Root Isolation for Bivariate Polynomial Systems with Local Generic Position Method

Jin-San Cheng
KLMM, AMSS
Chinese Academy of Sciences
and INRIA-Nancy, France
jcheng@amss.ac.cn

Xiao-Shan Gao
KLMM, AMSS
Chinese Academy of Sciences
xgao@mmrc.iss.ac.cn

Jia Li
KLMM, AMSS
Chinese Academy of Sciences
lijia@mmrc.iss.ac.cn

ABSTRACT

A local generic position method is proposed to isolate the real roots of a bivariate polynomial system

$$\Sigma = \{f(x,y), g(x,y)\}.$$

In this method, the roots of the system are represented as linear combinations of the roots of two univariate polynomial equations $t(x) = 0$ and $T(X) = 0$:

$$\{x = \alpha,\, y = \frac{\beta - \alpha}{s} \mid \alpha \in V(t(x)), \beta \in V(T(X)), |\beta - \alpha| < S\},$$

where s, S are constants satisfying certain conditions. The multiplicities of the roots of $\Sigma = 0$ are the same as that of the corresponding roots of $T(X) = 0$. This representation leads to an efficient and stable algorithm to isolate the real roots of Σ.

Categories and Subject Descriptors

I.1.2 [**SYMBOLIC AND ALGEBRAIC MANIPULATION**]: Algorithms—*Algebraic algorithms*

General Terms

Algorithm

Keywords

Bivariate polynomial system, generic position, root isolation, root bound.

1. INTRODUCTION

Solving polynomial equation systems is a fundamental problem in symbolic computation. In this paper, we consider the problem of real root isolation for bivariate polynomial equation systems. Let $f(x,y), g(x,y) \in \mathbb{Q}[x,y]$, where $\mathbb{Q}$ is the field of rational numbers. We call

$$\Sigma = \{f(x,y), g(x,y)\}. \tag{1}$$

zero-dimensional if $\gcd(f(x,y), g(x,y)) = 1$. Even this simple case has applications in nonlinear computational geometry such as topology determination of curves [2, 7, 8, 11, 13].

In this paper, a **local generic position** method is proposed to isolate the real roots of an equation system like (1). The concept of generic position was used in equation solving and topology determination for a long time [1, 4, 5, 7, 10, 12, 13, 15, 17, 21]. Simply speaking, a system of equations is said to be in a generic position if we can find a direction, say the x-axis, such that different zeros of (1) are projected to different points on the x-axis. If Σ is in a generic position, the roots of an equation system $\Sigma = 0$ have a rational univariate representation (RUR) [17]:

$$t(u) = 0, x = R_1(u), y = R_2(u) \tag{2}$$

where u is a new parameter, $t(u) \in \mathbb{Q}[u]$ and $R_1(u), R_2(u)$ are rational functions. As a consequence, solving multivariate equations is reduced to solving a univariate equation $t(u) = 0$ and to substituting the roots of $t(u) = 0$ into rational functions. This approach still has the following problem: for an isolation interval $[a, b]$ of a real root α of $t(u) = 0$, to determine the isolation interval of $R_1(\alpha)$ and $R_2(\alpha)$ under a given precision is not a trivial task. The local generic position method proposed in this paper will remedy this drawback.

In the local generic position method, the roots of the system Σ are represented as linear combinations of the roots of two univariate polynomial equations $t(x) = 0$ and $T(X) = 0$:

$$\{x = \alpha,\, y = \frac{\beta - \alpha}{s} \mid \alpha \in V(t(x)), \beta \in V(T(X)), |\beta - \alpha| < S\},$$

where s, S are constants satisfying certain given conditions. The multiplicities of the roots of $\Sigma = 0$ are also preserved in the corresponding roots of $T(X) = 0$. The major advantage of this representation is that we can obtain isolation boxes with any given precision for the roots of $\Sigma = 0$ from the isolation intervals of the roots of $t(x) = 0$ and $T(X) = 0$ easily. The methods are implemented in Maple and extensive experiments are done, which show that our approach is very efficient and stable, especially when the system has multiple roots.

Geometrically, the local generic method transforms Σ to a new system Σ' which is in a generic position. Furthermore, the roots of Σ with the same x-coordinate α are transformed to the region $[\alpha - S, \alpha + S] \times [-\infty, -\infty]$. This property allows us to recover these roots from the projections of the roots of Σ and Σ' to the x-axis.

Besides the generic position method, there exist quite a few methods for solving polynomial equation systems such as

ISSAC'09, July 28–31, 2009, Seoul, Republic of Korea.

the Gröbner basis method, the resultant method, the characteristic set method, and the subdivision based method. Here we compare our method with those that are devoted to bivariate equation systems.

In [7], Diochnos, Emiris, and Tsigaridas gave three algorithms to solve bivariate equation systems and analyzed their complexities. Algorithm GRID projects the roots of Σ to the x and y axes and checks whether a combination of the x- and y-coordinates is a solution of Σ. Assuming the equation system in generic position, the algorithm MRUR uses signed subresultant sequences to compute an RUR like (2) and finds the solution of the system by estimating the value of the rational functions $R(x)$ at the x-coordinates of the roots of Σ. This method is quite similar to the method given in [13]. The GRUR method projects the roots of the system to the x and y axes, for each x-coordinate α computes the GCD $H(\alpha, y)$ of the square-free parts of $f(\alpha, y)$ and $g(\alpha, y)$, and isolates the roots of $H(\alpha, y) = 0$ based on computations of algebraic numbers and the RUR techniques. Among the three algorithms, GRUR has the lowest complexity and performs best in experiments. Our algorithm only uses resultant computation and root isolation for univariate polynomial equations with rational coefficients. Our algorithm totally avoids computation over algebraic numbers and is more efficient than GRUR as shown by experimental results in Section 5.

The method by Hong, Shan, and Zeng [14] projects the roots of Σ to the x-axis and y-axis respectively and uses a numerical iteration method to decide whether the boxes formed by the projection intervals contain a root of Σ. The numerical method works for simple roots of Σ only. When the system has multiple roots, the RUR technique is used to isolate them. Comparing to this method, our method also computes two resultants of the same total degrees. Our method is a complete one, while the method given in [14] needs to use the RUR technique to find multiple roots.

The rest of this paper is organized as follows. In Section 2, the theory behind the local generic position method is presented. In Section 3, we estimate the bounds needed in the algorithm. In Section 4, we give the local generic position algorithm. Experimental results are presented in Section 5 and conclusions are given in Section 6.

2. LOCAL GENERIC POSITION

In this section, we present the theory behind the local generic position algorithm. The idea is to do a shear transformation $(x, y) \to (x + sy, y)$ so that the new equation system is in a "local generic position" with respect to the original equation system.

Let π be the projection map from the real plane to the x-axis:

$$\pi : \mathbb{R}^2 \longrightarrow \mathbb{R}, \qquad \text{such that } \pi(x, y) = x. \tag{3}$$

For a zero-dimensional system $\Sigma = \{f(x, y), g(x, y)\}$ defined in (1), let $t(x) \in \mathbb{Q}[x]$ be the resultant of $f(x, y)$ and $g(x, y)$ w.r.t y:

$$t(x) = \mathrm{Res}_y(f(x, y), g(x, y)). \tag{4}$$

Since Σ is zero-dimensional, we have $t(x) \not\equiv 0$. Then $\pi(V(\Sigma)) \subseteq V(t(x))$, where $V(f_1, \ldots, f_r)$ is the set of common real zeros of $f_1 = 0, \ldots, f_r = 0$. Let the real roots of $t(x) = 0$ be

$$\alpha_1 < \alpha_2 < \cdots < \alpha_m. \tag{5}$$

Using the notations in (1) and (5), let S, R, and s be rational numbers satisfying

$$\begin{aligned} S &< \frac{1}{2}\min\{\alpha_{i+1} - \alpha_i, i = 1, \ldots, m-1\}, \\ R &> \max\{|\beta|, \forall(\alpha, \beta) \in V(\Sigma)\}, \\ 0 &< s < \frac{S}{R}. \end{aligned} \tag{6}$$

For s satisfying (6), define an inversive linear map (a shear) from $\mathbb{R}^2$ to $\mathbb{R}^2$:

$$\psi_s : \quad (x, y) \quad \longmapsto \quad (X, Y) = (x + s\,y, y). \tag{7}$$

We also define $\psi_s(f(x, y)) = f(X - s\,Y, Y)$ for convenience. Geometrically, ψ_s maps a point (x_0, y_0) to the intersection point of the lines $y = y_0$ and $(x - x_0) = s\,y$. See Fig. 1 for an illustration.

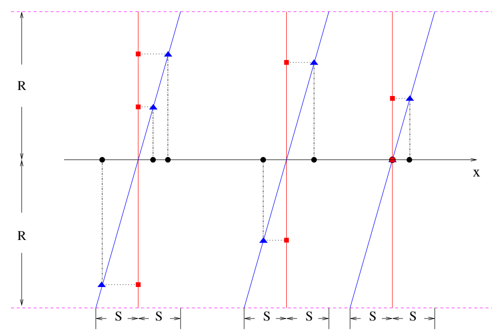

Figure 1: Map ψ_s: the red squares are the roots of $\Sigma = 0$; the blue triangles are the roots of $\psi_s(\Sigma)$; the black dots are the roots of univariate polynomial $T(X) = 0$.

An equation system Σ is said to be in a **generic position** if different roots of $\Sigma = 0$ have different x-coordinates. We have:

Lemma 2.1. *For S and R defined in (6) and ψ_s defined in (7), $\psi_s(\Sigma)$ is in a generic position. Furthermore, a root (α, β) of $\Sigma = 0$ is mapped to (η, β) where $\eta \in (\alpha - S, \alpha + S)$. See Fig. 1 for an illustration.*

Proof. For each α_i in (5), let $P_{i,j} = (\alpha_i, \beta_{i,j})$ be the corresponding roots of $\Sigma = 0$. We have $\psi_s(P_{i,j}) = (\alpha_i + s\beta_{i,j}, \beta_{i,j})$. Then, for the same i, different $\psi_s(P_{i,j})$ have different x-coordinates. Due to the conditions in (6), we have $|\alpha_i + s\beta_{i,j} - \alpha_i| = |s\beta_{i,j}| < (S/R) \cdot R = S$. That is, $\psi_s(P_{i,j}) \in R_i = (\alpha_i - S, \alpha_i + S) \times [-R, R]$. Since $S < \frac{1}{2}(\alpha_{i+1} - \alpha_i)$, R_i are disjoint for different i. Then different $\psi_s(P_{i,j})$ have different x-coordinates. This proves the lemma. ∎

We project the roots of $\psi_s(\Sigma)$ to the x-axis by computing the resultant $T(X)$:

$$\begin{aligned} T(X) &= \mathrm{Res}_y(\psi_s(f(x, y)), \psi_s(g(x, y))) \\ &= \mathrm{Res}_Y(f(X - s\,Y, Y), g(X - s\,Y, Y)). \end{aligned} \tag{8}$$

We hope that the zeros of $\Sigma = 0$ and the roots of $T(X)$ are in a one-to-one correspondence. This may fail when

$$h(X) = \gcd(\mathrm{LC}_Y(\psi_s(f(x, y))), \mathrm{LC}_Y(\psi_s(g(x, y)))) \tag{9}$$

has real roots, where $\mathrm{LC}_Y(f(X, Y))$ is the **leading coefficient** of $f(X, Y)$ w.r.t Y.

So if we ensure that $h(X) = 0$ has no real roots, then the real roots of $\Sigma = 0$ and the real roots of $T(X) = 0$ are in a one-to-one correspondence. We can select the parameter s properly so that $\mathrm{LC}_Y(\psi_s(f(x, y)))$ or $\mathrm{LC}_Y(\psi_s(g(x, y)))$ is a constant and $h(X) = 0$ has no real roots.

Write $f(x, y)$ and $g(x, y)$ as the sum of their homogeneous parts:

$$\begin{aligned} f(x, y) &= f_p(x, y) + \cdots + f_0 \\ g(x, y) &= g_q(x, y) + \cdots + g_0 \end{aligned}$$

where f_i and g_i are homogeneous polynomials with total degree i. It is clear that when

$$f_p(-s, 1) \neq 0 \text{ or } g_q(-s, 1) \neq 0, \tag{10}$$

$h(X)$ is a constant. It is always possible to chose an s such that (6) is satisfied. Then we further have

Lemma 2.2. *Let s be a rational number satisfying (6) and (10). Then π is a one-to-one and multiplicity preserving map between the roots of $\psi_s(\Sigma)$ and the roots of $T(X) = 0$, where $T(X)$ is defined in (8).*

Furthermore, let the roots of $T(X) = 0$ in $(\alpha_i - S, \alpha_i + S)$ be

$$\beta_{i,1} < \beta_{i,2} < \cdots < \beta_{i,m_i}, i = 1, \ldots, m \tag{11}$$

where α_i is defined in (5). Then the inversion of the π is:

$$\pi^{-1}(\beta_{i,j}) = (\beta_{i,j}, (\beta_{i,j} - \alpha_i)/s). \tag{12}$$

Proof. By the property of the resultant, $\pi(V(\psi_s(\Sigma)) \subset V(T(X))$. By Lemma 2.1, we can derive that different roots of $\psi_s(\Sigma)$ are mapped to different roots of $T(X) = 0$. Furthermore, by (10), the leading coefficient of $\psi_s(f)$ or $\psi_s(g)$ does not vanish. Then by the property of the resultant, $\pi(V(\psi_s\Sigma)) = V(T(X))$. Hence π is one-to-one. Based on the theory in Section 1.6 of [9], we can conclude that π is also multiplicity preserving.

$\pi^{-1}(\beta_{i,j})$ can be obtained as follows. By the proof of Lemma 2.1, a root $Q_{i,j} = (\beta_{i,j}, \gamma_{i,j})$ of $\psi_s(\Sigma) = 0$ is projected one-to-one to a root of $T(X) = 0$ in $(\alpha_i - S, \alpha_i + S)$. Then, from the definition of ψ_s, $Q_{i,j}$ is on the line defined by $(x - \alpha_i) = s\,y$ (the skewed lines in Fig. 1). Then, we have $\gamma_{i,j} = (\beta_{i,j} - \alpha_i)/s$. ▮

The following result shows how to recover the roots of $\Sigma = 0$ from the roots of two univariate polynomial equations $t(x) = T(X) = 0$.

Theorem 2.3. *Use the notations introduced in this section. If (6) and (10) are satisfied, then $\theta = \pi \circ \psi_s$ is a one-to-one and multiplicity preserving map from $V(\Sigma)$ to $V(T(X))$. Furthermore, the roots of $\Sigma = 0$ can be obtained by the inversion of θ:*

$$\begin{aligned} \theta^{-1}(\beta_{i,j}) = \quad & (\alpha_i, (\beta_{i,j} - \alpha_i)/s), |\beta_{i,j} - \alpha_i| < S, \\ & i = 1, \ldots, m, j = 1, \ldots, m_i \end{aligned} \tag{13}$$

where $\alpha_i \in V(t(x))$, $\beta_{i,j} \in V(T(X))$ are defined in (5) and (11) respectively, $t(x), T(X)$ are defined by (4) and (8), respectively.

Proof. Since ψ_s is an inverse linear map, it is one-to-one and multiplicity preserving. Then by Lemma 2.2, θ is also one-to-one and multiplicity preserving. The inversion map $\theta^{-1} = \psi_s^{-1} \circ \pi^{-1}$ can be obtained directly from (12) and (7). ▮

As corollaries of Theorem 2.3, we have

Corollary 2.4. *Under the same condition of Theorem 2.3, we have*

$$\begin{aligned} V(\Sigma) = \quad & \{(\alpha, (\beta - \alpha)/s) \,|\, \alpha \in V(t(x)), \beta \in V(T(X) \\ & \quad \text{and } |\alpha - \beta| < S\}. \end{aligned} \tag{14}$$

Due to (14), if $|\alpha - \beta| < S$, we say that β is **associated with** α.

Corollary 2.5. *If we separate the real roots of $t(x) = 0$ and $T(X) = 0$ with precisions ρ_1 and ρ_2 respectively, then the roots computed with (14) have precision $\max\{\rho_1, \frac{\rho_1 + \rho_2}{s}\}$.*

From Theorem 2.3, the four-tuple

$$\{t(x), T(X), s, S\} \tag{15}$$

provides a representation for the roots of $\Sigma = 0$, and from this representation, we can compute the roots of Σ by solving two univariate equations. This method is called a **local generic position method** because the roots of $\Sigma = 0$ with the same x-coordinate α are mapped to $(\alpha - S, \alpha + S)$ and can be recovered with a linear map (14). This makes the precision control much easier than the usual generic position method where the roots of $\Sigma = 0$ are represented as a univariate rational function of the roots of $T(X) = 0$.

3. ESTIMATION OF PARAMETERS s, R

From Section 2, we need to know the values of the parameters S, R, and s defined in (6) in order to transform the equation system into a local generic position. In this section, we will show how to compute such parameters efficiently.

We can use the general root bounds for zero dimensional equation systems in [20, p. 341] to estimate S and R. But the results obtained in this way is far from optimal. In this section, we will show how to obtain better estimations for S, R, and s.

We will use intervals to isolate the roots of a univariate equation. Let $\square\mathbb{Q}$ denote the set of intervals of the form $[a, b]$ where $a \leq b \in \mathbb{Q}$. The **length** of an interval $I = [a, b] \in \square\mathbb{Q}$ is defined to be $|I| = b - a$. A set **BS** of disjoint intervals is called **isolation intervals** for the roots of $t(x) = 0$ if each root of $t(x) = 0$ is in an interval in **BS** and each interval in **BS** contains one root of $t(x) = 0$.

Let $t(x)$ be defined in (4) and the isolating intervals for the roots of $t(x) = 0$ be

$$\mathbf{BS} \quad = \quad \{[a_1, b_1], \ldots, [a_m, b_m]\}. \tag{16}$$

We can directly estimate S from the isolating intervals for the roots of $t(x) = 0$:

$$S = \frac{1}{2} \min\{a_{i+1} - b_i, i = 1, \ldots, m - 1\}. \tag{17}$$

Then we have

Lemma 3.1. *Let α_i be the roots of $t(x) = 0$ and $[a_i, b_i]$ the isolation interval for α_i. If S is taken as (17), then the roots of $T(X) = 0$ associated with α_i are in the intervals $(a_i - S, b_i + S), i = 1, \ldots, m$.*

Proof. It is clear that the S defined in (17) satisfies (6). By Lemma 2.1, roots of $T(X) = 0$ are in $(\alpha_i - S, \alpha_i + S)$ for some i. Since $\alpha_i \in [a_i, b_i]$, the roots of $T(X) = 0$ associated with α_i must be in $(a_i - S, b_i + S)$. ▮

A simple way to estimate R is as follows:

$$R = \mathrm{RB}(h(y)), \text{ where } h(y) = \mathrm{Res}_x(f(x, y), g(x, y)) \tag{18}$$

and $\mathrm{RB}(h(x))$ is the root bound of $h(x)$. If $h(x) = c_0 y^d + \cdots + c_d$, then $\mathrm{RB}(h(y))$ can be taken as $1 + \max\{|c_1|, \ldots, |c_d|\}/|c_0|$ (page 322, [3]). In this method, we need to compute a resultant. When the degrees of f and g in x are low, we can use this approach. Otherwise, we can avoid the resultant computation by using the concept of sleeve functions (see [6, 16] for details). We will explain this approach below.

Given $f \in \mathbb{Q}[x, y]$, we decompose it uniquely as $f = f^+ - f^-$, where each $f^+, f^- \in \mathbb{Q}[x, y]$ has positive coefficients and with minimal number of monomials. Given an isolating interval $I = [a, b]$ for a root α of a univariate equation $t(x) = 0$, we define

$$\begin{aligned} f_I^u(y) = f^+(b, y) - f^-(a, y) \in \mathbb{Q}[y], \\ f_I^d(y) = f^+(a, y) - f^-(b, y) \in \mathbb{Q}[y]. \end{aligned} \tag{19}$$

Then $\Box f(I,y) = [f_I^d, f_I^u]$ is called a **sleeve** of $f(\alpha, y)$ due to the following reasons. We assume that $a, b \geq 0$ in the rest of our paper, since we can consider $F(x,y) = f(-x,y)$ for $-\alpha$ in $[-b,-a]$ when $a, b < 0$. When considering $y \geq 0$, the following result is clearly true (Fig. 2).

LEMMA 3.2. *We have* $f_I^d(y) \leq f(\alpha, y) \leq f_I^u(y)$*, or equivalently,* $f(\alpha,y) \in \Box f(I,y)$*. Furthermore, when* $|I|$ *approaches to zero, the interval* $\Box f(I,y)$ *converts to* $f(\alpha,y)$ *for each* y*.*

We can use the sleeve to estimate the root bound R.

LEMMA 3.3. *For* $\Sigma = \{f(x,y), g(x,y)\}$*, let* $I_i = [a_i, b_i]$*,* $i = 1,\ldots,m$ *be the intervals defined in (16). If* $a_i \geq 0$*,* $b_i \geq 0$*,* $f_{I_i}^d(y)$*,* $f_{I_i}^u(y)$ *have the same degree in* y*, and their leading coefficients in* y *have the same sign, then we can take*

$$R = \max\{RB(f_{I_i}^d), RB(f_{I_i}^u), RB(\bar{f}_{I_i}^d), RB(\bar{f}_{I_i}^u), i = 1,\ldots,m\} \tag{20}$$

where $\bar{f} = f(x,-y)$.

Proof. Consider the case that $d_i = \deg(f_{I_i}^d(y), y) = \deg(f_{I_i}^u(y), y)$ is odd and the leading coefficients of $f_{I_i}^u$ and $f_{I_i}^d$ are positive. Other cases can be treated similarly. Then, there exists a positive number r_1 such that $f_{I_i}^d(y) > 0$ for $y > r_1$. By Lemma 3.2, we have $f(\alpha_i, y) > f_{I_i}^d(y) > 0$ for $y > r_1$. Then, the largest positive root of $f(\alpha_i, y) = 0$ is bounded by $\mathrm{RB}(f_{I_i}^d(y))$. See Fig. 2 for an illustration.

Note that if c^u and c^d are the leading coefficients of $f_{I_i}^u$ and $f_{I_i}^d$, then the leading coefficients of $\bar{f}_{I_i}^u$ and $\bar{f}_{I_i}^d$ are $c^u(-1)^d = -c^u < 0$ and $c^d(-1)^d = -c^d < 0$ respectively, since d is odd. Then there exists a positive number r_2 such that $\bar{f}_{I_i}^d(y) > 0$ for $y > r_2$. By Lemma 3.2, we have $\bar{f}(\alpha_i, y) > \bar{f}_{I_i}^u(y) > 0$ for $y > r_2$. Then, the largest positive root of $\bar{f}(\alpha_i, y) = 0$, or equivalently, the absolute value of the smallest negative root of $f(\alpha_i, y) = 0$, is bounded by $\mathrm{RB}(\bar{f}_{I_i}^u(y))$. ∎

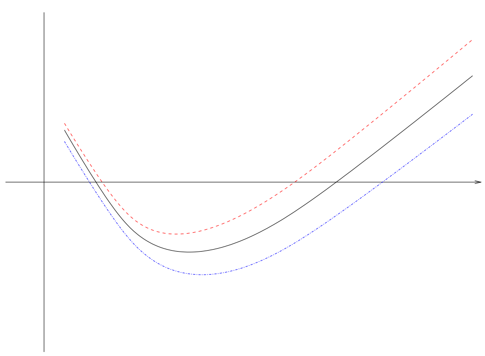

Figure 2: An illustration for sleeve: the dot curves are the sleeve for the solid curve.

In the above lemma, instead of $f(x,y)$, we can also use $g(x,y)$ to compute R. If $f_{I_i}^d(y)$ or $f_{I_i}^u(y)$ does not have the same degree, we can subdivide the interval I_i. When I_i is sufficiently small, they will have the same degree.

Based on the above result, we give the following algorithm to estimate R.

ALGORITHM 3.4. $\mathbf{RootB}(f(x,y), t(x), \mathbf{BS})$ *The inputs are* $f(x,y) \in \mathbb{Q}[x,y]$*,* $t(x)$ *defined in (4),* $\mathbf{BS}$ *isolation intervals of* $t(x) = 0$ *defined in (16). Output is an* R *satisfying (6).*

1. Without loss of generality, we assume that $t(x)$ is an irreducible polynomial. Otherwise, we will execute the following steps for the irreducible factors of $t(x)$ and output the maximal R obtained from these factors.
2. If $a_1 < 0$, do a translation $x := x + a_1$ and still use $t(x)$, $f(x,y)$, and $I_i = [a_i, b_i]$ to denote the translated polynomials and intervals.
3. Write $f = F(x)y^d + F_{d-1}(x)y^{d-1} + \cdots + F_0(x)$. We assume that $t(x)$ is not a factor of $F(x)$; otherwise, we may remove $F(x)y^d$ from f since we have $t(x) = 0$.
4. For each root $I \in \mathbf{BS}$, let $\alpha \in I$ be the root of $t(x) = 0$ in I. Then $F(\alpha) \neq 0$. We assume that $p = F_I^u F_I^d > 0$, where F_I^u and F_I^d are computed with (19); otherwise we repeatedly subdivide I and still denote I as the new interval containing α[1] until $p > 0$.
5. As a consequence, $f_I^u(y)$ and $f_I^d(y)$ have same degree and their leading coefficients have the same sign. Then, by Lemma 3.3, we compute R according to (20).

Proof of the correctness. The correctness is obvious. We just need to show that Step 4 will terminate when we subdivide I. By Lemma 3.2, the coefficients of $f_I^u(y)$ and $f_I^d(y)$ can approximate the coefficients of $f(\alpha, y)$ as close as we want. Since $F(\alpha) \neq 0$, when I is sufficiently subdivided, $F_I^u * F_I^d > 0$. And the program will terminate. ∎

Now, we show how to compute s which satisfies (6) and (10). One way to do this is as follows.

LEMMA 3.5. *Let* $d = \deg(f(x,y))$ *and* S, R *rational numbers satisfying (6). Then one of* $s_i = \frac{(3d+i)S}{(4d+2)R}, i = 1,\ldots,d+1$ *must satisfy (10) and thus can be used as* s*.*

Proof. Each s_i satisfies (6). Since $f_d(x,y)$ is homogenous and is of total degree d, $f_d(x,1) = 0$ can have at most d roots. Then, one of the s_i must satisfy $f_d(-s_i, 1) \neq 0$. ∎

Ideally, we want the bitsize of s to be as small as possible to make the computation of $T(X)$ easier. For instance, $\frac{1}{3}$ is much better than $\frac{1000004}{3000001}$. As a heuristic, we may take $s \in \mathbb{Q}$ satisfying (10) and with the smallest bitsize.

4. ROOT ISOLATION OF BIVARIATE POLYNOMIAL SYSTEMS

In this section, we will present the local generic position method for real root isolation. We first find the parameters R, S, and s, then obtain $T(x)$ with (8), and finally isolate the real roots of the equation system with (14) by isolating the real roots of $t(x) = 0$ and $T(X) = 0$.

Let $\Box\mathbb{Q}^2$ be the set of interval boxes of the form $[a,b] \times [c,d]$ where $[a,b], [c,d] \in \Box\mathbb{Q}$. The **length** of an interval box $\mathbf{B} = [a,b] \times [c,d] \in \Box\mathbb{Q}^2$ is defined to be $|\mathbf{B}| = \max\{b-a, d-c\}$.

Let $\Sigma = \{f(x,y), g(x,y)\}$ and $\xi = (\xi_1, \xi_2)$ be a root of $\Sigma = 0$. Then an interval box $\mathbf{B} = [a_1, b_1] \times [a_2, b_2] \in \Box\mathbb{Q}^2$ is called an **isolation box** of ξ if $\xi_i \in (a_i, b_i)$ and ξ is the only root of $\Sigma = 0$ in $\mathbf{B}$. A set $\mathbf{BS}$ of disjoint interval boxes is called **isolation boxes** for $\Sigma = 0$ if each real root of $\Sigma = 0$ is in a box in $\mathbf{BS}$ and each box in $\mathbf{BS}$ contains one root of $\Sigma = 0$. A set of root isolation boxes of $\Sigma = 0$ is called **ϵ-isolation boxes** if each box has size smaller than a given positive number ϵ.

In this section, we will present an algorithm to compute a set of ϵ-isolation boxes for $\Sigma = \{f(x,y), g(x,y)\}$.

In Theorem 2.3, roots of $\Sigma = 0$ are represented by algebraic numbers. In the following, we will give an interval version of this result, which leads to an algorithm directly.

Let the isolation boxes for α_i in (5) and $\beta_{i,j}$ in (11) be

$$\begin{aligned} \mathbf{B} &= \{[a_1,b_1],\ldots,[a_m,b_m]\} \\ \mathbf{B}_i &= \{[c_{i,1},d_{i,1}],\ldots,[c_{i,m_i},d_{i,m_i}]\}, i = 1,\ldots,m, \end{aligned} \tag{21}$$

respectively. Theorem 4.1 shows how to compute isolation boxes for $\Sigma = 0$.

[1]This can be easily done since $t(x)$ is irreducible.

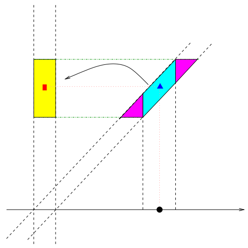

Figure 3: Recover an isolation box

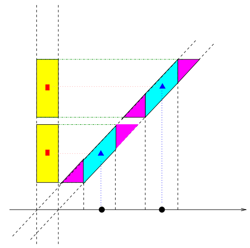

Figure 4: Separation of two boxes

THEOREM 4.1. *Let ϵ be a positive number and s a number satisfying (6) and (10). If the intervals in (21) satisfy*

$$\begin{aligned} b_i - a_i < \epsilon \text{ and } \quad & b_i - a_i + d_{i,j} - c_{i,j} < s\epsilon, \\ & i = 1, \dots, m, j = 1, \dots, m_i \qquad (22) \\ b_i - a_i < \min\{ \quad & c_{i,j+1} - d_{i,j}, j = 1, \dots, m_i - 1\}, \\ & i = 1, \dots, m \qquad (23) \end{aligned}$$

then a set of ϵ-isolation boxes for the roots $P_{i,j} = (\alpha_i, \beta_{i,j})$ of $\Sigma = 0$ are

$$\mathbf{B}_{i,j} = [a_i, b_i] \times [(c_{i,j} - b_i)/s, (d_{i,j} - a_i)/s]. \qquad (24)$$

Proof. Since $[a_i, b_i]$ is an isolation interval of α_i and $[c_{i,j}, d_{i,j}]$ is an isolation interval of $\beta_{i,j}$, from the basic rules of interval computation, we have $P_{i,j} = (\alpha_i, \beta_{i,j}) \in \mathbf{B}_{i,j}$. See Fig. 3 for an illustration. We need only to show that $|\mathbf{B}_{i,j}| < \epsilon$ and $\mathbf{B}_{i,j}$ are disjoint.

From (22), we have $|\mathbf{B}_{i,j}| = \max\{b_i - a_i, (d_{i,j} - a_i)/s - (c_{i,j} - b_i)/s\} < \max\{\epsilon, (b_i - a_i + d_{i,j} - c_{i,j})/s\} < \max\{\epsilon, \epsilon\} = \epsilon$.

Finally, we will show that different $\mathbf{B}_{i,j}$ are disjoint. If $i \neq k$, then it is obvious that $\mathbf{B}_{i,j}$ and $\mathbf{B}_{k,s}$ are disjoint. Consider $\mathbf{B}_{i,j}$ and $\mathbf{B}_{i,k}$ for $j \neq k$. From the construction procedure for $\mathbf{B}_{i,j}$, we need only to show that for $i = 1, \dots, m_i - 1$, $\mathbf{B}_{i,j}$ and $\mathbf{B}_{i,j+1}$ are disjoint which is equivalent to the condition $(d_{i,j} - a_i)/s < (c_{i,j+1} - b_i)/s$. See Fig. 4 for an illustration. Since $s > 0$, this is equivalent to $b_i - a_i < c_{i,j+1} - d_{i,j}$ which is valid by (23). ▮

It is well-known on how to isolate the real roots for a univariate polynomial equation, which is given as the following algorithm.

ALGORITHM 4.2. $\mathbf{RootIsolU}(t(x), \epsilon)$. *Input $t(x) \in \mathbb{Q}[x]$. Output the set of isolation intervals $[a_i, b_i], i = 1, \dots, m$ for the real roots of $t(x) = 0$ such that $|b_i - a_i| < \epsilon$ and $a_1 < a_2 < \cdots < a_m$.*

Now we can give the algorithm to compute the isolation boxes for $\Sigma = 0$.

ALGORITHM 4.3. $\mathbf{LGP}(\Sigma, \epsilon)$. *$\Sigma = \{f(x, y), g(x, y)\}$ is a zero-dimensional bivariate system and ϵ is a positive number. Output a set of ϵ-isolation boxes* **BS** *for the roots of $\Sigma = 0$.*

1. Let $t(x) = \mathrm{Res}_y(f(x, y), g(x, y))$.
2. Set $\rho_1 = \epsilon$ and compute $\mathbf{B}$, where $\mathbf{B} = \mathbf{RootIsolU}(t(x), \rho_1) = \{[a_1, b_1], \dots, [a_m, b_m]\}$.
3. Compute R with Algorithm **RootBN** with input $f, t, \mathbf{B}$.
4. Let $D = \frac{1}{2}\min\{|a_{i+1} - b_i|, i = 1, \dots, m - 1\}$.
   (a) If $D > 2\epsilon$, let $\epsilon_1 = \epsilon, S = D - \epsilon_1$ and compute s according to Lemma 3.5.
   (b) If $D \leq 2\epsilon$, let $\epsilon_1 = D/2, S = D - \epsilon_1$ and compute s according to Lemma 3.5.
5. Compute $T(X) = \mathrm{Res}_y(f(X - sY, y), g(X - sY, Y))$.
6. Set $\rho_2 = \min\{s\epsilon/2, \epsilon_1\}$ and compute $\mathbf{T} = \mathbf{RootIsolU}$ $(T(X), \rho_2) = \{[p_1, q_1], \dots, [p_t, q_t]\}$ and the multiplicities of the corresponding roots, if needed.
7. Compute $\rho = \min\{|p_{i+1} - q_i|, i = 1, \dots, t - 1\}$. Let $\theta = \min(\frac{\rho}{2}, \epsilon_1, s\epsilon/2)$. If $\rho_1 > \theta$, set $\rho_1 = \theta$ and compute $\mathbf{B} = \mathbf{RootIsolU}(t(x), \rho_1)$.
8. For each element $[p_j, q_j] \in \mathbf{T}$, there exists a unique $[a_i, b_i]$ in $\mathbf{B}$ such that $[p_j, q_j] \subset [a_i - D, b_i + D]$. Then $[a_i, b_i] \times [\frac{p_j - b_i}{s}, \frac{q_j - a_i}{s}]$ is an isolation box of one root of $\Sigma = 0$. And the multiplicity of the root of $\Sigma = 0$ in the isolation box is the multiplicity of the root of $T(X)$ in $[p_j, q_j]$. Let **BS** be the set of these boxes and return **BS**.

Proof of Correctness of Algorithm 4.3. From Step 2, we know $b_i - a_i < \epsilon$. From Steps 6 and 7, we know $b_i - a_i < s\epsilon/2$, $q_j - p_j < s\epsilon/2$ and hence $b_i - a_i + q_j - p_j < s\epsilon$. Then, condition (22) is valid. From Step 7, we know that condition (23) is also valid. From Steps 3 and 4, it is clear that conditions (6) and (10) are satisfied. Then, Theorem 4.1 can be used to compute the isolation boxes. What we need to do is to chose those $[p_j, q_j]$ which are associated with a given $[a_i, b_i]$, which is the purpose of Step 8.

We will show that Step 8 is correct. By Lemma 3.1, a root β_j of $T(X) = 0$ associated with a root α_i of $r(x) = 0$ is in $(a_i - S, b_i + S)$. By Step 6, the isolation interval $[p_j, q_j]$ of β_j satisfies $q_j - p_j < \epsilon_1$. By Step 4, $D = S + \epsilon_1$. Then $q_j < \beta_j + q_j - p_j < \beta_j + \epsilon_1 < b_i + S + \epsilon_1 = b_i + D$. Similarly, $p_j > a_i - D$. Therefore, each $[p_j, q_j]$ is in a unique $[a_i - D, b_i + D]$. The isolating box for a root (α_i, β_j) of $\Sigma = 0$ is formed based on (24). ▮

REMARK 4.4.
1. *In the algorithm, we may need to isolate the roots of $t(x) = 0$ twice. When isolating their roots in the second time, we need only subdivide the existing intervals. It is not necessary to start the isolation procedure from scratch.*
2. *An advantage of this method is that we need only to isolate the roots of $T(X) = 0$ once and isolate the roots of $t(x) = 0$ at most twice for a given specific precision. In other words, we do not need to repeatedly subdivide the isolation intervals as in most existing methods.*
3. *The most time consuming step of the algorithm is Step 5 and Step 6. There are two reasons for this. First, the shear transformation changes a spare polynomial into a dense one. Second, if the bitsize of s is large, the coefficients of $T(X)$ could be very large.*

EXAMPLE 4.5. *We use a simple example to illustrate the algorithm. Let $\Sigma = \{x^2 - y^2 - 1, 2x^2 + 3y^2 - 6\}$, and $\epsilon = 10^{-3}$.*

1. *$t(x) = (-5 * x^2 + 9)^2$.*
2. *$\rho_1 = 10^{-3}$ and $\mathbf{B} = \{[-\frac{687}{512}, -\frac{1373}{1024}], [\frac{1373}{1024}, \frac{687}{512}]\}$.*
3. *Compute R, we get $R = 1$.*
4. *$S = \frac{1373}{1024}$. Since $S > 2\epsilon$, we choose 1 to replace S in the computation of s. We obtain $s = 1$.*
5. *$T(X) = 5 * X^4 - 26 * X^2 + 5$.*
6. *$\rho_2 = s\epsilon/2 = 10^{-3}/2$ and $\mathbf{T} = \{[-\frac{1145}{512}, -\frac{4579}{2048}], [-\frac{229}{512}, -\frac{915}{2048}], [\frac{915}{2048}, \frac{229}{512}], [\frac{4579}{2048}, \frac{1145}{512}]\}$. The multiplicities of all the roots are one.*
7. *$\rho = \frac{3663}{2048}$. $\theta = \min\{\frac{3663}{4096}, 10^{-3}, 1 * 10^{-3}/2\} = 10^{-3}/2$. Since $\rho_1 > \theta$, refine $\mathbf{B}$ with $\rho_1 = \theta$ and derive $\mathbf{B} = \{[-\frac{687}{512}, -\frac{2747}{2048}], [\frac{2747}{2048}, \frac{687}{512}]\}$.*
8. *For each element of $\mathbf{T}$, recover the isolation box of the corresponding root of $\Sigma = 0$. Consider the first element $\mathbf{T}_1 = [-\frac{1145}{512}, -\frac{4579}{2048}]$. It is easy to check that $\mathbf{T}_1$ is associated with $\mathbf{B}_1 = [-\frac{687}{512}, -\frac{1373}{1024}]$. Then, the corresponding isolation box can be computed with (24), which is $[-\frac{687}{512}, -\frac{2747}{2048}]$ $\times[-\frac{1833}{2048}, -\frac{1831}{2048}]$. The multiplicity of the root of the system is one. In a similar way, we can find other isolation boxes.*

5. IMPLEMENTATION AND EXPERIMENTAL RESULTS

We have implemented Algorithm 4.3 as a software package LGP in Maple, which is available at http://www.mmrc.iss.ac.cn/~ xgao/software.html. Extensive experiments with this package show that this approach is efficient and stable, especially for bivariate equation systems with multiple roots.

We compare our method with Discoverer [19], GRUR[7], Hybrid [14], and Isolate[17]. Discoverer is a tool for solving problems about polynomial equations and inequalities. GRUR is a tool to solve bivariate equation systems. Hybrid is a numeric and symbolic hybrid algorithm for solving bivariate equation systems. Isolate is a tool to solve general equation systems based on the Realsolving C library by Rouillier.

We did three sets of experiments. All the results are collected on a PC with a 3.2GHz CPU, 2.00G memory, and running Microsoft Windows XP. We use Maple 12 in the experiments. The precision in the experiments is set to be 10^{-3}. In these experiments, f and g are generated as follows.

- Both f and g are randomly generated dense polynomials with the same degree and with integer coefficients between -99 and 99. The results are given in Fig. 5. In order to give more details about the results, we show the timings of Isolate, Hybrid, and LGP in Fig. 6 with a smaller time scaling.
- Both f and g are randomly generated sparse polynomials in the same degree, with sparsity 10%, and with integer coefficients between -99 and 99. The results are given in Fig. 7 and Fig. 8.
- The third set of experiments is done with polynomial systems with multiple roots. We randomly generate a polynomial $h(x, y, z)$ and take $f(x, y) = \mathrm{Res}_z(h, h_z)$, $g(x, y) = f_y(x, y)$. Since $f(x, y)$ is the projection of a space curve to the xy-plane, it most probably has singular points and $f = g = 0$ is an equation system with multiple roots. The results are given in Fig. 9 and Fig. 10.

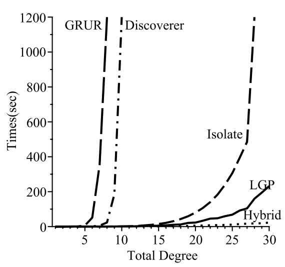

Figure 5: Σ consists of dense polynomials and has no multiple roots.

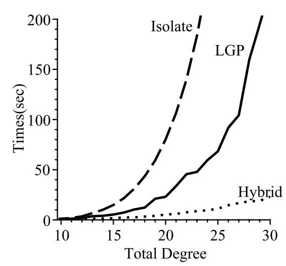

Figure 6: Same as Fig. 5, with a smaller time scaling.

For each possible degree, we generate ten examples and the results are the average values for the ten examples. According to Figures 5, 7, and 9, we have the following observations.

- In all cases, GRUR and Discoverer generally work for equation systems with degrees not higher than ten within reasonable time.
- In the first two cases, the equations are randomly generated and hence have no multiple roots. For systems without multiple roots, Hybrid is the fastest method, which is significantly faster than LGP and Isolate. Both Hybrid and LGP compute two resultants and isolate their real roots. LGP is slow, because the polynomials obtained by the shear map are usually dense and with large coefficients.

  We also observe that all methods spend more time with sparse polynomials than with dense polynomials in the same high degree. This phenomenon needs further exploration.
- For systems with multiple roots, LGP is the fastest method, which is significantly faster than Hybrid and Isolate. Note that our method is quite stable for equation systems with and without multiple roots. Isolate is also quite stable, but slower than LGP for bivariate equation systems.

Of course, we should mention that Discoverer and Isolate can be used to solve general polynomial equations and even inequalities. Our comparison is limited to the bivariate case.

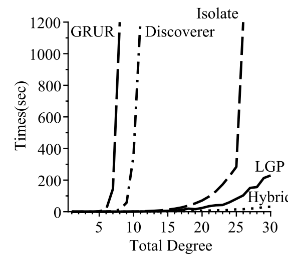

Figure 7: Σ consists of sparse polynomials and has no multiple roots.

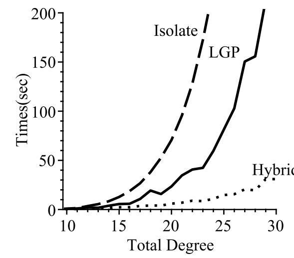

Figure 8: Same as Fig. 7, with a smaller time scaling.

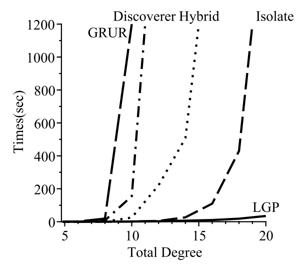

Figure 9: Σ is a system with multiple roots.

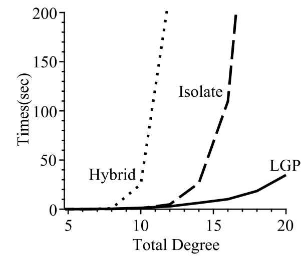

Figure 10: A smaller time scaling of Fig. 9.

Example 5.1. *A critical step to compute the topology for a plane curve is to determine its x-critical points. In this example, we compute all x-critical points of an algebraic curve $f = 0$ (the solid curve in Figure 11), which is defined below. The polynomial $f(x, y)$ is actually one of the examples used in the third set of experiments shown in Fig. 9. We need to solve the equation system $\Sigma = \{f, \frac{\partial f}{\partial y}\}$. The method LGP uses 9.6 seconds, Isolate uses 80.1 seconds, Hybrid uses 326.4 seconds, and GRUR and Discover both give no results in 1200 seconds with the same PC as mentioned before.*

$f = 16 + 40\,x - 72\,y - 119\,x^{12}y^2 - 4755\,x^6y^4 - 1803\,x^8y^5 - 983\,x^6y^7 + 4582\,x^4y^9 + 153\,x^7y^6 + 402\,x^{10}y^3 + 201\,xy^{12} + 39\,x^{11}y^2 + 6221\,x^5y^3 - 1692\,xy^{11} - 3216\,x^4y^8 + 4922\,x^6y^3 + 585\,x^{11}y^3 - 2301\,x^3y^{10} + 5749\,x^3y^9 + 6066\,x^2y^4 - 19\,x^{13}y + 1414\,xy^6 + 1516\,x^5y^9 - 2349\,xy^5 - 3315\,x^2y^3 + 1153\,x^8y^4 + 2757\,y^{10}x + 784\,x^6y^8 + 5439\,x^3y^6 - 818\,x^4y^4 - 131\,x^{12}y - 625\,x^8y^3 - 1174\,x^9y - 1244\,x^9y^5 - 16\,x^7y^5 - 3435\,x^4y^5 + 337\,x^2y^{12} + 1165\,x^8y^6 - 2276\,x^4y^{10} - 420\,x^7y^7 - 348\,x^{10}y^4 + 309\,xy^{13} + 3246\,x^6y^5 + 1597\,x^7y^3 + 1055\,x^8y^2 + 98\,x^5y^8 - 234\,x^3y^{11} - 401\,x^{11}y + 659\,x^{10}y^2 - 847\,x^{10}y + 559\,x^2y^{10} - 1542\,x^5y^7 - 1075\,x^6y^6 - 486\,x^9y^3 - 665\,x^7y^4 + 1821\,x^2y^9 + 294\,x^5y^6 - 616\,x^4y^7 - 5413\,x^3y^8 + 1441\,x^9y^2 - 2176\,xy^9 - 2641\,x^2y^8 + 2670\,x^5y^5 + 3845\,x^4y^6 + 628\,x^3y^7 + 902\,xy^8 - 880\,x^8y - 983\,x^7y^2 + 190\,x^2y^7 - 5890\,x^5y^4 + 75\,x^7y + 826\,x^6y - 617\,xy^7 - 8154\,x^3y^5 + 3832\,x^2y^6 + 3728\,x^4y^3 + 918\,x^5y - 2759\,x^4y^2 - 3400\,x^5y^2 - 984\,x^3y^2 + 2641\,xy^4 + 879\,x^4y - 1269\,x^2y^{11} + 450\,x^9y^4 + 640\,x^3y + 829\,x^2y^2 + 44\,x^2y + 992\,xy^2 - 284\,xy - 919\,x^3y^3 - 6516\,x^2y^5 + 5401\,x^3y^4 - 24\,x^{14}y^2 - 2896\,x^6y^2 - 256\,y^3 - 116\,x^3 - 56\,x^2 - 113\,x^4 + 172\,y^2 + 238\,y^4 - 134\,y^5 + 305\,x^9 - 107\,x^5 + 222\,x^8 - 94\,x^6 + 105\,y^6 +$

$$55\,x^7+993\,y^8-1564\,y^9+81\,x^{12}+157\,x^{11}+1497\,y^{10}+250\,x^{10}+179\,y^{13}-96\,y^{14}+50\,y^{12}+34\,x^{13}+9\,x^{14}-766\,y^{11}-380\,y^7+3\,x^{15}+19\,y^{15}-2014\,y^3x-324\,x^{11}y^4+281\,x^{12}y^3-76\,x^{13}y^2+201\,x^{10}y^6-41\,x^{11}y^5-111\,x^9y^7+3\,x^{14}y+6\,x^{15}y-64\,x^{12}y^4+49\,x^{13}y^3+60\,x^4y^{12}+66\,x^5y^{11}+26\,x^3y^{13}+8\,x^8y^8-136\,x^6y^{10}+175\,x^2y^{13}+17\,x^7y^9-138\,xy^{14}-53\,x^2y^{14}-290\,x^8y^7-882\,x^5y^{10}+183\,x^6y^9+144\,x^4y^{11}+832\,x^9y^6-355\,x^{10}y^5+236\,x^7y^8+245\,x^3y^{12}+16\,xy^{15}.$$

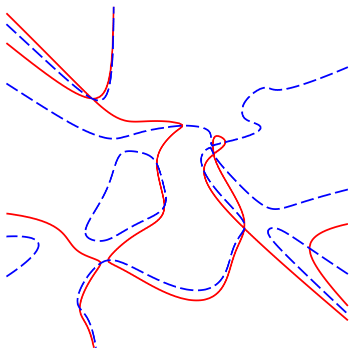

Figure 11: The intersections of $f=0$ and $\frac{\partial f}{\partial y}=0$.

6. CONCLUSION

In this paper, we propose a local generic position method to solve bivariate polynomial equation systems. The method can be used to represent the roots of a bivariate equation system as the linear combination of the roots of two univariate equations. As a result, root isolation for bivariate systems is reduced easily to root isolation of univariate equations. The multiplicities of the roots are also derived.

The results of this paper can be extended to isolate the real roots of bivariate equation systems with more than two polynomials by using the resultant systems for several polynomials given in [18]. It is also possible to extend the method to multivariate equation systems. But, the procedure is very complicated. It is an interesting problem to give a simple and effective algorithm for multivariate equation solving based on the idea of local generic position.

Acknowledgement

The authors would like to thank Prof. Hoon Hong and Ms. Meijing Shan for providing the Maple code of their method.

7. REFERENCES

[1] M. E. Alonso, E. Becker, M. F. Roy, and T. Wörmann. Zeros, multiplicities, and idempotents for zerodimensional systems. In *Algorithms in Algebraic Geometry and Applicatiobns*, pages 1–15. Birkhauser, 1996.

[2] D. S. Arnon, G. Collins, and S. McCallum. Cylindrical algebraic decomposition, ii: an adjacency algorithm for plane. *SIAM J. on Comput.*, 13(4):878–889, November 1984.

[3] S. Basu, R. Pollack, and M. F. Roy. *Algorithms in Real Algebraic Geometry*. Springer, Berlin, 2003.

[4] E. Becker and T. Wörmann. Radical computations of zero-dimensional ideals and real root counting. *Mathematics and Computers in Simulation*, 42(4-6):561–569, November 1996.

[5] J. F. Canny. Some algebraic and geometric computation in pspace. In *ACM Symp. on Theory of Computing*, pages 460–469. SIGACT, 1988.

[6] J. S. Cheng, X. S. Gao, and C. K. Yap. Complete numerical isolation of real roots in zero-dimensional triangular systems. *Journal of Symbolic Computation*, 44(7):768–785, July 2009.

[7] D. I. Diochnos, I. Z. Emiris, and E. P. Tsigaridas. On the complexity of real solving bivaraite systems. In *ISSAC 2007*, pages 127–134. ACM, 2007.

[8] A. Eigenwillig, M. Kerber, and N. Wolpert. Fast and exact geometric analysis of real algebraic plane curves. In *ISSAC 2007*, pages 151–158. ACM, 2007.

[9] W. Fulton. *Introduction to Intersection Theory in Algebraic Geometry*. Providence, R.I, Washington, DC, 1984.

[10] X. S. Gao and S. C. Chou. On the theory of resolvents and its applications. *Sys. Sci. and Math. Sci.*, 12:17–30, 1999.

[11] X. S. Gao and M. Li. Rational quadratic approximation to real algebraic curves. *Computer Aided Geometric Design*, 21(8):805–828, 2004.

[12] M. Giusti and J. Heintz. Algorithmes - disons rapides -pour la dècomposition d'une variètè algébrique en composantes irréducibles et équidimensionnelles. In *Proc MEGA' 90*, pages 169–193. Birkhäuser, 1991.

[13] L. Gonzalez-Vega and I. Necula. Efficient topology determination of implicitly defined algebraic plane curves. *Computer Aided Geometric Design*, 19(9):719–743, 2002.

[14] H. Hong, M. Shan, and Z. Zeng. Hybrid method for solving bivariate polynomial system. In *SRATC 2008*, pages http://www.is.pku.edu.cn/˜xbc/SRATC2008/meijing.pdf., 2008.

[15] H. Kobayashi, S. Moritsugu, and R. W. Hogan. Solving systems of algebraic equations. In *Proc. of ISSAC 1988*, pages 139–149. ACM, 1988.

[16] Z. Lu, B. He, Y. Luo, and L. Pan. An algorithm of real root isolation for polynomial systems. In *Proc. SNC'05*, pages 94–107, 2005.

[17] F. Rouillier. Solving zero-dimensional systems through the rational univariate representation. *Applicable Algebra in Engineering, Communication and Computing*, 9(5):433–461, May 1999.

[18] B. L. Van der Waerden. *Einfürung in die Algebraischen Geometrie*. Springer, Berlin, 1973.

[19] B. Xia and L. Yang. An algorithm for isolating the real solutions of semi-algebraic systems. *Journal of Symbolic Computation*, 34(5):461–477, 2002.

[20] C. K. Yap. *Fundamental problems of algorithmic algebra*. Oxford Press, 2000.

[21] K. Yokoyama, M. Noro, and T. Takeshima. Computing primitive elements of extension fields. *ournal of Symbolic Computation*, 8(6):553–580, 1989.