Lecture Notes in Computer Science*, pages 134–183. 1975.

[13] Hazel Everett, Sylvain Lazard, Daniel Lazard, and Mohab Safey El Din. The voronoi diagram of three lines. In *Proceedings of the twenty-third annual symposium on Computational geometry*, SCG '07, pages 255–264. ACM, 2007.

[14] Aurélien Greuet and Mohab Safey El Din. Deciding reachability of the infimum of a multivariate polynomial. In Anton Leykin, editor, *Proceedings of the 36th international symposium on Symbolic and algebraic computation*, ISSAC '11, pages 131–138, New York, NY, USA, 2011. ACM.

[15] D. Yu. Grigor'ev and N. Vorobjov. Solving systems of polynomial inequalities in subexponential time. *Journal of Symbolic Computation*, 5(1C2):37 – 64, 1988.

[16] D.Yu. Grigor'ev and N. Vorobjov. Counting connected components of a semialgebraic set in subexponential time. *computational complexity*, 2:133–186, 1992.

[17] Jonathan. D. Hauenstein. Numerically computing real points on algebraic sets. *ArXiv e-prints*, May 2011.

[18] Jonathan D. Hauenstein and Frank Sottile. Algorithm 921: alphacertified: Certifying solutions to polynomial systems. *ACM Trans. Math. Softw.*, 38(4):28:1–28:20, August 2012.

[19] Joos Heintz, Marie-Francoise Roy, and Pablo Solern. Description of the connected components of a semialgebraic set in single exponential time. *Discrete & Computational Geometry*, 11:121–140, 1994.

[20] D. Henrion and J.B. Lasserre. Detecting global optimality and extracting solutions in GloptiPoly. In *Positive polynomials in control*, volume 312 of *Lecture Notes in Control and Inform. Sci.*, pages 293–310. Springer, Berlin, 2005.

[21] R. Krawczyk. Newton-algorithmen zur bestimmung von nullstellen mit fehlerschranken. *Computing*, pages 187–201, 1969.

[22] J.B. Lasserre. *Moments, Positive Polynomials and Their Applications*. Imperial College Press, 2009.

[23] J.B. Lasserre, M. Laurent, and P. Rostalski. Semidefinite characterization and computation of zero-dimensional real radical ideals. *Foundations of Computational Mathematics*, 8:607–647, 2008.

[24] Tsung-Lin Lee, Tien-Yien Li, and Chih-Hsiung Tsai. HOM4PS-2.0: a software package for solving polynomial systems by the polyhedral homotopy continuation method. *Computing*, 83(2-3):109–133, 2008.

[25] A. Leykin, J. Verschelde, and A. Zhao. Newton's method with deflation for isolated singularities of polynomial systems. *Theoretical Computer Science*, 359(1):111–122, 2006.

[26] Nan Li and Lihong Zhi. Verified error bounds for isolated singular solutions of polynomial systems. 2012. Preprint, arxiv.org/pdf/1201.3443.

[27] Nan Li and Lihong Zhi. Verified error bounds for isolated singular solutions of polynomial systems: case of breadth one. *Theoretical Computer Science*, 479:163–173, 2013.

[28] Yue Ma and Lihong Zhi. Computing real solutions of polynomial systems via low-rank moment matrix completion. In J. van der Hoeven and M. van Hoeij, editors, *Proceedings of the 37th international symposium on Symbolic and algebraic computation*, pages 249–256, New York, NY, USA, 2012. ACM.

[29] Angelos Mantzaflaris and Bernard Mourrain. Deflation and certified isolation of singular zeros of polynomial systems. In Anton Leykin, editor, *Proceedings of the 36th international symposium on Symbolic and algebraic computation*, ISSAC '11, pages 249–256, New York, NY, USA, 2011. ACM.

[30] R. E. Moore. A test for existence of solutions to nonlinear systems. *SIAM Journal on Numerical Analysis*, 14(4):pp. 611–615, 1977.

[31] James Renegar. On the computational complexity and geometry of the first-order theory of the reals. *Journal of Symbolic Computation*, 13(3):255 – 299, 1992.

[32] Fabrice Rouillier. Efficient algorithms based on critical points method. In Saugata Basu and Laureano Gonzlez-Vega, editors, *Algorithmic and Quantitative Aspects of Real Algebraic Geometry in Mathematics and Computer Science*, pages 123–138. American Mathematical Society, 2001.

[33] Fabrice Rouillier, Marie-Françoise Roy, and Mohab Safey El Din. Finding at least one point in each connected component of a real algebraic set defined by a single equation. *J. Complexity*, 16(4):716–750, 2000.

[34] S.M. Rump. Solving algebraic problems with high accuracy. In *Proc. of the symposium on A new approach to scientific computation*, pages 51–120, San Diego, CA, USA, 1983.

[35] S.M. Rump. INTLAB - INTerval LABoratory. In Tibor Csendes, editor, *Developments in Reliable Computing*, pages 77–104. Kluwer Academic Publishers, Dordrecht, 1999.

[36] S.M. Rump and S. Graillat. Verified error bounds for multiple roots of systems of nonlinear equations. *Numerical Algorithms*, 54(3):359–377, 2009.

[37] Mohab Safey El Din and Éric Schost. Polar varieties and computation of one point in each connected component of a smooth real algebraic set. In J. R. Sendra, editor, *Proceedings of the 2003 international symposium on Symbolic and algebraic computation*, ISSAC '03, pages 224–231, New York, NY, USA, 2003. ACM.

[38] A. Seidenberg. A new decision method for elementary algebra. *Annals Math.*, 60:365–374, 1954.

[39] F. Shen, W. Wu, and B. Xia. Real Root Isolation of Polynomial Equations Based on Hybrid Computation. *ArXiv e-prints*, July 2012.

[40] Andrew J. Sommese and Jan Verschelde. Numerical homotopies to compute generic points on positive dimensional algebraic sets. *J. Complex.*, 16(3):572–602, September 2000.

[41] Andrew J. Sommese and Charles W. Wampler. Numerical algebraic geometry. In S. Smale J. Renegar, M. Shub, editor, *The mathematics of numerical analysis (Park City, UT, 1995)*, pages 749–763. Lectures in Appl. Math.,32, Amer. Math. Soc., Providence, RI, 1996.

[42] Andrew J. Sommese and Charles W. Wampler. *The numerical solution of systems of polynomials - arising in engineering and science*. World Scientific, 2005.

[43] Silke J. Spang. *On the computation of the real radical*. Thesis, Technische Universität Kaiserslautern, 2007.

[44] H.J. Stetter. *Numerical Polynomial Algebra*. SIAM, 2004.

[45] Jan Verschelde. Algorithm 795: Phcpack: a general-purpose solver for polynomial systems by homotopy continuation. *ACM Trans. Math. Softw.*, 25(2):251–276, 1999.