MM Research Preprints
KLMM, AMSS, Academia Sinica
Vol. 26, 1–19, August, 2008

Topology Determination and Isolation for Implicit Plane Curves

Jin-San Cheng, Xiao-Shan Gao, Jia Li
Key Laboratory of Mathematics Mechanization
Institute of Systems Science, AMSS, Chinese Academy of Sciences

Abstract. A method is proposed to generate an isolation for a plane curve, which is a set of boxes covering the curve, having the same topology as the curve, and approximating the curve to any given precision. The method uses symbolic computation to guarantee correctness and uses interval analysis whenever possible to enhance efficiency. This leads to a quite effective hybrid method for plane curve isolation.

Keywords. Plane curve, topology, isolating box, certified meshing, interval analysis.

1. Introduction

Let $\mathcal{C}_f : f(x,y) = 0$ be a real plane curve defined by a bivariate polynomial $f(x,y)$ with rational number as coefficients. Determining the topology of $\mathcal{C}_f$ is a basic problem in computational geometry and geometric modeling.

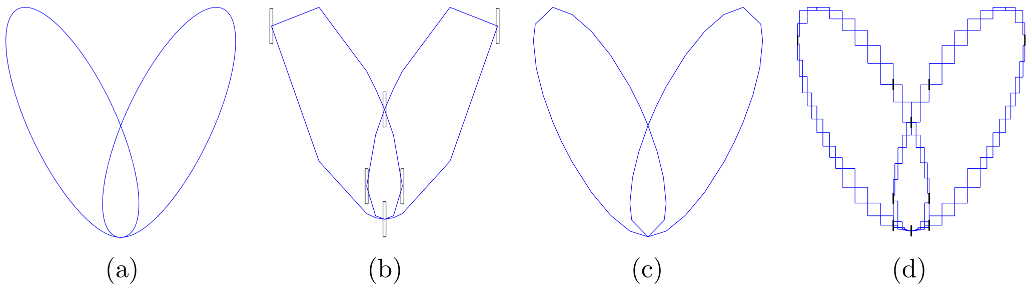
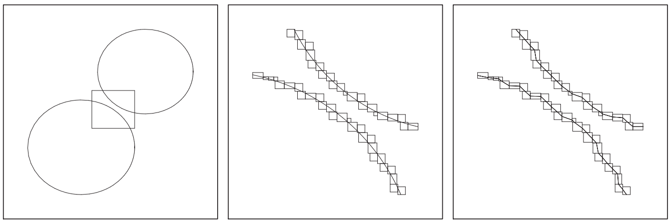


Fig. 1. Different types of approximations to a curve.

Depending on the applications, three types of output could be given. The simplest one is a plane graph which has the same topology as $\mathcal{C}_f$. The diagram in Fig. 1(b) is such a graph for the curve in Fig. 1(a). Another output is a **certified meshing** to $\mathcal{C}_f$, which is a topologically correct piecewise linear approximation with any given precision, as shown in Fig. 1(c). The third output is an **isolation** for $\mathcal{C}_f$, which is a set of boxes covering $\mathcal{C}_f$ and having the same topology as $\mathcal{C}_f$. An example of an isolation is given in Fig. 1(d). The isolation is more difficult to compute but has better properties and more applications. For instance, an isolation separates two curve branches close to each other, while a certified meshing cannot do this if the precision is not small enough (See Fig. 2). This separation property is useful in collision detection between two curves and motion planning.



(a) (b) (c)

Fig. 2. (a): Two circles $(x^2+y^2-25)*((x-6)^2+(y-8)^2-20)=0$, (b): isolating boxes, (c): meshing.

There exist many approaches in the literature to determine the topology for plane curves [1, 2, 3, 4, 5, 6, 8, 9, 10, 11, 12, 13, 14, 17, 18, 19, 20, 22]. Two main approaches are the subdivision approach and the cylindrical algebraic decomposition (CAD) approach.

The subdivision approach uses interval analysis to check the valuation range of $f(x,y)$ on a box $\mathbf{B}$, denoted as $\Box f(\mathbf{B})$. If $0 \notin \Box f(\mathbf{B})$, then $\mathbf{B} \cap \mathcal{C}_f = \emptyset$ and we can discard $\mathbf{B}$. Otherwise, subdivide $\mathbf{B}$ until certain termination condition is satisfied. This approach has two ingredients: the method to compute $\Box f(\mathbf{B})$ and the termination condition. Many interesting methods are proposed to compute $\Box f(\mathbf{B})$ and an excellent survey is given in [16]. For the termination condition, the simplest one is to give a threshold, say, the size of a pixel. And if the box is smaller than the threshold, the algorithm will terminate. Advanced termination criteria are proposed by Snyder and Plantinga-Vegter [21, 18]. The main drawback of the subdivision approach is that it does not guarantee correctness near a singular point of $\mathcal{C}_f$. In [6], a complete subdivision method is proposed. But, the method is not practical due to worst case estimation of zero bounds.

The CAD approach is based on the pioneering work of Collins, which can be used to divide the plane into cylindrical cells such that $f(x,y)$ has the same sign on each of the cells. Then to determine the topology of the curve, we need only to give the adjacency information between the cells [3, 4]. The CAD based methods have two major ingredients. First, determine the x-critical points of the curve, that is, points P satisfying $f(P)=f_y(P)=0$. Second, determine how these points are connected. CAD based methods are complete in the sense that it always computes the correct topology. On the other hand, it is generally slower than the subdivision method due to symbolic computation.

Most subdivision methods give certified meshing as the output. Most CAD based methods compute a topological graph for the curve. The methods in [1, 6, 11] can also give a certified meshing for a curve.

In this paper, we emphasize on computing an isolation for a given curve. We use symbolic computation to guarantee the correctness near singularities and use interval analysis whenever possible to enhance the efficiency. Such a hybrid approach is also used in [1]. Such a hybrid approach is also used in [1]. Our method is new in the following aspects. First, the idea of segregating boxes for critical points of curves proposed in [3] and improved in [15, 20] is extended to isolating boxes for curve segments. We also propose an interval based method to compute such isolating boxes. Second, we introduce a new subdivision method to generate isolating boxes for curve without critical points. The methods are implemented in Maple and extensive experiments show that our approach is very efficient.

The rest of this paper is organized as follows. In Section 2., we give an overview of our method. In Section 3., we show how to compute an isolation for the curve near a singularity point. In Section 4., we show how to compute an isolation for curves without critical points. In Section 6., we conclude the paper.

2. Overview of our algorithm

Let $\mathcal{C}_f : f(x,y) = 0$ be a plane curve where $f(x,y) \in \mathbb{Q}[x,y]$ is a square-free polynomial. We further assume that $f(x,y)$ has no factors of the form $u(x) \in \mathbb{Q}[x]$ or $v(y) \in \mathbb{Q}[y]$, which means that the curve has no horizontal or vertical line components. Otherwise, we may consider the curve $f(x,y)/(u(x)v(y))$ and then add the information about $u(x) = 0$ and $v(y) = 0$ later, which is easy.

We use intervals to isolate real numbers: let $\square\mathbb{Q}$ denote the set of intervals of the form $[a,b]$ where $a < b \in \mathbb{Q}$. The **length** of an interval box $\mathbf{B} = [a_1,b_1] \times [a_2,b_2] \in \square\mathbb{Q}^2$ is defined to be $|\mathbf{B}| = \max_i(b_i - a_i)$. The edges of a box $\mathbf{B}$ is denoted by $\partial\mathbf{B}$.

We will consider the part of $\mathcal{C}_f$ in a bounding box

$$\mathbb{B} = [\mathcal{X}_1, \mathcal{X}_2] \times [\mathcal{Y}_1, \mathcal{Y}_2] \in \square\mathbb{Q}^2 \tag{1}$$

which is denoted as $\mathcal{C}_{\mathbb{B}} = \mathcal{C} \cap \mathbb{B}$. In this paper, $\mathbb{B}$ is always assumed to be of this form.

For given f and $\mathbb{B}$, we will compute a set $\mathbf{BS}$ of boxes satisfying the following properties.

C1 $\mathbf{BS}$ covers $\mathcal{C}_{\mathbb{B}}$. That is, $\mathcal{C}_{\mathbb{B}} \subset R(\mathbf{BS}) = \cup_{\mathbf{B}\in\mathbf{BS}}\mathbf{B}$.

C2 Two boxes $\mathbf{B}_1$ and $\mathbf{B}_2$ in $\mathbf{BS}$ are either disjoint or overlap at one of their edges, and in the later case, $\mathcal{C}_f$ meets the overlapping part in one point, that is, $|\partial\mathbf{B}_1 \cap \partial\mathbf{B}_2 \cap \mathcal{C}_f| = 1$.

C3 The region covered by the boxes, that is $R(\mathbf{BS})$, has the same topology as $\mathcal{C}_{\mathbb{B}}$.

A set of boxes $\mathbf{BS}$ satisfying the above conditions is called a set of **isolating boxes**, or simply an **isolation**, for $\mathcal{C}_{\mathbb{B}}$ (See Fig. 1 and Fig. 2). Furthermore, if each box in $\mathbf{BS}$ has size smaller than a give precision ϵ, then $\mathbf{BS}$ is called an **ϵ-isolation** for $\mathcal{C}_{\mathbb{B}}$.

The first step of our algorithm is to isolate the critical points of $\mathcal{C}_f$. A point P is an **x-critical (y-critical) point** of $\mathcal{C}_f$ if $f(P) = f_y(P) = 0$ ($f(P) = f_x(P) = 0$). A **critical point** is either an x-critical point or a y-critical point. If P is both x-critical and y-critical, then it is a **singular point** of $\mathcal{C}_f$.

For purposes that will be explained in the second step, we will isolate the critical points, not the x-critical points as in most previous work. Let $g(x) = \mathrm{Res}_y(f, \frac{\partial f}{\partial y})$,$h(x) = \mathrm{Res}_y(f, \frac{\partial f}{\partial x})$. Then, we use the method given in [7] to isolate the real zeros for the triangular system $\Sigma = \{g(x)h(x), f(x,y)\}$.

Following the idea proposed in [3, 15, 20], we require that the isolating box for a critical point to be **segregating** in the sense that the top and bottom edges of the box do not meet the curve (Fig. 5). We extend the concept of segregating boxes to isolating boxes for curve segments and propose an interval analysis method to compute such boxes.

After the first step, all critical points of $\mathcal{C}_{\mathbb{B}}$ are contained in a set of boxes $\mathbf{SB}$. Then C_f has no critical points in the region $R_N = \mathbb{B} \setminus \cup_{\mathbf{S}\in\mathbf{SB}}\mathbf{S}$. In our case, R_N is the union of boxes.

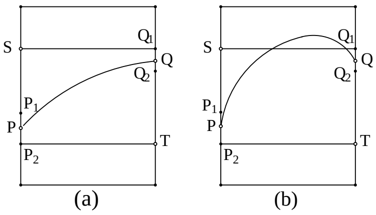


Fig. 3. Generating isolating boxes.

The second step of our algorithm is to isolate $\mathcal{C}_f$ in a box $\mathbf{B}$ where $\mathcal{C}_\mathbf{B}$ has no critical points. We propose two methods to do this.

The first method is conceptually simple. Let $\mathbf{B} = [a, b] \times [c, d] \in \square\mathbb{Q}^2$. Then we divide $[a, b]$ into smaller intervals $a_0 = a < a_1 < \cdots < a_m = b$ and isolate the roots of $f(a_i, y) = 0$. Since there exist no critical points in $\mathbf{B}$, $f(a_i, y)$ has a fixed number of real roots, say m. Let $[c_{i,j}, d_{i,j}], j = 1, \ldots, m$ be the isolating intervals for these roots. Since $\mathbf{B}$ contains no critical points,

$$\mathbf{B}_{i,j} = [a_i, a_{i+1}] \times [\min\{c_{i,j}, c_{i+1,j}\}, \max\{d_{i,j}, d_{i+1,j}\}]$$

cover the curve, as shown by Fig. 3(a). If $\mathbf{B}$ contains a y-critical point, then $\mathbf{B}_{i,j}$ do not necessarily cover the curve, as shown by Fig. 3(b). We prove that when $[a, b]$ is sufficiently subdivided, $\mathbf{B}_{i,j}$ will form an isolation for $\mathcal{C}_\mathbf{B}$. The drawback of this method is that it needs to isolate the roots of many univariate polynomial equations.

The second method is based on the idea of marching cube. Since box $\mathbf{B} = [a, b] \times [c, d]$ contains no critical points of $\mathcal{C}_f$, after sufficient subdivision, $\mathbf{B}$ can satisfy the following condition

$$0 \notin \square f_y(a, [c, d]), 0 \notin \square f_y(b, [c, d]), 0 \notin \square f_x([a, b], c), \text{ and } 0 \notin \square f_x([a, b], c)$$

where $\square f_x([a, b], c)$ is the range function [16]. If the above condition is satisfied, then $f(x, y)$ is monotone on each edge of $\mathbf{B}$. As a consequence, for an edge PQ of $\mathbf{B}$, if $f(P)f(Q) > 0$ $AB \cap \mathcal{C}_f = \emptyset$, otherwise PQ intersects $\mathcal{C}_f$ at one point. From this, we can obtain boxes each of which contains only one curve segment of $\mathcal{C}_f$ and these boxes form an isolation for the curve. The method is inspired by the work of Snyder [21], but is more efficient, because we need only to compute the range function on the edges of the boxes.

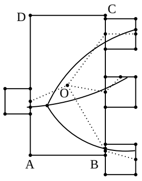


Fig. 4. Meshing the curve

Finally, the boxes generated from the above two steps form an isolation for the curve. If needed, we can easily obtain a **certified meshing** to $\mathcal{C}_f$. This meshing generation is easy due to condition **C2** in the definition of isolating boxes. We first construct the **auxiliary**

points. If two boxes $\mathbf{B}$ and $\mathbf{S}$ overlap, $\partial\mathbf{B} \cap \partial\mathbf{S}$ must be a line segment PQ. Let M be the midpoint of PQ, which is an auxiliary point. The second step is to connect the auxiliary points. If box $\mathbf{B}$ contains a critical point, we will add the line segments OP where O is the center of $\mathbf{B}$ and P an auxiliary point on $\partial\mathbf{B}$. If box $\mathbf{B}$ does not contain a critical point, due to the isolating property, $\mathbf{B}$ must have two auxiliary points P and Q. We just need to add PQ. In Fig. 4, the dotted lines are the lines segments generated in this way. Also see Fig. 1(d) and Fig. 2(c) for illustrations.

A distinctive feature of our approach is that we do not need to compute the number of curve branches connecting to a singular point. This information is implicitly given in our output, that is, the number of curve branches connecting to a singular point S is the number of boxes overlapping with the isolation box of P. See Fig. 4 for an illustration.

3. Isolate the curve near critical points

Let $\mathcal{C}_f : f(x, y) = 0$ be a curve defined by a square-free polynomial $f(x, y)$. In this section, we will show how to compute isolating boxes for $\mathcal{C}_f$ near its critical points.

3.1. Three types of isolating boxes

Let $\mathbf{BS}$ be an isolation for $\mathcal{C}_f$. Then each box $\mathbf{B} \in \mathbf{BS}$ must be an **isolating box** for $\mathcal{C}_f$ in the sense that $\mathbf{B}$ and $\mathcal{C}_\mathbf{B}$ have the same topology. In this section, we will define these isolating boxes used in our algorithm.

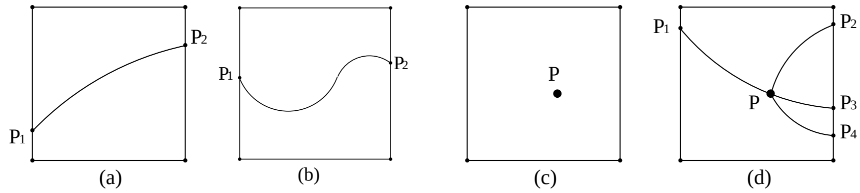


Fig. 5. Isolating boxes. (a) type one, (b) not isolating, (c) type two, (d) type three.

A box $\mathbf{B} = [a, b] \times [c, d] \in \square\mathbb{Q}^2$ is called **segregating** w.r.t $\mathcal{C}_f$ if

$$\mathcal{C}_f \cap [a, b] \times [c, c] = \mathcal{C}_f \cap [a, b] \times [d, d] = \emptyset.$$

A box $\mathbf{B}$ is called an **isolating box** for $\mathcal{C}_f$ if it satisfies one of the following conditions.

- $\mathcal{C}_\mathbf{B}$ is a single curve segment whose end points are on the edges of $\mathbf{B}$. Furthermore, $\mathcal{C}_\mathbf{B}$ can be considered as a **monotone** function in x: $y = \theta(x)$, as shown in Fig. 5(a). Such a curve segment of $\mathcal{C}_f$ is called a **monotone segment** and B is called a **type one isolating box** of $\mathcal{C}_f$. The box in Fig. 5(b) is not an isolating box, because $y = \theta(x)$ is not a monotone function in x.

- $\mathbf{B}$ contains an isolated point of $\mathcal{C}_f$, as shown in Fig. 5(c). B is called a **type two isolating box** of $\mathcal{C}$.

- $\mathcal{C}_\mathbf{B}$ consists of a critical point P of $\mathcal{C}_f$ and several monotone curve segments connecting P and points on the left or right edges of $\mathbf{B}$, as shown in Figure 5(d). $\mathbf{B}$ is called a

type three isolating box of $\mathcal{C}$. Note that a type three isolating box is segregating w.r.t $\mathcal{C}_f$.

Type three isolating boxes are used in [3, 15, 20] to count curve branches connecting to a critical point.

We will give a criterion for a box to be isolating. A point is called an x- or y-**extremal point** of curve $\mathcal{C}_f$ if $\mathcal{C}_f$ achieves a local extremum value at this point in the x- or y-direction. We have

Lemma 3.1 *Let* $\mathbf{B} \in \Box\mathbb{Q}^2$ *be a box containing no critical points of* $\mathcal{C}_f$. *Then* $\mathcal{C}_\mathbf{B}$ *consists of disjoint monotone curve segments with end points on the edges of* $\mathbf{B}$ *(See Fig. 6(a) for an illustration). Furthermore, if* $\mathcal{C} \cap \mathbf{B}$ *contains only one curve segment,* $\mathbf{B}$ *is a type one isolating box for* $\mathcal{C}$.

Proof. Let $\{\mathcal{C}_1, \ldots \mathcal{C}_t\}$ be the connected curve segments in $\mathcal{C}_\mathbf{B}$. Then $\mathcal{C}_i \cap \mathcal{C}_j = \emptyset$ for $i \neq j$, since intersection points are critical points of $\mathcal{C}_f$. Now consider one of the $\mathcal{C}_i$, say $\mathcal{C}_1$. Since there exist no x-critical points, for each x there exists at most one y such that $(x, y) \in \mathcal{C}_1$. Since $\mathcal{C}_1$ is connected, $\mathcal{C}_1$ can be written as a continuous function $y = \theta_1(x)$. Since there exist no y-critical points, $\mathcal{C}_1$ has no y-extremal points inside $\mathbf{B}$, or equivalently, $y = \theta_1(x)$ is monotone. The endpoints of $\mathcal{C}_1$ must be on the edges of $\mathbf{B}$. Otherwise, they are extremal points and thus critical points of $\mathcal{C}_f$, which is impossible. We have proved the first part of the Lemma. If $\mathcal{C}_\mathbf{B}$ consists of only one curve segment, then by the result we just proved, $\mathbf{B}$ is a type one isolating box for $\mathcal{C}$. ▮

Lemma 3.2 *Let* $\mathbf{B} = [a, b] \times [c, d] \in \Box\mathbb{Q}^2$ *be a box segregating w.r.t* $\mathcal{C}$, $P = (\alpha, \beta) \in \mathcal{C}$ *a point inside* $\mathbf{B}$, $\mathcal{C} \cap ([\alpha, \alpha] \times [c, d]) = \{P\}$, *and* $\mathcal{C}_\mathbf{B} \setminus \{P\}$ *contains no critical points of* $\mathcal{C}$. *Then* $\mathbf{B}$ *is either a type two or a type three isolating box of* $\mathcal{C}$.

Proof. Since there exist no critical points in $\mathbf{B}_1 = [a, \alpha) \times [c, d]$, by Lemma 3.1, $\mathcal{C} \cap \mathbf{B}_1$ consists of disjoint and monotone curve segments with their left endpoints on the left hand side edge of $\mathbf{B}$. Since $\mathcal{C} \cap [\alpha, \alpha] \times [c, d] = \{P\}$, each curve segment in $\mathcal{C} \cap \mathbf{B}_1$ must be with P as their right end point. Otherwise, the curve will contain x-extremal points in $\mathbf{B}_1$, which is impossible since such points are x-critical points. As a consequence, $\mathcal{C} \cap \mathbf{B}_1$ contains disjoint and monotone curve segments starting from P and ending at some points on the left edge of $\mathbf{B}$. Curve segments in $\mathcal{C} \cap \mathbf{B}_2 = (\alpha, b] \times [c, d]$ can be treated similarly. Therefore, if there exist curve segments in $\mathcal{C} \cap \mathbf{B}$, $\mathbf{B}$ is a type three isolating box. Otherwise, it is a type two isolating box. ▮

3.2. Computing isolating boxes of types two and three

In order to compute isolating boxes, we need an algorithm to isolate the real roots of equation systems.

Let $\Sigma = \{h(x), f(x, y)\}$ be a **triangular system** where $h \in \mathbb{Q}[x]$ and $f \in \mathbb{Q}[x, y]$. By **isolating the real roots** of $\Sigma = 0$, we mean to compute a set of disjoint boxes **BS** such that each real root of $\Sigma = 0$ is contained in one box in **BS** and each box in **BS** contains only one root of $\Sigma = 0$. The algorithm from [7] is used to isolate the real roots of triangular systems. For convenience, we always assume that the root is not on the edges of its isolating box. We write this as the following algorithm.

Algorithm 3.3 RootIsol$(\Sigma, \mathbb{B}, \epsilon)$. *$\Sigma = \{h(x), f(x,y)\}$ is a triangular system of polynomials in $\mathbb{Q}[x,y]$, $\mathbb{B}$ is a box as defined in (1), ϵ is a positive number. Output a set of isolation boxes $\mathbf{P}$ for the real zeroes of Σ in $\mathbb{B}$ and each box with sizer smaller than ϵ.*

We also need the **inclusion function**. For a box $\mathbf{B} \in \square\mathbb{Q}^n$ and a polynomial $f(x_1, \ldots, x_n)$, an inclusion function $\square f(\mathbf{B})$ returns an interval in $\square\mathbb{Q}$ such that $f(P) \in \square f(\mathbf{B})$ for all $P \in \mathbf{B}$ and $|\square f(\mathbf{B})|$ approaches to zero when $|\mathbf{B}|$ approaches to zero. For methods to construct such functions, please consult [16].

In the following, we will give an algorithm to compute isolating boxes for the curve near its critical points. All type two and type three isolating boxes will be computed in this algorithm. We will first give the algorithm and then explain it below.

Algorithm 3.4 IsoBoxC$(f(x,y), \mathbb{B}, \epsilon)$. *$\mathcal{C} : f(x,y) = 0$ is the curve, $\mathbb{B}$ is defined in (1), and ϵ is a positive number. Output boxes $\mathbf{B}_i \subset \mathbb{B}$ containing all the critical points of $\mathcal{C}_{\mathbb{B}}$ and ϵ-isolations for $\mathcal{C}_{\mathbf{B}_i}$.*

1. Let $g(x) = \mathrm{Res}_y(f, \frac{\partial f}{\partial y})$, $h(x) = \mathrm{Res}_y(f, \frac{\partial f}{\partial x})$.
2. Let $\{\mathbf{B}_{i,j} = [a_i, b_i] \times [c_{i,j}, d_{i,j}], i = 1, \ldots, m, j = 1, \ldots, m_i\} =$ **RootIsol**$(\{g(x)h(x), f(x,y)\}, \mathbb{B}, \epsilon)$.
3. While $0 \in \square f([a_i, b_i], c_{i,j})$ or $0 \in \square f([a_i, b_i], d_{i,j})$, repeat $\{[a_1, b_1], \ldots, [a_m, b_m]\} =$ **RootIsol**$(g(x)h(x), \rho)$, where $\rho = (b_i - a_i)/2$.
4. Output $\mathbf{B}_i = [a_i, b_i] \times [\mathcal{Y}_1, \mathcal{Y}_2]$ and $\mathbf{B}_{i,j}$.

In Step 1, the critical points of $\mathcal{C}_f$ are projected to the x-axis. In Step 2, we compute the isolating boxes for the points of $\mathcal{C}_f$ lifted from the roots of $g(x)h(x) = 0$. In Step 3, we refine these isolating boxes until they become segregating.

The correctness of Steps 1 and 2 is obvious. In Step 3, let $\mathbf{B}_{i,j}$ be an isolating box for a root $R = (\alpha, \beta)$ of $gh = f = 0$. Since R is not on the edges of $\mathbf{B}_{i,j}$, we have $f(\alpha, c_{i,j})f(\alpha, d_{i,j}) \neq 0$. If $0 \notin \square f([a_i, b_i], c_{i,j})$ and $0 \notin \square f([a_i, b_i], d_{i,j})$, then $\mathbf{B}_{i,j}$ is segregating. Otherwise, by the convergent property of the inclusion function, this step will terminate and give isolating boxes for $\mathcal{C}_f$ after sufficient refinements. This method to find the segregating box is new and purely based on interval analysis.

4. Isolate a curve without critical points

Let $\mathcal{C}_f : f(x,y) = 0$ be a curve defined by a square-free polynomial $f(x,y)$ and $\mathbf{B} \in \square\mathbb{Q}^2$ such that $\mathbf{B}$ is segregating w.r.t $\mathcal{C}_f$ and $\mathbf{B}$ contains no critical points of $\mathcal{C}_f$. In this section, we will give two methods to compute isolating boxes for $\mathcal{C}_{\mathbf{B}}$.

4.1. A root isolation based method

Let $\mathbf{B} = [a,b] \times [c,d]$. We first construct a set of boxes covering $\mathcal{C}_{\mathbf{B}}$. We divide $[a,b]$ into n equal intervals with length smaller than a given precision ϵ. Let $n = \lfloor (b-a)/\epsilon \rfloor + 1$, $t = \frac{b-a}{n}$, and $t_i = a + it, i = 0, \ldots, n$. Since $\mathbf{B}$ contains no critical points of $\mathcal{C}_f$, for each $x_0 \in [a,b]$, by Lemma 3.1, line $x = x_0$ intersects with $\mathcal{C}_{\mathbf{B}}$ in a fixed number, say m, of points. Let $[c_{i,j}, d_{i,j}], j = 1, \ldots, m$ be the isolating intervals with length smaller than ϵ for the real roots of $f(t_i, y) = 0$. Assume that $c_{i,1} < c_{i,1} < \cdots < c_{i,m}$. Then we can construct the following boxes for $i = 0 \ldots, n-1, j = 1, \ldots, m$ (See Fig. 6(b)):

$$\mathbf{B}_{i,j} = [t_i, t_{i+1}] \times [\min\{c_{i,j}, c_{i+1,j}]\}, \max\{d_{i,j}, d_{i+1,j}]\}]. \tag{2}$$

It is clear that $\mathcal{C}_{\mathbf{B}} \subset \cup \mathbf{B}_{i,j}$. So what we need to do is to refine the boxes such that they become isolating. This procedure is given in the following algorithm.

Algorithm 4.1 IsoBoxR$(f(x,y), \mathbf{B}, \epsilon)$. *$\mathcal{C}_f : f(x,y) = 0$ is the curve, $\mathbf{B}$ is a box segregating w.r.t $\mathcal{C}_f$ and contains no critical points of $\mathcal{C}_f$, and $\epsilon > 0$. Output a set of ϵ-isolation boxes for $\mathcal{C}_{\mathbf{B}}$.*

1. Construct the initial boxes in (2). Let $\rho = \epsilon$.

2. If the following conditions are valid for all appropriate subscripts, output $\{\mathbf{B}_{i,j}\}$.

$$|\mathbf{B}_{i,j}| < \epsilon \tag{3}$$
$$\mathbf{B}_{i,j} \cap \mathbf{B}_{i,j+1} = \emptyset, \tag{4}$$
$$\mathbf{B}_{i,j} \cap \mathbf{B}_{i+1,j+1} = \mathbf{B}_{i,j} \cap \mathbf{B}_{i+1,j-1} = \emptyset. \tag{5}$$

   Conditions (4) and (5) guarantee that isolation boxes for different curve segments are disjoint.

3. Otherwise, let $\rho = \rho/2$, $\mathbf{B}_{i,j} = [a_i, b_i] \times [e_{i,j}, f_{i,j}]$ the box with largest size not satisfying one of the conditions, and $r = \frac{a_i + b_i}{2}$. Isolate the roots of $f(r, y) = 0$ with precision ρ; replace $\mathbf{B}_{i,j}.j = 1, \ldots, m$ with two sets of similarly constructed boxes over $[a_i, r]$ and $[r, b_i]$. Still use (2) to denote these boxes. Go to Step 2.

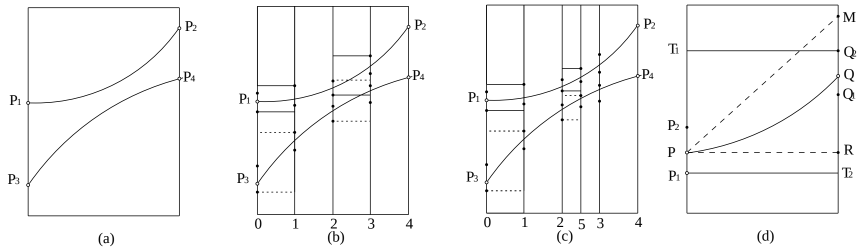


Fig. 6. Compute type one isolating boxes

Figure 6 is an illustration of the procedure. In Figure 6(b), $\mathbf{B}$ is divided into five smaller boxes. The solid points form the isolation intervals for the intersection points of $x = t_i$ and the curve. In the box between t_0 and t_1, condition (4) is satisfied. In the box between t_2 and t_3, condition (4) is not satisfied. We then add a new line $x = t_5$ and condition (4) is now satisfied for the boxes between t_2 and t_5. The isolation in Fig. **??**(d) is generated with this algorithm.

Proof of Correctness of Algorithm 4.1. We need only to show that conditions (3), (4), (5) will be satisfied after sufficient subdivision. Note that the width for each box will be as small as possible after sufficient subdivision, say $|\mathbf{B}_i|_w < \rho$. Also, from Step 3, we know that the isolation intervals $[c_{i,j}, d_{i,j}]$ of the roots of y will also be as small as possible, say also less than ρ. Then, we will show that if ρ is small enough, conditions (3), (4), (5) will be satisfied.

Since $\mathcal{C}_f$ has no critical points in $\mathbf{B}$ and $\mathcal{C}_f \cap \mathbf{B}$ is a closed set, the slopes of all the tangent lines of $\mathcal{C}_f$ for points in $\mathcal{C}_\mathbf{B}$ have a maximal value $s_m \neq 0$. Let us consider a curve segment PQ as shown in Figure 6(d), where (P_1, P_2) and (Q_1, Q_2) are the isolation intervals for P and Q respectively, $\mathbf{S} = P_1T_1Q_2T_2$ is the isolating box, and PM is a line with slope s_m. Then the curve segment must be under line PM. As a consequence, the height of $\mathbf{S}$ must satisfy $|\mathbf{S}|_h < P_1P_2 + Q_1Q_2 + MR \leq 2\rho + s_m\rho = (s_m + 2)\rho$. So, if $\rho < \epsilon/(s_m + 2)$, we have $|\mathbf{S}| < \epsilon$ and condition (3) is satisfied.

Let $\mathcal{C}_i$ be the i-th curve segment of $\mathcal{C} \cap \mathbf{B}$ arranged bottom up and $\mathbf{B}_i$ the corresponding box. Since $\mathcal{C}_i$ are disjoint, we have $d_m = \min\{|y_i - y_j|, \forall(x, y_i) \in \mathcal{C}_i, (x, y_i) \in \mathcal{C}_j\} > 0$. If $\rho < \frac{d_m}{2(s_m+2)}$, then $|\mathbf{B}_i|_h < (s_m + 2)\rho < d_m/2$ and $\mathbf{B}_i \cap \mathbf{B}_{i+1} = \emptyset$ will be valid. Thus, condition (4) will be satisfied. Otherwise, the y-distance between $\mathcal{C}_i$ and $\mathcal{C}_{i+1}$ will be smaller than $|\mathbf{B}_i|_h + |\mathbf{B}_{i+1}|_h < d_m$, a contradiction to the definition of d_m. Condition (5) can be proved in a similar way. ▮

4.2. A marching cube based method

Since $\mathbf{B}$ contains no critical points of $\mathcal{C}_f$, we could use the marching cube methods of Snyder [21] or Pantinga-Vegter [18] to mesh $\mathcal{C}_\mathbf{B}$. But these methods cannot be used directly because they do not generate isolating boxes. That is, a box in their output could contain more than one curve segments (Fig. 7). In this section, we introduce a subdivision method to generate isolating boxes.

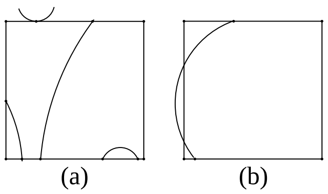


Fig. 7. (a): box generated by Snyder's method; (b) box generated by Pantinga-Vegter's method

For a box $\mathbf{S} = [a, b] \times [c, d]$, we consider the following **monotone condition** on the edges of $\mathbf{S}$:

$$0 \notin \square f_y(a, [c, d]), 0 \notin \square f_y(b, [c, d]), 0 \notin \square f_x([a, b], c), \text{ and } 0 \notin \square f_x([a, b], c). \tag{6}$$

If condition (6) is satisfied, then $f(a, y), f(b, y)$ are monotone functions in y and $f(x, c)$, $f(x, d)$ are monotone functions in x. Thus, we can easily check whether $\mathcal{C}_f$ intersects with the edges of $\mathbf{S}$. For instance, let $P = (a, c)$ and $Q = (a, d)$ be two vertices of $\mathbf{S}$. Then if $f(P)f(Q) > 0$, then $PQ \cap \mathcal{C}_f = \emptyset$; otherwise, $f(P)f(Q) \leq 0$ and $\mathcal{C}_f$ intersects with PQ. Furthermore, since $f(a, y)$ is monotone in y, $\mathcal{C}_f$ intersects with PQ at a single point. That is, we have

$$f(P)f(Q) \leq 0 \Leftrightarrow |PQ \cap \mathcal{C}_f| = 1. \tag{7}$$

To simplify the discussion, we assume that $\mathcal{C}_f$ does not pass through the vertices of $\mathbf{S}$. Since $\mathcal{C}_f$ has no critical points, in order for $\mathbf{S}$ to contain one curve segment of $\mathcal{C}_f$, we just need to require that $\mathbf{S}$ intersects $\mathcal{C}_f$ at two of its edges. We summarize these as the following lemma.

Lemma 4.2 *Let $\mathbf{S}$ be a box satisfying (6) and containing no critical points of $\mathcal{C}_f$. Then an edge of $\mathbf{S}$ either does not intersect with $\mathcal{C}_f$ or intersects with $\mathcal{C}_f$ at one point. Furthermore, if two edges of $\mathbf{S}$ intersect with $\mathcal{C}_f$, then $\mathbf{S}$ is a type one isolating box for $\mathcal{C}_f$.*

If $\mathcal{C}_f$ intersects with more than two edges of $\mathbf{S}$, we continue to subdivide $\mathbf{S}$. This leads to the following algorithm.

Algorithm 4.3 IsoBoxM$(f(x,y), \mathbf{B}, \epsilon)$. *$\mathcal{C}_f : f(x,y) = 0$ is a curve with no critical points in box $\mathbf{B}$. Output an ϵ-isolation for $\mathcal{C}_{\mathbf{B}}$.*

1. Let $\mathbf{BS} = \{\mathbf{B}\}$ and $\mathbf{IB} = \emptyset$. Repeat the following steps until $\mathbf{BS} = \emptyset$.
   (a) Let $\mathbf{S} \in \mathbf{BS}$ and $\mathbf{BS} = \mathbf{BS} \setminus \{\mathbf{S}\}$.
   (b) If $0 \notin \Box f(\mathbf{S})$, then $\mathcal{C}_f \cap \mathbf{S} = \emptyset$. We discard $\mathbf{S}$.
   (c) If condition (6) is satisfied, add S to $\mathbf{IB}$; otherwise subdivide $\mathbf{S}$ into four equal boxes and add them to $\mathbf{BS}$.
2. If $|\mathbf{S}| > \epsilon$ for $\mathbf{S} \in \mathbf{IB}$, subdivide $\mathbf{S}$ into four equal boxes and replace $\mathbf{S}$ with them in $\mathbf{IB}$.
3. Repeat the following steps until $\mathbf{IB} = \emptyset$ and output $\mathbf{BS}$..
   (a) Let $\mathbf{S} \in \mathbf{IB}$ and $\mathbf{IB} = \mathbf{IB} \setminus \{\mathbf{S}\}$.
   (b) If no edges of $\mathbf{S}$ intersect with $\mathcal{C}_f$, then $\mathcal{C}_f \cap \mathbf{S} = \emptyset$ and discard $\mathbf{S}$.
   (c) If two edges of $\mathbf{S}$ intersect with $\mathcal{C}_f$. Add $\mathbf{S}$ to $\mathbf{BS}$
   (d) Subdivide $\mathbf{S}$ into four equal boxes, add them to $\mathbf{IB}$.

The isolation boxes in Fig. 2(c) is generated with this algorithm.

Step 1 subdivides $\mathbf{B}$ into smaller boxes such that a box either does not intersection with $\mathcal{C}_f$ or condition (6) is valid. Since $\mathcal{C}_{\mathbf{B}}$ has no critical points, for a sufficiently small box containing $\mathcal{C}_f$, condition (6) will be satisfied. By the convergent property of the inclusion function, if a box $\mathbf{B}$ does not intersects with $\mathbf{B}$ then $0 \notin \Box f(\mathbf{B})$ will be satisfied after $\mathbf{B}$ is sufficiently subdivided. Thus, Step 1 will terminate.

Step 2 subdivides $\mathbf{B}$ into boxes whose sizes are smaller than ϵ.

Step 3 further subdivides $\mathbf{B}$ into isolating boxes for $\mathcal{C}_f$. Note that each box in $\mathbf{IB}$ satisfies condition (6). Then, we can use (7) to check whether an edge of $\mathbf{S}$ intersect with $\mathbf{S}$. The correctness of steps (b) and (c) is due to Lemmas 3.1 and 4.2 respectively. Since $\mathcal{C}_{\mathbf{S}}$ consists of disjoint curve segments, this step will terminate Also note that two boxes could share an edge not intersecting with $\mathcal{C}_f$. If this happens, we will shrink one of the boxes a little bit to make them separate. Some of the boxes in Fig. 2(c) are generated in this way.

5. The algorithm and experiments

Now, we can give the algorithm to compute an ϵ-isolation for a curve $\mathcal{C}_f$. The algorithm first constructs the isolating boxes for the curve near critical points with Algorithm 3.4 and then computes the isolating boxes for the curve in the rest of the regions with Algorithm 4.1 or 4.3.

Algorithm 5.1 IsoCur$(f(x,y),\mathbb{B},\epsilon)$. *$\mathcal{C}: f(x,y)=0$ is the curve, $\mathbb{B}$ is a box given in (1), and $\epsilon > 0$. Output an ϵ-isolation for $\mathcal{C}_{\mathbb{B}}$.*

1. Execute Algorithm 3.4 with input $f(x,y)$, $\mathbb{B}$, and ϵ to compute isolating boxes

$$\mathbf{BS} = \{\mathbf{B}_{i,j} = [a_i, b_i] \times [c_{i,j}, d_{i,j}], i = 1, \ldots, m, j = 1, \ldots, m_i\}.$$

   Note that all critical points of $\mathcal{C}_{\mathbb{B}}$ are contained in $\mathbf{B}_{i,j}$.

2. Let $b_0 = \mathcal{X}_1$, $a_{m+1} = \mathcal{X}_2$. Let $\mathbf{S}_i = [b_i, a_{i+1}] \times [\mathcal{Y}_1, \mathcal{Y}_2], i = 0, \ldots, m$. Note that $\mathcal{C}_f$ has no critical points in $\mathbf{S}_i$. Execute Algorithm 4.1 or 4.3 with input $f(x,y)$, $\mathbf{S}_i$, and ϵ to generate isolating boxes $\mathbf{BS}_i$

3. Output $\mathbf{BS} \cup \cup_{i=1}^{m+1} \mathbf{BS}_i$ or a linear meshing as described in Section 2.

In an isolation for a curve, two boxes overlap if and only if the curve pass through their common edge. In order to satisfy this condition, in Step 2, we need further assume that each box generated by Algorithm 4.1 or 4.3 meets at most one of the boxes $\mathbf{B}_{i,j}$ generated in Step 1. Otherwise, we need further subdivide the boxes in $\mathbf{BS}_i$. We do not add this to the description of the algorithms to keep them simple.

We will show several examples with our algorithm implemented with Maple. The timings are collected with a laptop with 2 core 1.73G CPU, 1 G memory, Windows Vista OS.

Example 1. Consider the curve in Fig. 1.

$$g = 2\,x^4 - 3\,x^2\,y + y^2 - 2\,y^3 + y^4.$$

The computing time TS for root isolation of triangular system is 0.064 seconds. The computing time for boxes approximation under the precision 0.2 with Algorithm 4.1 is 0.193 seconds. The computing time for boxes approximation under the precision 0.2 with Algorithm 4.3 is more than 1 hours!. Its approximating figures are as shown in Fig. 1 and Fig. **??**.

Remark: The reason why Algorithm 4.3 takes a long time is that there exist curve branches which are very close to each other even when they are disjoint. It will take the program a long time to separate the curve branches by repeated subdivision.

The following examples are implemented with Algorithm 4.1.

Example 2. Consider the approximation of the following curve. $g = 2 + 7\,x - 7\,y - 14\,x^3 + 7\,x^5 - x^7 - 16\,y^2 + 14\,y^3 + 20\,y^4 - 7\,y^5 - 8\,y^6 + y^7 + y^8 - 42\,y^2x - 70\,y^3x^2 + 35\,xy^4 + 70\,y^2x^3 + 42\,yx^2 - 35\,x^3y^4 + 7\,x^6y - 21\,x^5y^2 - 35\,yx^4 + 21\,x^2y^5 + 35\,y^3x^4 - 7\,xy^6$. The computing time for root isolation of triangular system is 1.592 seconds. The computing time for boxes approximation under the precision 0.05 is 2.418 seconds. The curve and its approximating figures are as shown in Fig. 8.

Example 3. Consider the approximation of the curve: $g = -3 + 12\,y^2 + 2\,y^4 - 12\,y^6 + y^8 + 12\,x^2 - 28\,y^2x^2 + 12\,y^4x^2 + 4\,y^6x^2 - 18\,x^4 + 20\,y^2x^4 + 2\,y^4x^4 + 12\,x^6 - 4\,x^6y^2 - 3\,x^8$. The computing time for root isolation of triangular system is 0.064 seconds. The computing time for boxes approximation under the precision 0.3 is 0.390 seconds. Its approximating figure is as shown in Fig. 9.

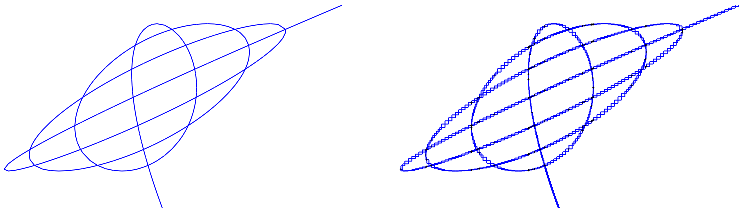


Fig. 8. Given curve and its boxes approximation

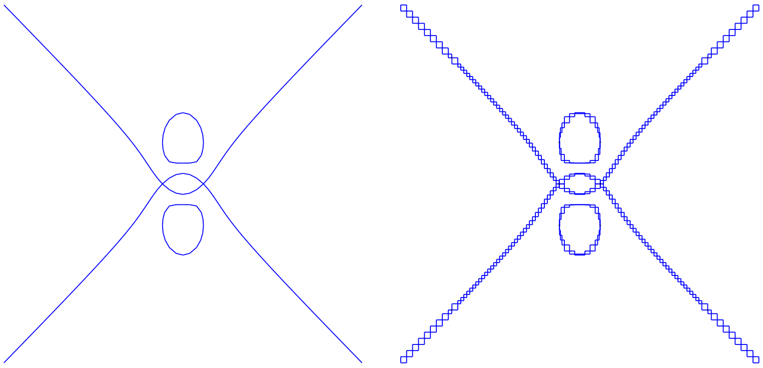


Fig. 9. Given curve and its boxes approximation

Example 4. Consider the approximation of the following curve.

$$g = x^8 + 4\,x^6y^2 + 6\,y^4x^4 + 4\,y^6x^2 + y^8 - 4\,x^6 - 12\,y^2x^4 - 12\,y^4x^2 - 4\,y^6 + 16\,y^2x^2.$$

The computing time for root isolation of triangular system is 0.020 seconds. The computing time for boxes approximation under the precision 0.07 is 0.218 seconds. The curve and its approximating figures are as shown in Fig. 10.

6. Conclusion

In this paper, we give an algorithm to generate an isolation for a plane curve, which is a set of boxes covering the curve, having the same topology as the curve, and approximating the curve to any given precision. A certified meshing can be generated from the isolation.

Key ingredients of the method include: a new method to generate isolating boxes and a new subdivision method to generate isolating boxes for curves without critical points. Both methods are totally based on interval arithmetics and are quite efficient.

References

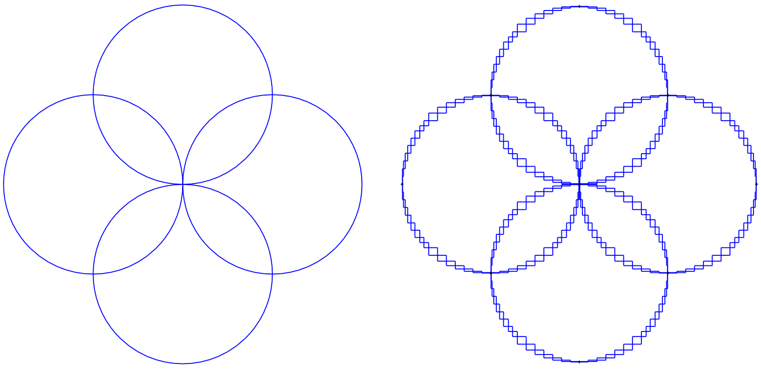


Fig. 10. Given curve and its boxes approximation

[1] L. Alberti and B. Mourrain. *Visualization of implicit algebraic curves*, Pacific Graphics, 303-312, 2007.

[2] S. Arnborg and H. Feng, Algebraic decomposition of regular curves. *J. Symbolic Comput.*, 5(1, 2), 131-140, 1988.

[3] 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, 1984.

[4] D.S. Arnon and S. McCallum, *A polynomial time algorithm for the topological type of a real algebraic curve*, *J. Symbolic Comput.*, 5, 213-236, 1988.

[5] C.W. Brown, *Constructing cylindrical algebraic decompositions of the plane quickly*, http://www.cs.usna.edu/ wcbrown/, 2002.

[6] M. Burr, S.W. Choi, B. Galehouse, and C.K. Yap, Complete subdivision algorithms, II: isotopic meshing of singular algebraic cuvres, *Proc. ISSAC 2008*, 87-94, ACM Press, New York, 2008.

[7] J.S. Cheng, X.S. Gao, and C.K. Yap, Complete numerical isolation of real roots in zero-dimensional triangular systems, *Proc. ISSAC 2007*, 92-99, ACM Press, New York, 2007.

[8] J.S. Cheng, S. Lazard, L. Penaranda, M. Pouget, F. Rouillier and E. Tsigaridas, On the topology of Planar Algebraic Curves, *Proc. EuroCG'08, Nancy-March 18-20*, 213-216, Nancy, 2008.

[9] K. Dey, K. Mehlhorn, and E.A. Ramos, *Curve reconstruction: connecting dots with good reason*, *Comput. Geom. Theory Appl.*, 15, 229-244, 2000.

[10] A. Eigenwillig, M. Kerber, and N. Wolpert, Fast and exact geometric analysis of real algebraic plane curves. *Proc. of ISSAC2007*, 151-158, ACM Pess, 2007.

[11] A. Eigenwillig, L. Kettner, E. Schönmer, and N. Wolpert, Complete, exact, and efficient computations with cubic curves. *Proc. of 20th Annu. Symp. Comput. Geom.*, 409-418, 2004.

[12] X.S. Gao and M. Li, Rational quadratic approximation to real algebraic curves, *Computer Aided Geometric Design*, 21, 805-828, 2004.

[13] L. Gonzlez-Vega and M.E. Kahoui, An improved upper complexity bound for the topology computation of a real algebraic plane curve, *Jornal of Complexity*, 12, 527-544, 1996.

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

[15] H. Hong, An efficient method for analyzing the topology of plane real algebraic curves, *Mathematics and Computers in Simulation,* 42(4-6), 571-582, 1996.

[16] R. Martin, S. Shou, I. Voiculscu, A. Bowyer, G. Wang, Comparison of interval methods for plotting algebraic curves, *Computer Aided Geometric Design*, 19, 553-587, 2002.

[17] M. Kerber, *Analysis of real algebraic plane curves*, Master's thesis, Universität des Saarlandes, Saarbrücken, Germany, 2006.

[18] S. Plantinga and G. Vegter, Isotopic meshing of implicit surfaces, *The Visual Computer*, 23, 45-58, 2007.

[19] H. Ratschek and J. Rokne, Scci-hybrid methods for 2d curve tracing, *Int. J.of Image and Graphics*, 5(3), 447-479, 2005.

[20] R. Seidel and N. Wolpert, On the exact computation of the topology of real algebraic curves, *SCG2005*, 107-116, 2005.

[21] J.M. Snyder, Interval analysis for computer graphics, *Proc. 19th Annual Conf. on Computers*, 26(2), 121-130, 1992.

[22] N. Wolpert, Jacobi curves: computing the exact topology of arrangments of non-sinular algebraic curves. *Proc. of ESA 2003*, 532-543, LNCS 2832, Springer, 2003.