A Complete Algorithm for Counting Real Solutions of Polynomial Systems of Equations and Inequalities*

Bican Xia[1]
School of Mathematical Sciences
Peking University, Beijing 100871, China

Xiaorong Hou
Chengdu Institute of Computer Applications
Academia Sinica, Chengdu 610041, China

Abstract We present a complete and practical algorithm which can determine the number of distinct real solutions of a given polynomial system of equations and inequalities with integer coefficients mechanically. Based on this algorithm, a program called `nearsolve` has been implemented in Maple. The algorithm and program have been successfully applied to many problems with various backgrounds and to automated discovering and proving for inequality-type theorems.

Keywords real root-counting, polynomial system, resultant, computer algebra, algorithm

1 Introduction

Solving a system of polynomial equations and inequalities is a very important problem, which can find its applications in various fields. In some examples of interest in practice and theory, the major issue is to determine whether a given system has real solutions or not, and to count them if the number is finite. In fact, the problem of counting real solutions of a polynomial system has been extensively studied for many years [4, 7].

*This work is supported in part by the National "973" Project (G1998030602) of China and in part by "95" Key Project on Fundamental Research of Academia Sinica.

[1]who is to receive correspondence.

Suppose we are given a system

$$PSC: \begin{cases} p_1(x_1, x_2, ..., x_s) = 0, \\ p_2(x_1, x_2, ..., x_s) = 0, \\ \cdots\cdots, \\ p_s(x_1, x_2, ..., x_s) = 0, \\ g_1(x_1, x_2, ..., x_s) \geq 0, ..., g_r(x_1, x_2, ..., x_s) \geq 0, \\ g_{r+1}(x_1, x_2, ..., x_s) > 0, ..., g_t(x_1, x_2, ..., x_s) > 0, \\ h_1(x_1, x_2, ..., x_s) \neq 0, ..., h_m(x_1, x_2, ..., x_s) \neq 0, \end{cases}$$

where $p_i(1 \leq i \leq s)$, $g_j(1 \leq j \leq t)$, $h_k(1 \leq k \leq m)$ are all polynomials with integer coefficients and equations $\{p_1 = 0, p_2 = 0, \cdots, p_s = 0\}$ has a finite number of common solutions. The question is how many distinct real solutions this system has.

If the ideal generated by $p_1, ..., p_s$ is zero dimensional, then it is well known that Wu's method, Gröbner basis method, or subresultant method can be used to transform the system of equations into one or more systems in triangular form (see, for example, [12, 13, 2, 10, 11, 8, 1, 16]). Therefore, in next two sections, we only consider triangular equations and the problem we discuss is to compute the number of distinct real solutions of following system TSC,

$$TSC: \begin{cases} f_1(x_1) = 0, \\ f_2(x_1, x_2) = 0, \\ \cdots\cdots, \\ f_s(x_1, x_2, ..., x_s) = 0, \\ g_1(x_1, x_2, ..., x_s) \geq 0, ..., g_r(x_1, x_2, ..., x_s) \geq 0, \\ g_{r+1}(x_1, x_2, ..., x_s) > 0, ..., g_t(x_1, x_2, ..., x_s) > 0, \\ h_1(x_1, x_2, ..., x_s) \neq 0, ..., h_m(x_1, x_2, ..., x_s) \neq 0, \end{cases}$$

where $f_i(1 \leq i \leq s)$, $g_j(1 \leq j \leq t)$, $h_k(1 \leq k \leq m)$ are all polynomials with integer coefficients and $\{f_1, f_2, \cdots, f_s\}$ is a normal ascending chain [17] (also see Section 2 for the definition).

In Section 2, we present a practical algorithm which, dealing with regular TSC (see Definition 2.5), is the kernel part of our method for solving this problem. Section 3 is devoted to some special techniques to handle irregular TSC. Section 4 includes some examples solved by a software named `nearsolve` which implements our algorithms under Maple.

2 General Algorithm

First of all, let us introduce some notations and concepts. In this section, all the polynomials are in $Z[x_1, ..., x_s]$. For any polynomial P with positive degree, the *leading variable* x_l of P is the one with greatest index l that effectively appears in P. By a *triangular set*, we mean a set of polynomials $\{f_i(x_1, ...x_i), f_{i+1}(x_1, ...x_{i+1}), ..., f_l(x_1, ...x_l)\}$ in which the leading variable of f_i is x_i.

Definition 2.1 (Discriminant)

Given a polynomial $g(x)$, let resultant(g, g'_x, x) be the Sylvester resultant of g and g'_x with respect to (w.r.t.) x. We call it the *discriminant* of g w.r.t. x and denote it by Discrim(g, x) or simply by Discrim(g) if its meaning is clear.

It should be pointed out that the definition of discriminant here is slightly different from others which are the quotient of resultant(g, g'_x, x) by the leading coefficient of $g(x)$.

Definition 2.2 (Resultant and Pseudo-remainder w.r.t. a Triangular Set)

Given a polynomial g and a triangular set $\{f_1, f_2, ..., f_s\}$, let

$$r_s := g, \quad r_{s-i} := \text{resultant}(r_{s-i+1}, f_{s-i+1}, x_{s-i+1}), \quad i = 1, 2, ..., s;$$

$$q_s := g, \quad q_{s-i} := \text{prem}(q_{s-i+1}, f_{s-i+1}, x_{s-i+1}), \quad i = 1, 2, ..., s,$$

where resultant(p, q, x) means the Sylvester resultant of p, q w.r.t. x and prem(p, q, x) means the pseudo remainder of p divided by q w.r.t. x.

Let res$(g, f_s, ..., f_i)$ and prem$(g, f_s, ..., f_i)$ denote r_{i-1} and q_{i-1} $(1 \le i \le s)$, respectively, and call them the *resultant* and *pseudo-remainder* of g w.r.t. the triangular set $\{f_i, f_{i+1}, ..., f_s\}$, respectively.

Definition 2.3[16, 17] (Normal Ascending Chain)

Given a triangular set $\{f_1, f_2, ..., f_s\}$, by I_i $(i = 1, 2, ..., s)$ denote the leading coefficient of f_i in x_i. A triangular set $\{f_1, f_2, ..., f_s\}$ is called a *normal ascending chain* if

$$I_1 \ne 0, \quad \text{res}(I_i, f_{i-1}, ..., f_1) \ne 0, \quad i = 2, ..., s.$$

Definition 2.4 (Critical Polynomial of System TSC)

Given system TSC. For every $f_i (i \ge 2)$, let

$$\begin{aligned} Bf_2 &= \text{Discrim}(f_2, x_2), \\ Bf_i &= \text{res}\,(\text{Discrim}(f_i, x_i), f_{i-1}, f_{i-2}, \cdots, f_2), i > 2. \end{aligned}$$

For $\forall q \in \{g_j (1 \le j \le t)\} \bigcup \{h_k (1 \le k \le m)\}$, let

$$Bq = \text{res}\,(q, f_s, f_{s-1}, \cdots, f_2).$$

We define

$$BP = \prod_{2 \le i \le s} Bf_i \cdot \prod_{1 \le j \le t} Bg_j \cdot \prod_{1 \le j \le m} Bh_k.$$

Clearly, BP is a polynomial in x_1 and is called the *critical polynomial* of system TSC w.r.t. x_1.

Definition 2.5 (Regular TSC)

System TSC is *regular* if resultant$(BP(x_1), f_1(x_1), x_1) \ne 0$.

Definition 2.6 (Near Roots)

Given system TSC. Suppose all the distinct real roots of $f_1(x_1)$ are $\alpha_1, \cdots, \alpha_n$. If a list of real numbers, $[r_1, \cdots, r_n]$, satisfies that for $\forall i (1 \le i \le n)$,

1) if $s > 1$, system

$$\begin{cases} f_2(\alpha_i, x_2) = 0, \\ \cdots\cdots, \\ f_s(\alpha_i, x_2, ..., x_s) = 0, \\ g_1(\alpha_i, x_2, ..., x_s) \geq 0, ..., g_r(\alpha_i, x_2, ..., x_s) \geq 0, \\ g_{r+1}(\alpha_i, x_2, ..., x_s) > 0, ..., g_t(\alpha_i, x_2, ..., x_s) > 0, \\ h_1(\alpha_i, x_2, ..., x_s) \neq 0, ..., h_m(\alpha_i, x_2, ..., x_s) \neq 0, \end{cases}$$

and system

$$\begin{cases} f_2(r_i, x_2) = 0, \\ \cdots\cdots, \\ f_s(r_i, x_2, ..., x_s) = 0, \\ g_1(r_i, x_2, ..., x_s) \geq 0, ..., g_r(r_i, x_2, ..., x_s) \geq 0, \\ g_{r+1}(r_i, x_2, ..., x_s) > 0, ..., g_t(r_i, x_2, ..., x_s) > 0, \\ h_1(r_i, x_2, ..., x_s) \neq 0, ..., h_m(r_i, x_2, ..., x_s) \neq 0, \end{cases}$$

have the same number of distinct real solutions and,

2) if $s = 1$, for $\forall q \in \{g_j(1 \leq j \leq t)\} \bigcup \{h_k(1 \leq k \leq m)\}$, $\text{sign}(q(\alpha_i)) = \text{sign}(q(r_i))$, where

$$\text{sign}(x) = \begin{cases} 1, & x > 0, \\ 0, & x = 0, \\ -1, & x < 0. \end{cases}$$

then $[r_1, \cdots, r_n]$ is called a list of *near roots* of $f_1(x_1)$ w.r.t. system TSC. If $f_1(x_1)$ has no real roots, the list of *near roots* is defined to be the empty list $[\,]$.

In the rest of this section, we consider regular systems of the form TSC only. So, no $Bh_k(1 \leq k \leq m)$ has common roots with $f_1(x_1)$, which means every solution of $\{f_1 = 0, f_2 = 0, \cdots, f_s = 0\}$ satisfies $h_k \neq 0$ $(1 \leq k \leq m)$. Thus, we need only to consider regular system TSC without h_k.

Algorithm: `nearroots`

input: two polynomials $p(x)$ and $q(x)$ which have no common roots

output: a list of near roots of $p(x)$ w.r.t. system $\{p(x) = 0, q(x) > 0\}$

Step 1 If $p(x)$ has no real roots, output an empty list $[\,]$. If $q(x)$ has no real roots and $p(x)$ has u distinct real roots, output a list $[0, \cdots, 0]$ in which the number of 0 is u. Otherwise

Step 2 Suppose all the distinct real roots of $q(x)$ are $\alpha_1, \cdots, \alpha_n$. By modified Uspensky algorithm [3] or existing tools in computer algebra system (e.g. function *realroot* in Maple), we can get a sequence of intervals, $[a_1, b_1], \cdots, [a_n, b_n]$, which satisfies,

1) $\alpha_i \in [a_i, b_i]$ for $i = 1, \cdots, n$,

2) $[a_i, b_i] \bigcap [a_j, b_j] = \emptyset$ for $i \neq j$,

3) no roots of $p(x)$ are in any $[a_i, b_i]$,

4) $a_i, b_i (1 \leq i \leq n)$ are all rational numbers.

Step 3 Now we've got a sequence of intervals

$$(-\infty, a_1), \cdots, (b_i, a_{i+1}), \cdots, (b_n, \infty).$$

For every interval in this sequence, determine the number of distinct real roots in it of $p(x)$. If in $(-\infty, a_1)$, $p(x)$ has $u_0 (> 0)$ distinct roots, choose u_0 rational numbers from $(-\infty, a_1)$, say $a_1 - 1$ (u_0 times). If in (b_i, a_{i+1}), $p(x)$ has $u_i (> 0)$ distinct roots, choose u_i rational numbers $(b_i + a_{i+1})/2$. If in (b_n, ∞), $p(x)$ has $u_n (> 0)$ distinct roots, choose u_n rational numbers $b_n + 1$. Finally, sort and output all the numbers we have chosen.

Theorem 2.1 *Given a regular system TSC. Suppose all the distinct real roots of $f_1(x_1)$ are $\alpha_1 < \cdots < \alpha_n$, then, by calling* `nearroots`$(f_1, BP(x_1))$ *we can get a list of rational numbers, $[r_1, \cdots, r_n]$, which is a list of near roots of $f_1(x_1)$ w.r.t. system TSC.*

Proof. If $s = 1$, the conclusion is obvious. So, suppose $s > 1$. Because system TSC is regular, f_1 has no common roots with the critical polynomial $BP(x_1)$. So, by calling `nearroots`$(f_1, BP(x_1))$ we get a list of rational numbers $[r_1, \cdots, r_n]$. We will demonstrate that this is indeed a list of near roots of $f_1(x_1)$ w.r.t. system TSC.

For $\forall i_0 (1 \leq i_0 \leq n)$, we know from Step 3 in algorithm `nearroots` that α_{i_0} and r_{i_0} both lie between some two consecutive roots (say, β_{i_0} and β_{i_0+1}) of $BP(x_1) = \prod_{2 \leq i \leq s} Bf_i \cdot \prod_{1 \leq j \leq t} Bg_j$. Clearly, the sign of each Bf_i and Bg_j is invariant on the interval $(\beta_{i_0}, \beta_{i_0+1})$.

First of all, $Bf_2 = \text{Discrim}(f_2, x_2) \neq 0$ on the interval $(\beta_{i_0}, \beta_{i_0+1})$ implies the number of distinct real roots of f_2 is invariant on $(\beta_{i_0}, \beta_{i_0+1})$. Furthermore, $Bf_3 \neq 0$ on $(\beta_{i_0}, \beta_{i_0+1})$ implies, if $f_2 = 0$, $\text{Discrim}(f_3, x_3) \neq 0$ on $(\beta_{i_0}, \beta_{i_0+1})$, which means the number of distinct real solutions of equations $\{f_2 = 0, f_3 = 0\}$ is invariant on $(\beta_{i_0}, \beta_{i_0+1})$. Continuing similar discussions, we get that the number of distinct real solutions of equations $\{f_2 = 0, \cdots, f_s = 0\}$ is invariant on $(\beta_{i_0}, \beta_{i_0+1})$. Secondly, $Bg_j \neq 0$ on $(\beta_{i_0}, \beta_{i_0+1})$ implies, if $\{f_2 = 0, \cdots, f_s = 0\}$, $g_j \neq 0$ on $(\beta_{i_0}, \beta_{i_0+1})$, which means the number of distinct real solutions of system TSC without f_1 is invariant on $(\beta_{i_0}, \beta_{i_0+1})$. The proof is complete.

Algorithm: `nearsolve`

input: regular system TSC

output: the number of distinct real solutions of system TSC

Step 1 $i := 1$. By `nearroots` we can get a list of near roots of $f_1(x_1)$, say rs_1. If $rs_1 = [r_{11}, \cdots, r_{k_1 1}]$, let $pts_1 := [[r_{11}], \cdots, [r_{k_1 1}]]$. Otherwise, if $rs_1 = [\ ]$, let $pts_1 := [\ ]$;

Step 2 If $pts_i = [\ ]$, let $pts := [\ ]$ and go to Step 4. Otherwise, if

$$pts_i = [[r_{11}, \cdots, r_{1i}], \cdots, [r_{k_i 1}, \cdots, r_{k_i i}]],$$

J1) $j := 1$.

J2) Substitute $x_1 = r_{j1}, \cdots, x_i = r_{ji}$ into system TSC in which $f_1, \cdots, f_i$ has been deleted. For this new system, by `nearroots` we get a list of near roots of $f_{i+1}(x_{i+1})$, say rs_{i+1}^j. If $rs_{i+1}^j = [\ ]$,let $pts_{i+1}^j := [\ ]$. Otherwise, if $rs_{i+1}^j = [r_1, \cdots, r_{l_j}]$, let

$$pts_{i+1}^j := [[r_{j1}, \cdots, r_{ji}, r_1], \cdots, [r_{j1}, \cdots, r_{ji}, r_{l_j}]],\ j := j+1,$$

and goto J3);

J3) If $j \le k_i$, goto J2). Otherwise, let

$$pts_{i+1} := \bigcup_{1 \le j \le k_i} pts_{i+1}^j,\ i := i+1,$$

where $\bigcup_{1 \le j \le k_i} pts_{i+1}^j$ means a list which consists of all the members in each pts_{i+1}^j and then, goto Step 3;

Step 3 If $i = s$, let $pts := pts_i$ and goto Step 4. Otherwise, goto Step 2;

Step 4 If $pts = [\ ]$, output 0. Otherwise, suppose

$$pts = [v_1, \cdots, v_n] = [[r_{11}, \cdots, r_{1s}], \cdots, [r_{n1}, \cdots, r_{ns}]],$$

if u members of pts make every $g_j > 0$ true, output u.

Proof (of the correctness of algorithm `nearsolve`).

If equations $\{f_1 = 0, \cdots, f_s = 0\}$ has no real solutions, pts in Step 4 is obviously an empty list $[\ ]$. If equations $\{f_1 = 0, \cdots, f_s = 0\}$ have n distinct real solutions, $[w_1, \cdots, w_n]$, then by Theorem 2.1, pts in Step 4 must be

$$[v_1, \cdots, v_n] = [[r_{11}, \cdots, r_{1s}], \cdots, [r_{n1}, \cdots, r_{ns}]],$$

in which

$$\forall i(1 \le i \le n)\ \forall j(1 \le j \le t)\ \mathrm{sign}(g_j(w_i)) = \mathrm{sign}(g_j(v_i)).$$

That ends the proof.

3 Special Cases

In this section, we deal with irregular system TSC, i.e. resultant$(BP, f_1, x_1) = 0$. Our main tool in this section is WR algorithm [16, 17]. Here are some related definitions and results.

Definition 3.1[16, 17] (Simplicial)

A normal ascending chain $\{f_1, f_2, ..., f_s\}$ is *simplicial* with respect to a polynomial g if either prem$(g, f_s, ..., f_1) = 0$ or res$(g, f_s, ..., f_1) \neq 0$.

Theorem 3.1 [16, 17] *For a triangular set* $AS: \ \{f_1, f_2, ..., f_s\}$ *and a polynomial* g*, there is a constructive algorithm which can decompose* AS *into some normal ascending chains* $AS_i: \ \{f_{i1}, f_{i2}, ..., f_{is}\}\,(1 \leq i \leq n)$*, in which every chain is simplicial w.r.t.* g *and this decomposition satisfies that* $Zero(AS) = \bigcup_{1 \leq i \leq n} Zero(AS_i)$.

This decomposition is called the WR decomposition of AS w.r.t. g and the algorithm is called the WR algorithm. By Theorem 3.1, the condition that the triangular set $\{f_1, f_2, ..., f_s\}$ in system TSC is a normal ascending chain is always accessible.

Definition 3.2[14] (Discrimination Matrix)

Given a polynomial with general symbolic coefficients,

$$f(x) = a_0 x^n + a_1 x^{n-1} + \cdots + a_n,$$

the following $2n \times 2n$ matrix in terms of the coefficients,

$$\begin{bmatrix}
a_0 & a_1 & a_2 & \cdots & a_n & & & & \\
0 & na_0 & (n-1)a_1 & \cdots & a_{n-1} & & & & \\
 & a_0 & a_1 & \cdots & a_{n-1} & a_n & & & \\
 & 0 & na_0 & \cdots & 2a_{n-2} & a_{n-1} & & & \\
 & & & \cdots & \cdots & & & & \\
 & & & \cdots & \cdots & & & & \\
 & & & a_0 & a_1 & a_2 & \cdots & a_n & \\
 & & & 0 & na_0 & (n-1)a_1 & \cdots & a_{n-1} &
\end{bmatrix}$$

is called the *discrimination matrix* of $f(x)$, and denoted by Discr(f). By d_k or $d_k(f)$ denote the determinant of the submatrix of Discr(f), formed by the first k rows and the first k columns for $k = 1, 2, \cdots, 2n$.

Definition 3.3[14] (Discriminant Sequence)

Let $D_k = d_{2k}, k = 1, \cdots, n$. We call the n-tuple

$$[D_1, \ D_2, \cdots, D_n]$$

the *discriminant sequence* of $f(x)$. Obviously, the last term D_n is Discrim(f, x).

Definition 3.4[17, 9] (Principal Subresultants)

Let D_k^t be the submatrix of Discr(f), formed by the first $2n - 2k$ rows, the first $2n - 2k - 1$ columns and the $(2n - 2k + t)$th column, where $0 \leq k \leq n - 1, \ 0 \leq t \leq 2k$.

Let $|D_k^t| = \det(D_k^t)$. We call $|D_k^0|$ $(0 \le k \le n-1)$ the kth *principal subresultant* of $f(x)$. Obviously, $|D_k^0| = D_{n-k}$ $(0 \le k \le n-1)$.

Definition 3.5[17, 9] (Subresultant Polynomial Chain)
For $k = 0, 1, ..., n-1$, let

$$Q_{n+1}(f, x) = f(x),\ Q_n(f, x) = f'(x),$$

$$Q_k(f, x) = \sum_{t=0}^{k} |D_k^t| x^{k-t} = |D_k^0| x^k + |D_k^1| x^{k-1} + \cdots + |D_k^k|.$$

We call $\{Q_0(f, x),\ Q_1(f, x), ..., Q_{n+1}(f, x)\}$ the *subresultant polynomial chain* of $f(x)$.

Theorem 3.2 [17] *Suppose $\{f_1, f_2, ..., f_j\}$ is a normal ascending chain, where K is a field and $f_i \in K[x_1, ..., x_i]$, $i = 1, 2, ..., j$ and $f(y) = a_0 y^n + a_1 y^{n-1} + \cdots + a_{n-1} y + a_n$ is a polynomial in $K[x_1, ..., x_i][y]$, let*

$$PD_k = \mathrm{prem}(|D_k^0|, f_j, ..., f_1) = \mathrm{prem}(D_{n-k}, f_j, ..., f_1),\ 0 \le k \le n-1.$$

If for some $k_0 \ge 0$,

$$\mathrm{res}(a_0, f_j, ..., f_1) \ne 0$$

$$PD_0 = \cdots = PD_{k_0-1} = 0,\quad \mathrm{res}(|D_{k_0}^0|, f_j, ..., f_1) \ne 0,$$

then, in $K[x_1, ..., x_j]/(f_1, ..., f_j)$, we have $\gcd(f, f'_x) = Q_{k_0}(f, x)$.

Now, we deal with irregular system TSC with resultant$(BP, f_1, x_1) = 0$. The idea is to decompose this system by WR algorithm into some new regular systems and thus, to which the general algorithm `nearsolve` in Section 2 is applicable.

- If there is some Bh_k so that resultant$(BP, Bh_k, x_1) = 0$, do WR decomposition of $\{f_1, f_2, ..., f_s\}$ w.r.t. h_k and, without loss of generality, suppose we get two new chains $\{A_1, A_2, ..., A_s\}$ and $\{C_1, C_2, ..., C_s\}$, in which $\mathrm{prem}(g_j, A_s, ..., A_1) = 0$ but $\mathrm{res}(g_j, C_s ..., C_1) \ne 0$. If we replace $\{f_1, f_2, ..., f_s\}$ by $\{C_1, C_2, ..., C_s\}$ in system TSC, the new system is regular. Obviously, another system obtained by replacing $\{f_1, f_2, ..., f_s\}$ with $\{A_1, A_2, ..., A_s\}$ in system TSC, has no real solution.

- If there is some Bg_j so that resultant$(BP, Bg_j, x_1) = 0$, do WR decomposition of $\{f_1, f_2, ..., f_s\}$ w.r.t. g_j and, without loss of generality, suppose we get two new chains $\{A_1, A_2, ..., A_s\}$ and $\{C_1, C_2, ..., C_s\}$, in which $\mathrm{prem}(g_j, A_s, ..., A_1) = 0$ but $\mathrm{res}(g_j, C_s ..., C_1) \ne 0$. Now, if in system TSC we have $g_j > 0$, we simply replace $\{f_1, f_2, ..., f_s\}$ by $\{C_1, C_2, ..., C_s\}$ and the new system is regular. If in system TSC we have $g_j \ge 0$, we first get a system $TSC1$ by replacing $\{f_1, f_2, ..., f_s\}$ with $\{C_1, C_2, ..., C_s\}$ and then, get another system $TSC2$ by replacing $\{f_1, f_2, ..., f_s\}$ with $\{A_1, A_2, ..., A_s\}$ and deleting g_j from it. These two systems are both regular.

- If there is some Bf_i so that resultant$(BP, Bf_i, x_1) = 0$, let $[D_1, \cdots, D_{n_i}]$ be the discriminant sequence of f_i w.r.t. x_i. First of all, we do WR decomposition of $\{f_1, ..., f_{i-1}\}$ w.r.t. D_{n_i} and, without loss of generality, suppose we get two new chains $\{A_1, ..., A_{i-1}\}$ and $\{C_1, ..., C_{i-1}\}$, in which $\mathrm{prem}(D_{n_i}, A_{i-1}, ..., A_1) = 0$ but $\mathrm{res}(D_{n_i}, C_{i-1} ..., C_1) \neq 0$. Step 1, replacing $\{f_1, ..., f_{i-1}\}$ with $\{C_1, ..., C_{i-1}\}$, we will get a regular system. Step 2, let us consider the system obtained by replacing $\{f_1, ..., f_{i-1}\}$ with $\{A_1, ..., A_{i-1}\}$ which is still irregular. Consider D_{n_i-1}, the next term in $[D_1, \cdots, D_{n_i}]$. If $\mathrm{res}(D_{n_i-1}, A_{i-1}, ..., A_1) = 0$, do WR decomposition of $\{A_1, ..., A_{i-1}\}$ w.r.t. D_{n_i-1}. Keep repeating the same procedure [1] until at a certain step we have, for certain D_{i_0} and $\{\bar{A}_1, ..., \bar{A}_{i-1}\}$, $\mathrm{res}(D_{i_0}, \bar{A}_{i-1}, ..., \bar{A}_1) \neq 0$ and $\forall j$ $(i_0 < j \leq n_i)$, $\mathrm{prem}(D_j, \bar{A}_{i-1}, ..., \bar{A}_1) = 0$. By Theorem 3.2, we have $\gcd(f_i, f_i') = Q_{n_i-i_0}(f_i, x_i)$ in $K[x_1, ..., x_{i-1}]/(\bar{A}_1, ..., \bar{A}_{i-1})$. Now, let $\bar{f}_i$ be the pseudo quotient of f_i divided by $\gcd(f_i, f_i')$ and replace $\{f_1, ..., f_{i-1}, f_i\}$ with $\{\bar{A}_1, ..., \bar{A}_{i-1}, \bar{f}_i\}$, the new system will be regular.

- If by repeating above three kinds of processes, we decompose irregular system TSC into some regular system $TSC_i\,(1 \leq i \leq n)$ and we have $Zero(TSC_{i1}) \subseteq Zero(TSC_{i2})$ for some $i1 \neq i2$, $1 \leq i1, i2 \leq n$, then delete system TSC_{i1}.

Thus, we have got

Theorem 3.3 *For an irregular system TSC, there is a constructive algorithm which can decompose TSC into some regular systems TSC_i. Let $NZero(\cdot)$ denote the number of distinct real solutions of a given system, then this decomposition satisfies $NZero(TSC) = \sum NZero(TSC_i)$.*

4 Examples

We have implemented the algorithms described in the above two sections in Maple. As mentioned in Section 1, combining with Wu's method or Gröbner basis method, our algorithms are applicable to system PSC. In our implementation, if the input system is of the form PSC, we use Wu's method to transform the system of equations to one or more systems in triangular form. The calling sequences for system TSC and PSC are respectively

$$\texttt{nearsolve}([f_1, \cdots, f_s], [g_1, \cdots, g_r], [g_{r+1}, \cdots, g_t], [h_1, \cdots, h_m], [x_1, \cdots, x_s]);$$

and

$$\texttt{nearsolve}([p_1, \cdots, p_s], [g_1, \cdots, g_r], [g_{r+1}, \cdots, g_t], [h_1, \cdots, h_m], [x_1, \cdots, x_s]);\ .$$

This program has been applied to automated discovering for inequality-type theorems and has been included in our package DISCOVERER [15].

[1] This procedure must terminate because $\{f_1, ..., f_s\}$ being a normal ascending chain implies $\mathrm{res}(I_i, f_{i-1}, ..., f_1) \neq 0$ and $D_1 = n_i I_i^2$ implies $\mathrm{res}(D_1, f_{i-1}, ..., f_1) \neq 0$.

Example 4.1 This example originated from a special case of so-called P-3-P problem [5, 6, 15]. Given system

$$\begin{cases} p_1(x,y,z,a,b) = x^2 + y^2 - xy - 1 = 0, \\ p_2(x,y,z,a,b) = y^2 + z^2 - yz - a^2 = 0, \\ p_3(x,y,z,a,b) = z^2 + x^2 - zx - b^2 = 0, \\ x > 0, y > 0, z > 0, a - 1 \geq 0, b - a \geq 0, a + 1 - b > 0, \end{cases}$$

the question is how many distinct real solutions this system has if $a = 2$ and $b = 21/10$. Substituting $a = 2, b = 21/10$ into that system, we have

$$\begin{cases} p_1(x,y,z) = x^2 + y^2 - xy - 1 = 0, \\ p_2(x,y,z) = y^2 + z^2 - yz - 4 = 0, \\ p_3(x,y,z) = 100z^2 + 100x^2 - 100zx - 441 = 0, \\ x > 0, y > 0, z > 0. \end{cases}$$

By calling

$$\texttt{nearsolve}([p_1, p_2, p_3], [\,], [x, y, z], [\,], [x, y, z]);$$

we get the answer is 1.

Example 4.2 The number of distinct real solutions of system

$$\begin{cases} f_1(x) = -5000000x^6 + 1875000x^4 - 68125x^2 + 8 = 0, \\ f_2(x,y) = (-1000 - 8000x^3)y^2 + (-8000x^4 + 4000x^2 + 8)y - 117x \\ \qquad -8000x^5 + 3000x^3 = 0, \\ f_3(x,y,z) = (8x + 8y)z + 8xy - 1 = 0, \\ x + y > 0, y + z > 0, z + x > 0. \end{cases}$$

is 5.

Example 4.3 How many distinct real solutions does the following system have?

$$\begin{cases} f_1(b) = 0, \\ f_2(b,c) = 0, \\ f_3(b,c,d) = 0, \\ f_4(b,c,d,e) = 0, \\ b > 0, c > 0, d > 0, e > 0, c - d \neq 0, \end{cases}$$

where

$$\begin{aligned} f_1 = {} & 241538508382138075462768483549507937558926051383237186598921\backslash \\ & 351434775082998337615592652313777086354071766371461311171128509762\backslash \\ & 9761b^{32} + 635066713778840598710749498577504496793070850884097947974\backslash \\ & 380292191777777224247909356698829052300189668676627063462218165 26\backslash \\ & 252732166640b^{30} - 617516729685594231347246877282308919087789340602 36\backslash \\ & 333460795119633799976734997949468942620276039633337231215472829 57\backslash \end{aligned}$$

$$
\begin{aligned}
&288249561157268480000b^{28} + 27390034646753639766624212069599001290967\backslash\\
&5944831268619419963947375736698335046092233994317817055192976 2470\backslash\\
&251477314187497028082105057280b^{26} - 14371451662375545794776393518 90\backslash\\
&5979491561814339246077914862723414402467431025889585529684328 2026\backslash\\
&273568967644536703423955174325414264832 0b^{24} - 298609258728339835915\backslash\\
&1187320928040065994286379338588944475146452773892605985950218 4401\backslash\\
&2436391877650836905308408943702288447254625779712b^{22} + 435447287511\backslash\\
&2985246215589621601392927034444281183521252549255181277184403 3485\backslash\\
&662489132077458388407791801673830767006425164301268418560b^{20} - 3414\backslash\\
&8807436745609347300495600312270857833357366729397393592991014 1878\backslash\\
&35405659193523959392478147852967299724900570030261090683128381 44b^{18} +\\
&8223756555269861165757056665815244377194121394959540192025015 5054\backslash\\
&5049085149044464549229420580826884899873578176474995576252160 5406\backslash\\
&72b^{16} - 1846586391153461422277140725472721854055307790398149406 0726\backslash\\
&3349347328428027423682270256946111377666109617855536427391671 1096\backslash\\
&2872320b^{14} + 210458398154301515264393872128757750183426243499830192\backslash\\
&2594581580196154317969812721378800827355158137115610536502092 5245\backslash\\
&8008412160b^{12} - 139786675574463317676421937828553960047493569539985\backslash\\
&1825067720609793412281015523288310487856480352735638714141311 7228\backslash\\
&188455403 52b^{10} + 585188215302425253433702248414515313183366444530 00\backslash\\
&0749744203303733447689121059491379312443203975837106226735111 6039\backslash\\
&560626176b^{8} - 155390183378463952221186520858978074062380250509 9793\backslash\\
&4777821414922700387593995537411137322766733076998082737318834 9301\backslash\\
&88288b^{6} + 3139960540165071204464736713291845422900077966277745 6747\backslash\\
&4226322417862962960468650427340236503415025338777895317253652 48b^{4} -\\
&2214798152846620823775109546914376369749948855722620121351416 6702\backslash\\
&188991127101416805749416908807763189989750987555 4816b^{2} + 6072087665\backslash\\
&3402761142507664131495320256148247367176990429610550229613067 7639\backslash\\
&952549181438379528451116769583982 1824
\end{aligned}
$$

$$
\begin{aligned}
f_2 = {} & 2075\, b^{16}\, c^{12} + 284580\, b^{14}\, c^{10} + 357840\, b^{12}\, c^{9} + 10185588\, b^{12}\, c^{8} + 20167488\, b^{10}\, c^{7} \\
& + (21285312\, b^{8} - 62355744\, b^{10})\, c^{6} - 99610560\, b^{8}\, c^{5} \\
& + (-4855244976\, b^{8} - 361573632\, b^{6})\, c^{4} + (-37158912\, b^{4} - 3758980608\, b^{6})\, c^{3} \\
& + (54181472832\, b^{6} + 429235200\, b^{4})\, c^{2} + (4897760256\, b^{4} + 488374272\, b^{2})\, c \\
& - 123974556480\, b^{4} + 18874368 + 9432723456\, b^{2}
\end{aligned}
$$

$$f_3 = 9\,b^4\,d^3 + 45\,b^4\,c\,d^2 + (35\,b^4\,c^2 - 486\,b^2)\,d - 108\,b^2\,c - 264 + 10\,b^4\,c^3$$

$$\begin{aligned} f_4 &= (36\,d^2\,b^2 - 8\,c^2\,b^2 - 28\,d\,b^2\,c)\,e + 15\,b^4\,c\,d^2 + 6\,b^4\,c^3 + 21\,b^4\,c^2\,d - 144\,b^2\,c \\ &\quad + 9\,b^4\,d^3 - 120 - 648\,b^2\,d. \end{aligned}$$

We call

$$\texttt{nearsolve}([f_1, f_2, f_3, f_4], [\,], [b, c, d, e], [c - d], [b, c, d, e]);$$

and get the number of distinct real solutions is 6.

Example 4.4 Irregular system

$$\begin{cases} f_1(r) = r^4 + 24r^3 + 194r^2 + 472r - 15 = 0, \\ f_2(r, a) = a^3 - 2a^2 + (r^2 + 8r + 1)a - 8r = 0, \\ f_3(r, a, b) = ab^2 + (a^2 - 2a)b + 8r = 0, \\ f_4(r, a, b, c) = a + b + c - 2 = 0, \\ r > 0, a > 0, b > 0, c > 0, 1 - a > 0, 1 - b > 0, 1 - c > 0. \end{cases}$$

has 3 distinct real solutions.

The time spent for above four examples on a Pentium/266 (64Mb memories) PC with Maple 5.4 are 0.5 seconds, 3 seconds, 1996 seconds and 275 seconds respectively.

Remark As mentioned at the beginning of this section, for an input system of the form PSC, `nearsolve` calls Wu's algorithm to transform it into system(s) of the form TSC at first. So, the timing given for Example 4.1 includes a preliminary application of Wu's algorithm and the time spent on that is 0.1 seconds. If the input is an irregular system TSC, `nearsolve` decomposes it into regular systems by the algorithm described in Section 3 before calling `nearroots`. The time spent on that decomposition for Example 4.4 is nearly 4 seconds.

References

[1] P. Aubry, D. Lazard and M. Moreno Maza, On the theories of triangular sets, *J. Symb. Comput.* **28** 105-124 (1999).

[2] B. Buchberger, Gröbner bases: An algorithmic method in polynomial ideal theory, In *Multidimensional Systems Theory*, (Edited by N.K. Bose), pp. 184-232, D. Reidel, Dordrecht, (1985).

[3] G.E. Collins and R. Loos, Real Zeros of Polynomials, In *Computer Algebra: Symbolic and Algebraic Computation*, (Edited by B. Buchberger *et al.*), pp. 83-94, Springer-Verlag, New York, (1983).

[4] D. Cox, J. Little, and D. O'Shea, *Using Algebraic Geometry*, Springer-Verlag, New York (1998).

[5] E. Folke, Which triangles are plane sections of regular tetrahedra?, *American Mathematics Monthly* Oct. 788-789 (1994).

[6] X.S. Gao and H.F. Cheng, On the Solution Classification of the "P3P" Problem, In *Proc. of ASCM'98*, (Edited by Z. Li), pp. 185-200, Lanzhou University Press, Lanzhou, (1998).

[7] L. Gonzalez-Vega, F. Rouillier, M.-F. Roy and G. Trujillo, Symbolic Recipes for Real Solutions, In *Some Tapas of Computer Algebra*, (Edited by A.M. Cohen *et al.*), pp. 121-167, Springer-Verlag, New York, (1999).

[8] D. Lazard, A new method for solving algebraic systems of positive dimension, *Discrete Appl. Math.* **33** 147-160 (1991).

[9] R. Loos, Generalized Polynomial Remainder Sequences, In *Computer Algebra: Symbolic and Algebraic Computation*, (Edited by B. Buchberger *et al.*), pp. 115-137, Springer-Verlag, New York, (1983).

[10] D.M. Wang, An elimination method for polynomial systems, *J. Symb. Comput.* **16** 83-114 (1993).

[11] D.M. Wang, Decomposing polynomial systems into simple systems, *J. Symb. Comput.* **25** 295-314 (1998).

[12] W.T. Wu, On zeros of algebraic equations–An application of Ritt principle, *Kexue Tongbao* **31** 1-5 (1986).

[13] W.T. Wu, *Mechanical theorem proving in geometries: Basic principles* (translated from Chinese by X. Jin and D. Wang), Springer, New York, (1994).

[14] L. Yang, Recent advances on determining the number of real roots of parametric polynomials, *Journal of Symbolic Computation* **28** 225-242, (1999).

[15] L. Yang, X.R. Hou and B.C. Xia, Automated Discovering and Proving for Geometric Inequalities, In *Automated Deduction in Geometry*, (Edited by X.S. Gao *et al.*), *LNAI 1669*, pp. 30-46, Springer-Verlag, New York, (1999).

[16] L. Yang, J.Z. Zhang and X.R. Hou, An Efficient Decomposition Algorithm for Geometry Theorem Proving Without Factorization, In *Proceedings of Asian Symposium on Computer Mathematics 1995*, (Edited by H. Shi *et al.*), pp.33-41, Scientists Incorporated, Japan, (1995).

[17] L. Yang, J.Z. Zhang and X.R. Hou, *Nonlinear Algebraic Equation System and Automated Theorem Proving* (in Chinese), Shanghai Scientific and Technological Education Publishing House, Shanghai (1996).