Available online at www.sciencedirect.com

SCIENCE DIRECT

Computer Aided Geometric Design 21 (2004) 805–828

COMPUTER AIDED GEOMETRIC DESIGN

www.elsevier.com/locate/cagd

Rational quadratic approximation to real algebraic curves ☆

Xiao-Shan Gao *, Ming Li

Key Lab of Mathematics Mechanization, Institute of Systems Science, AMSS, Academia Sinica, Beijing 100080, China

Available online 18 August 2004

Abstract

An algorithm is proposed to give a global approximation of an implicit real plane algebraic curve with rational quadratic B-spline curves. The algorithm consists of four steps: topology determination, curve segmentation, segment approximation and curve tracing. Due to the detailed geometric analysis, high accuracy of approximation may be achieved with a small number of quadratic segments. The final approximation keeps many important geometric features of the original curve such as the topology, convexity and sharp points. Our method is implemented and experiments show that it may achieve better approximation bound with less segments than previously known methods. We also extend the method to approximate spatial algebraic curves.

Keywords: Real algebraic curve; Parametrization; Approximation; Topology determination

1. Introduction

An implicit real plane algebraic curve $\mathcal{C}$ of degree n is defined by $f(x, y) = 0$ where $f(x, y) \in \mathbb{R}[x, y]$ is a polynomial of degree n and $\mathbb{R}$ the field of real numbers. The curve is said to be *rational* if it can be additionally represented by rational parametric equations $x = \frac{x(t)}{d(t)}$ and $y = \frac{y(t)}{d(t)}$, where $x(t), y(t), d(t) \in \mathbb{R}[t]$ are of degrees at most n. Both the implicit and parametric representations of algebraic curves have important applications in CAGD. We can always convert a rational curve into an implicit representation,

☆ Partially supported by a NKBR Project of China and US NSF grant CCR-0201253.
* Corresponding author.
E-mail addresses: xgao@mmrc.iss.ac.cn (X.-S. Gao), mli@mmrc.iss.ac.cn (M. Li).

0167-8396/$ – see front matter
doi:10.1016/j.cagd.2004.07.009

which is called implicitization (Sederberg and Zheng, 2002). On the other hand, it is also desirable to generate a parametric representation for an implicitly defined algebraic curve, which is called parametrization. Methods to find parametric equations of implicit curves are given (Abhyankar and Bajaj, 1988; Sendra and Winkler, 1991; Gao and Chou, 1992). However only a small subset of real algebraic curves are rational. In general, an algebraic curve of arbitrary degree is rational if and only if its genus is equal to zero (Walker, 1978).

Approximation methods are therefore proposed to give a rational form for an implicit real algebraic curve. The methods can be categorized into three classes: the linear approximations, the points sampling approximations and the approximations based on power series. Ihm and Naylor surveyed some techniques for generating a linear approximation of an algebraic curve (Ihm and Naylor, 1991). Farouki proposed a segmentation method for algebraic curves and then used a polygon to approximate the curve (Farouki, 1989). However, the detail of the segmentation process is not presented in the paper. Given a model shape, curve or surface, expressed by a set of sample points on it, Pottmann et al. introduced the active B-spline curve or surface to approximate it (Pottmann et al., 2002). The method is further refined in (Yang et al., 2004). Based on the Implicit Function Theorem, Montaudouin et al. sought to represent a curve branch explicitly in one coordinate as function of the other one (Montaudouin et al., 1986). A technique was presented by Sederberg et al. to give a rational approximation of algebraic curves for some special cases (Sederberg et al., 1989). Using a combination of algebraic and numerical techniques, Bajaj and Xu constructed a C^1-continuous, piecewise rational approximation of a general plane algebraic curve (Bajaj and Xu, 1997). Interval cubic Bézier curves are used to approximate a plane algebraic curve (Chen and Deng, 2003). However, most of these methods rely on the local properties of the approximated curves without the consideration of their global properties, so they generally result in many pieces in the final approximations.

In this paper, we consider the rational quadratic approximation problem for a plane algebraic curve $\mathcal{C}$ with a global topology analysis. The resulted approximations are several rational quadratic B-spline curves, each of which is obtained from piecewise rational quadratic Bézier curves. The quadratic segment (or conics) is used since it has both the implicit and parametric forms and it is the freeform curve with the lowest degree and has many nice properties (Lee, 1985; Farin, 1989). The approximation algorithm mainly consists of the following steps:

(1) Topology determination. We find a rectangular bounding box $\mathcal{B}$ and a graph $\mathcal{G}$ such that the curve $\mathcal{C}$, $\mathcal{C}_{\mathcal{B}}$ (the part of $\mathcal{C}$ inside $\mathcal{B}$), and $\mathcal{G}$ have the same topology.
(2) Curve segmentation. Divide $\mathcal{C}$ into *triangle convex segments*, which have similar properties with conics.
(3) Segment approximation. We present a *shoulder point approximation* method to give a nice approximation to a triangle convex segment with conics expressed in a rational quadratic Bézier form.
(4) Curve tracing. Find a proper tracing order and convert the resulted approximation conics into rational quadratic B-spline curves, each of which gives a C^1 global parametrization for a curve branch.

Due to the detailed geometric analysis, high accuracy of approximation may be achieved with a small number of conics. The final approximation keeps many important geometric features of the original curve such as the topology, convexity and sharp points. The branches obtained in the tracing step provide a global parametrization and a refined topological structure of the curve. We implement our method in Maple and experiments show that our method may achieve better approximation bound with less segments than previously known methods.

We also extend our method to approximate a spatial algebraic curve $\mathcal{C}_S$ implicitly defined by the intersection of two algebraic surfaces with rational quadratic spline curves. The basic idea is that by performing a proper rotational transformation, the spatial curve is birational to a plane curve $C : R(x, y) = 0$ and the z-coordinate can be expressed as a rational function of x and y with $z = H(x, y)$. With this formula, the approximation of spatial curves is converted into approximation of plane curves.

The rest of the paper is organized as follows. The three main parts: topology determination and curve segmentation, segment approximation, curve tracing are illustrated in Sections 2, 3 and 4 respectively. The main algorithm and some experimental results for plane curve approximation are given in Section 5. The spatial case is illustrated in Section 6. We conclude this paper in Section 7.

2. Topology determination and curve segmentation

Throughout this paper, we assume that $f(x, y) \in \mathbb{Z}[x, y]$ is an irreducible polynomial of degree greater than two, where $\mathbb{Z}$ is the ring of integers. A plane algebraic curve $\mathcal{C}$ is implicitly defined by $f(x, y) = 0$. Let $\mathcal{B} = \{(x, y)\colon\ x_l \leqslant x \leqslant x_r,\ y_b \leqslant y \leqslant y_u\}$ be a bounding box. We use $\mathcal{C}_\mathcal{B}$ to denote the part of $\mathcal{C}$ inside $\mathcal{B}$. In this section, we will determine a bounding box $\mathcal{B}$ and a graph $\mathcal{G}$ such that $\mathcal{C}$, $\mathcal{C}_\mathcal{B}$ and $\mathcal{G}$ have the same topology. In the later sections, we will approximate $\mathcal{C}_\mathcal{B}$ instead of $\mathcal{C}$.

2.1. Preliminaries

A point $P = (x_0, y_0)$ is said to be a *singular point* on $\mathcal{C}$ if $f(x_0, y_0) = f_x(x_0, y_0) = f_y(x_0, y_0) = 0$. The *inflection points* or *flexes* of $\mathcal{C}$ are its non-singular points satisfying its Hession equation $H(f) = 0$ (Walker, 1978).

A *curve segment* S of $\mathcal{C}$ is an open ended and continuous part of $\mathcal{C}$ with two endpoints P_0 and P_2. The left (right) endpoint is the one with smaller (larger) x coordinate. If P_0 and P_2 have the same x coordinate, then the left (right) endpoint is the one with smaller (larger) y coordinate.

Let P_0 be an endpoint of a curve segment S. Then a tangent direction T_0 of S at P_0 always exists (Walker, 1978). If P_0 is the left (right) endpoint, T_0 is called the *left* (*right*) *tangent direction* of S, denoted by T_- (T_+). The *left* (*right*) tangent line is the line going through the left (right) endpoint with left (right) tangent direction.

We use $S[P_0, P_2]$ to denote a curve segment of curve $\mathcal{C}$ with left endpoint P_0 and right endpoint P_2 and $S[P_0, T_0, P_2, T_2]$ is also used when the left and right tangent directions T_0 and T_2 are also prescribed.

A curve segment $S[P_0, T_0, P_2, T_2]$ is said to be *triangle convex* if either

(1) The left and right tangent lines of S meet at a point P_1 and the line segment P_0P_2 and S form a convex region inside the *control triangle* $\triangle P_0P_1P_2$ of S; or
(2) T_0 and T_2 are parallel and the line segment P_0P_2 and the curve segment S form a convex region.

Triangle convex segments have many similar properties with conics.

S_P is said to be a *shoulder point* on a triangle convex segment $S[P_0, P_2]$ if S_P has the maximal distance to the line P_0P_2.

Lemma 1. *The shoulder point for a triangle convex segment $S[P_0, P_2]$ is unique.*

Proof. Suppose that there are two shoulder points S_P^1 and S_P^2. Since $S[P_0, P_2]$ is triangle convex, the line segment $S_P^1 S_P^2$ should lie inside the region formed by line P_0P_2 and S. Since S_P^1 and S_P^2 have maximal distance to P_0P_2, the line segment $S_P^1 S_P^2$ must be coincident with S. This is impossible because $f(x, y)$ is irreducible and of degree greater than two. □

A *graph* $\mathcal{G}$ is an ordered triple $(V(\mathcal{G}), E(\mathcal{G}), \psi_{\mathcal{G}})$ consisting of a nonempty set $V(\mathcal{G})$ of *vertices*, a set $E(\mathcal{G})$, disjoint from $V(\mathcal{G})$, of *edges*, and an *incidence function* $\psi_{\mathcal{G}}$ that associates with each edge of $\mathcal{G}$ an unordered pair of (not necessarily distinct) vertices of $\mathcal{G}$. The *degree* of a vertex v in $\mathcal{G}$ is the number of edges of $\mathcal{G}$ incident with v. A vertex of odd degree is called *odd vertex*. We usually use $e = (u, v)$ to denote an edge in $\mathcal{G}$ with vertices u and v.

From a set of curve segments $\mathcal{S}_S$, we can generate a plane graph $\mathcal{G}_S$ with a map $\mathcal{U}: \mathcal{S}_S \to \mathcal{G}_S$ such that

(1) $\mathcal{U}$ sends the endpoints of the segments in $\mathcal{S}_S$ to the vertices in $V(\mathcal{G}_S)$, and
(2) there exists an edge between two vertices v_1, v_2 in $\mathcal{G}$ if and only if v_1, v_2 are the endpoints of a curve segment in $\mathcal{S}_S$.

It can be seen that $\mathcal{U}$ is a bijection map and $\mathcal{U}^{-1}$ is used to denote the reverse.

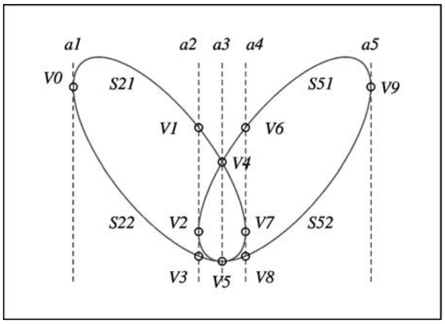
2.2. Topology determination

The topology determination of $\mathcal{C}$ produces a plane graph $\mathcal{G}$, which is topologically equivalent to $\mathcal{C}$ (Hong, 1996; Gonzalez-Vega and Necula, 2002). The algorithm in (Hong, 1996) is slightly modified to find a bounding box $\mathcal{B}$ such that $\mathcal{C}$ and $\mathcal{C}_{\mathcal{B}}$ have the same topology for later approximation.

Algorithm 1 (*Topology determination*). The input is a plane algebraic curve $\mathcal{C}$. The output is a bounding box $\mathcal{B} = \{(x, y)\colon\ x_l \leqslant x \leqslant x_r,\ \ y_b \leqslant y \leqslant y_u\}$ and a plane graph $\mathcal{G}_T$ such that $\mathcal{G}_T$, $\mathcal{C}_{\mathcal{B}}$, and $\mathcal{C}$ are topologically equivalent.

(1) Compute the discriminant $D(y) = \sum_{i=0}^{m} d_i y^i$ of $f(x, y)$ with respect to x and let $y_u = 1 + \frac{\max\{|d_0|,\ldots,|d_{m-1}|\}}{|d_m|}$. Then by Cauchy's inequality, all the roots of $D(y) = 0$ are in the interval $(y_b = -y_u, y_u)$.
(2) Compute the discriminant $\bar{D}(x)$ of $f(x, y)$ with respect to y and determine its real roots: $\alpha_1 < \cdots < \alpha_{s-1}$. Select two rational numbers x_l and x_r such that $x_l < \alpha_1$ and $x_r > \alpha_{s-1}$ and let $\alpha_0 = x_l$, $\alpha_s = x_r$. Now we have determined the bounding box $\mathcal{B}$.
(3) For every α_i, compute within $\mathcal{B}$ the real roots of $f(\alpha_i, y)$, $\beta_{i,0} < \cdots < \beta_{i,t_i}$.
(4) At each point $P_{i,j} = (\alpha_i, \beta_{i,j})$, count the numbers of branches of $\mathcal{C}_{\mathcal{B}}$ to the right and to the left.
(5) For each $0 \leqslant i < s$, the total number of branches to the right of points $P_{i,j}$ for all j must be the same as the total number of branches to the left of points $P_{i+1,k}$ for all k. Connect the points $P_{i,j}$ to the other endpoints P_{i+1,n_j} of the branches with edges, obeying the branch counts and get the graph $\mathcal{G}_T$.

Lemma 2. *$\mathcal{C}$ and $\mathcal{C}_B$ have the same topology.*



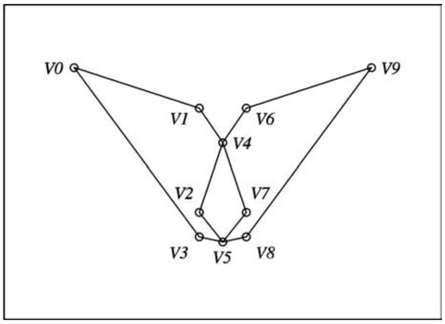


Fig. 1. A curve and its topology graph.

Proof. P is said to be a *vertical point* (*horizontal point*) of $\mathcal{C}$ if it is neither a singular point nor inflection point and $f(x_0, y_0) = 0$, $f_y(x_0, y_0) = 0$ ($f_x(x_0, y_0) = 0$). Vertical points are extremal points in the x-axis direction. If point (x_0, y_0) is a vertical point, then x_0 is a solution of the discriminant $\bar{D}(x) = 0$. Since all singular points, vertical points, and horizontal points are contained inside $\mathcal{B}$, the parts of $\mathcal{C}$ outside $\mathcal{B}$ are disjoint branches which have only one intersection with the boundary of $\mathcal{B}$. Hence $\mathcal{C}$ and $\mathcal{C}_{\mathcal{B}}$ have the same topology. □

Let $V = \{(\alpha_i, \beta_{i,j}),\ 0 \leqslant i \leqslant s,\ 0 \leqslant j \leqslant t_i\}$ and decompose it into $V = V_V \cup V_S \cup V_O$, where V_V is the set of vertical points, V_S is the set of singular points, and V_O is the other simple points. Label the generated curve segments in Algorithm 1 from left to right (i), top to bottom (j) as $\mathcal{S}_T = \{S_{i,j},\ 1 \leqslant i \leqslant s,\ 1 \leqslant j \leqslant s_i\}$ where s_i denotes the number of curve segments of $\mathcal{C}_{\mathcal{B}}$ with x in (α_{i-1}, α_i).

Consider the following curve for example

$$\mathcal{C}_0 : f_0(x, y) = 2x^4 - 3x^2 y + y^2 - 2y^3 + y^4 = 0. \tag{1}$$

The left part in Fig. 1 shows the corresponding symbols involved in the topology determination of $\mathcal{C}_0$ with Algorithm 1. The right figure shows the corresponding topology graph $\mathcal{G}_T$ of $\mathcal{C}_0$.

With the topology $\mathcal{G}_T$ for $\mathcal{C}_{\mathcal{B}}$, we can do the following basic operations for curve segments. Algorithm 2 try to obtain the intersection point of a vertical line with a specified segment, while Algorithm 3 determines which segment a specified point on $\mathcal{C}_{\mathcal{B}}$ is contained in.

Algorithm 2 (*Line curve intersection*). The inputs are a curve segment $S_{i_0,j_0} = S[P_0, P_2]$ in $\mathcal{S}_T$ for $\mathcal{C}_{\mathcal{B}}$ with $P_i = (x_i, y_i), i = 0, 2$ and an $\bar{x} \in (x_0, x_2)$. The output is the intersection point $\overline{P} = (\bar{x}, \bar{y})$ of the vertical line $x = \bar{x}$ with S.

(1) Let $g(y) = f(\bar{x}, y)$ and find all the solutions $y_1 > \cdots > y_s$ of $g(y) = 0$ within $\mathcal{B}$. Note that $g(y) = 0$ has no repeated roots.
(2) Since S_{i_0,j_0} is the j_0th segment of $\mathcal{C}_{\mathcal{B}}$ in the interval (x_0, x_2) from top to bottom, $\overline{P} = (\bar{x}, \bar{y}) = (\bar{x}, y_{j_0})$ should be on S_{i_0,j_0}.

Algorithm 3 (*Point containment*). The inputs are a point $\overline{P} = (\bar{x}, \bar{y})$ on $\mathcal{C}_{\mathcal{B}}$ and the segments set $\mathcal{S}_T$ of $\mathcal{C}_{\mathcal{B}}$. The output is a pair of footnotes (i_0, j_0) such that point $\overline{P}$ is on $S_{i_0,j_0} \in \mathcal{S}_T$.

(1) Select a unique α_{i_0} such that $\alpha_{i_0-1} \leqslant \bar{x} < \alpha_{i_0}$. If there exists only one segment S_{i_0,j_0} in the interval $(\alpha_{i_0-1}, \alpha_{i_0})$, output (i_0, j_0).
(2) If we can determine that each segment $S_{i_0,j}$ in the interval $(\alpha_{i_0-1}, \alpha_{i_0})$ is triangle convex and there exists just one segment S_{i_0,j_0} with $\overline{P}$ contained in its control triangle, output (i_0, j_0).
(3) Let $g(y) = f(\bar{x}, y)$. Isolate all the real roots of $g(y) = 0$ within $\mathcal{B}$ and get $y_1 > \cdots > y_r$. Suppose that $\bar{y}$ lies in the corresponding interval of y_{j_0} and output (i_0, j_0).

2.3. Flex computation and generation of triangle convex segments

We try to divide $\mathcal{C}_\mathcal{B}$ into triangle convex segments, so the division points must include flexes of $\mathcal{C}_\mathcal{B}$. A method to compute the real inflection points of cubic plane algebraic curves is given in (Chen and Wang, 2003). However there seems no work on computing the flexes of general implicit algebraic curves. Since this is not the central topics of this paper, we compute the flexes of $\mathcal{C}_\mathcal{B}$ directly from its definition by solving the equation system $f(x, y) = 0$ and $H(f) = 0$ with well known methods based on resultant computation. Let V_F be the set of the flexes on $\mathcal{C}_\mathcal{B}$.

Algorithm 4 (*Division at flexes*). The inputs are $\mathcal{S}_T$ and V_F. The output is a set of triangle convex segments $\mathcal{S}_F = \{S_{i,j,k},\ 1 \leqslant i \leqslant s,\ 1 \leqslant j \leqslant s_i,\ 1 \leqslant k \leqslant s_{ij}\}$ and its corresponding graph $\mathcal{G}_F$.

(1) For each $S_{i,j} \in \mathcal{S}_T$, find all the points in $V_F \cap S_{i,j}$ with Algorithm 3. List these points from left to right according to the x coordinate: $P_{i,j,k}$, $1 \leqslant k \leqslant s_{ij} - 1$.
(2) Divide the segment $S_{i,j}$ at the points $P_{i,j,k}$, $1 \leqslant k \leqslant s_{ij} - 1$, ending with the curve segments $S_{i,j,k}$, $1 \leqslant k \leqslant s_{ij}$.
(3) If there is no flex on $S_{i,j}$, let $S_{i,j,1} = S_{i,j}$ and $s_{ij} = 1$.
(4) Let $\mathcal{S}_F = \{S_{i,j,k},\ 1 \leqslant i \leqslant s,\ 1 \leqslant j \leqslant s_i,\ 1 \leqslant k \leqslant s_{ij}\}$ and $\mathcal{G}_F = \mathcal{U}(\mathcal{S}_F)$. It is clear that $\mathcal{S}_F$ takes $V = V_V \cup V_S \cup V_O \cup V_F$ as the endpoints of its segments.

Theorem 3. *Each curve segment $\mathcal{S}[P_0, P_2]$ in $\mathcal{S}_F$ is triangle convex.*

Proof. We may assume that $\mathcal{S}[P_0, P_2]$ is above the line P_0P_2. Let $P_i = (x_i, y_i),\ i = 0, 2$, and $P = (x, y)$ any point on S. Fig. 2 shows all the possible forms of S. Since there exist no singular points, flexes or vertical points on S and the sweeping angle of the tangent line from point P_0 to point P_2 is less than π, the slope $k(P)$ of S must be monotonic from P_0 to P_2 in this case. More precisely, it is decreasing with respect to the increasing of the x-coordinate of P. According to convex theory (Chang and Sederberg, 1997), a curve segment satisfying these conditions forms a convex region with P_0P_2. This proves the theorem for the case that the left and right tangent directions of S are parallel.

In the other cases, we need further to show that S is inside the control triangle $\triangle P_0P_1P_2$. For an arbitrary point $P = (x, y) \neq (x_0, y_0)$ on S, there must exists a point $\tilde{P}$ lying between P_0 and P with the

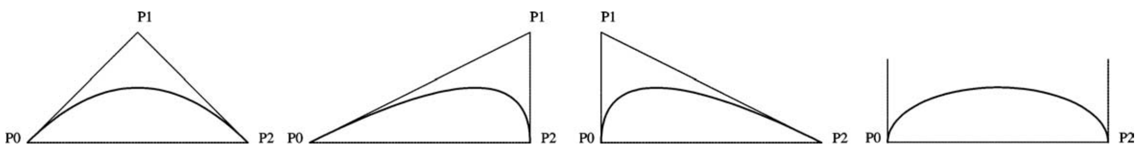


Fig. 2. Triangle convex segments.

maximal distance to P_0P. At this point, we have $k(\tilde{P}) = \frac{y-y_0}{x-x_0}$. On the other hand, there exists a point $(x, \bar{y})$ in the left tangent line of S such that $k(P_0) = \frac{\bar{y}-y_0}{x-x_0}$. Then

$$\frac{y-y_0}{x-x_0} = k(\tilde{P}) < k(P_0) = \frac{\bar{y}-y_0}{x-x_0}.$$

We have $y < \bar{y}$. Then the point $P = (x, y)$ lies below the point $(x, \bar{y})$, a point in the left tangent line of S. In a similar way, we have that all the points in S lie below the right tangent line of S. Hence S is inside the control triangle. □

The curve $\mathcal{C}_0$ in Fig. 1 does not have flexes. Then its topology graph need not to be modified. The curve in Fig. 14 has a flex point v_3.

2.4. Tangent direction computation

The tangent direction at a simple point can be easily obtained from its definition. In this section, we will give a method to compute the tangent directions at a singular point.

Let K be an algebraic closed field, and $\mathcal{C}$ a curve defined by $f(x, y) = 0$ over K. Suppose that all derivatives of $f(x, y)$, up to and including $(r-1)$th, vanish at P_0 but that at least one rth derivative does not vanish. The tangent directions (λ, μ) to $\mathcal{C}$ at P_0, correspond to the roots of

$$g(\lambda, \mu) = f_{x^r}\lambda^r + \binom{r}{1} f_{x^{r-1}y}\lambda^{r-1}\mu + \cdots + \binom{r}{r} f_{y^r}\mu^r = 0 \tag{2}$$

where all the partial derivatives are evaluated at P_0. But for a real algebraic curve defined by $f(x, y) \in \mathbb{R}[x, y]$, there does not exist such a one-one correspondence between the real roots of the equation $g(\lambda, \mu) = 0$ and the tangent directions of the real components containing P_0.

For example, let $\mathcal{C}$ be the plane curve defined by

$$f(x, y) = y^3 - 2y^2x + 15yx^4 - x^5.$$

The real roots of $g(\lambda, \mu) = 0$ at $P_0 = (0, 0)$ are $(1, 0)$ and $(1, 2)$. However, the curve has no real component with the tangent direction $(1, 0)$ at P_0 (Fig. 3).

In the following algorithm, we will propose a method to compute the set of tangent directions (λ, μ) of $\mathcal{C}$ at P_0, which is a subset of the set of real roots of $g(\lambda, \mu) = 0$.

Algorithm 5 (*Tangent directions at a singular point*). The input is a singular point $P_0 = (x_0, y_0)$ in V_S. The output is the left tangent directions T_{j-} for the segments $S_j \in \mathcal{S}_F$, $q \leqslant j \leqslant r$, with P_0 as its left endpoint. We assume that $S_1, \ldots, S_r$ are listed from top to bottom.

(1) Find all the solutions (λ_i, μ_i), $1 \leqslant i \leqslant s$, of the homogeneous algebraic equation $g(\lambda, \mu) = 0$ defined in (2). Let

$$k_i = \begin{cases} k_i = \frac{\mu_i}{\lambda_i}, & \lambda_i \neq 0; \\ +\infty, & \lambda_i = 0 \text{ and } \mu_i > 0; \\ -\infty, & \lambda_i = 0 \text{ and } \mu_i < 0. \end{cases}$$

(2) Sort k_i in a descending order and rename them as k_i, $1 \leqslant i \leqslant s$.

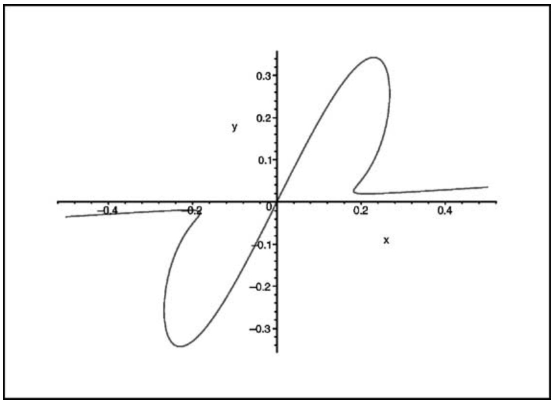


Fig. 3. Unexpected tangent direction.

(3) Let

$$\bar{k}_1 = \min(0, 2k_2), \quad I_1 = (-\infty, \bar{k}_1], \quad \text{when } k_1 = -\infty;$$
$$\bar{k}_s = \max(0, 2k_{s-1}), \quad I_s = [\bar{k}_s, +\infty), \quad \text{when } k_s = +\infty.$$
$$I_i = [k_i - \bar{\delta}, k_i + \bar{\delta}], \quad 1 \leqslant i \leqslant s, \quad \text{otherwise}.$$

Select a proper $\bar{\delta} < \delta$ such that $I_i \cap I_{i+1} = \phi$, $1 \leqslant i \leqslant s-1$.

(4) Take $\tilde{x}_j$ as the x-coordinate of the right endpoints of S_j, $1 \leqslant j \leqslant r$ and let

$$\tilde{x} = \min_{1 \leqslant j \leqslant r} \tilde{x}_j, \qquad \varepsilon = \frac{\tilde{x} - x_0}{100}.$$

(5) Find a point $(x_0 + \varepsilon, \bar{y_j})$ on S_j, $1 \leqslant j \leqslant r$ with Algorithm 2. Let $\bar{k}_j = \frac{\bar{y_j} - y_0}{\varepsilon}$, which is an approximation of the slope of S_j at P_0. If there exists a $\bar{k}_j$ which is not in $\cup I_i$, set $\varepsilon := \varepsilon/10$ and repeat this step. This step will end because $\bar{k}_j$ is approaching to the slope of some segment at point P_0.

(6) Suppose that $\bar{k}_j$ is in I_{n_j}, $1 \leqslant n_j \leqslant s$. Then the left tangent direction T_{j-} of S_j is $(\lambda_{n_j}, \mu_{n_j})$, $1 \leqslant j \leqslant r$.

We can compute the right tangent directions T_+'s in a similar way as Algorithm 5 by taking $-\varepsilon$ instead of ε. Add the tangent information to each segment in $\mathcal{S}_F$ to obtain $\mathcal{S}_{\vec{F}}$ and set $\mathcal{G}_{\vec{F}} = \mathcal{U}(\mathcal{S}_{\vec{F}})$ be the graph representation.

The tangent directions of the segments in Fig. 1 at the singular points are V_4: $(1, \sqrt{3})$, $(1, -\sqrt{3})$; V_5: $(1, 0)$.

2.5. Segments combination

Two methods based on graph disposal are proposed to combine some curve segments in $\mathcal{S}_{\vec{F}}$ under the condition that the triangle convexity of the segments is kept. The following algorithm considers the segments combination at simple points.

Algorithm 6 (*Segments combination-1*). The input is $\mathcal{G}_{\vec{F}}$. The output is a new graph $\mathcal{G}_{\vec{C}}$ topologically equivalent to $\mathcal{G}_{\vec{F}}$ and it has less edges than those of $\mathcal{G}_{\vec{F}}$.

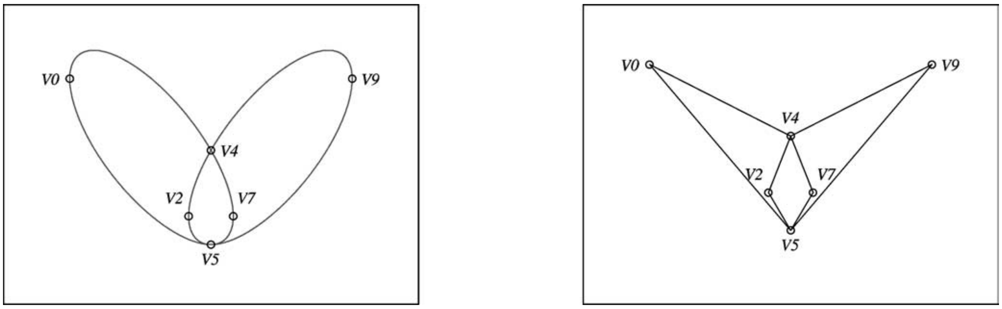


Fig. 4. Segments combination and the corresponding graph-1.

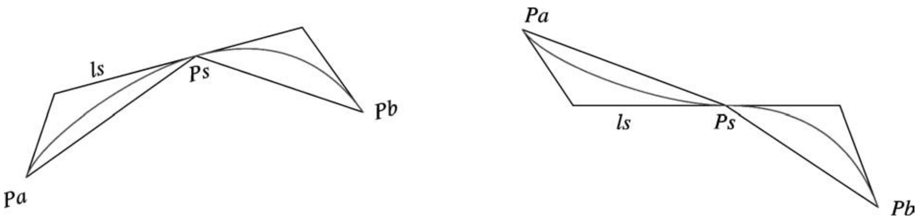


Fig. 5. Convexity maintenance.

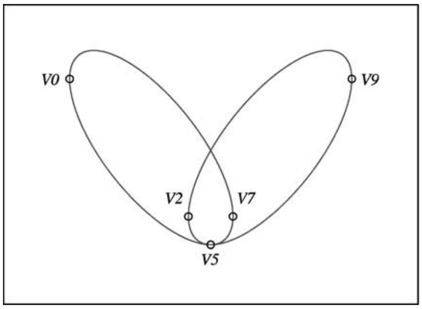
(1) Let the set of vertices of the graph $\mathcal{G}_{\vec{F}}$ be $V(\mathcal{S}_{\vec{F}}) = V_V \cup V_S \cup V_O \cup V_F$.
(2) For all $P_0 \in V_O$ and edges $E_{i_0,j_0,k_0} = (P_0, P_1)$, $E_{i_1,j_1,k_1} = (P_0, P_2)$, combine them as one edges E_J where $J = \{\{i_0, j_0, k_0\}, \{i_1, j_1, k_1\}\}$. The information in J is needed, e.g., in Algorithm 2.
(3) Keeping those edges containing no points in V_O unchanged, we obtain $\mathcal{G}_{\vec{C}}$. Let $\mathcal{S}_{\vec{C}} = \mathcal{U}^{-1}(\mathcal{G}_{\vec{C}})$. We have $V(\mathcal{G}_{\vec{C}}) = V_V \cup V_S \cup V_F$.

Since only two edges meet at a simple point, the graph topology does not change after removing simple points. The combined segments are still triangle convex because a combined segment always lies between two vertical lines. Fig. 4 shows the combined segments and its corresponding graph for those in Fig. 1.

The following algorithm tries to combine curve segments at certain singular points ensuring that the resulted approximation curves have the same topology with the original curve.

Algorithm 7 (*Segments combination-2*). The input is $\mathcal{G}_{\vec{F}}$. The output is a refined plane graph $\mathcal{G}_{\vec{C}}$ such that each segment in $\mathcal{S}_{\vec{C}} = \mathcal{U}^{-1}(\mathcal{G}_{\vec{C}})$ is triangle convex.

(1) Simplify $\mathcal{G}_{\vec{F}}$ to $\mathcal{G}_{\vec{C}}$ with Algorithm 6. Let $V_f \subset V_S$ be the set of singular points with degree four and not all the left or right tangent directions at the point are the same.
(2) For a point $P_S \in V_f$, let E_S be the set of edges in $\mathcal{G}_{\vec{C}}$ with P_S as an endpoint. If E_S is empty, go to step 5.
(3) Find two edges E_{i_0,j_0,k_0} and E_{i_1,j_1,k_1} in E_S such that they share the same tangent direction at P_S and they are at the same side of the tangent line l_s at P_S. In Fig. 5 the left case satisfies this condition while the right one does not.
(4) Combine the edges E_{i_0,j_0,k_0} and E_{i_1,j_1,k_1} into a new edge and refine $\mathcal{G}_{\vec{C}}$ as step 2 in Algorithm 6.
(5) Let $V_f = V_f \setminus \{P_S\}$ go to step 2 until V_f is empty. Set $\mathcal{S}_{\vec{C}} = \mathcal{U}^{-1}(\mathcal{G}_{\vec{C}})$.



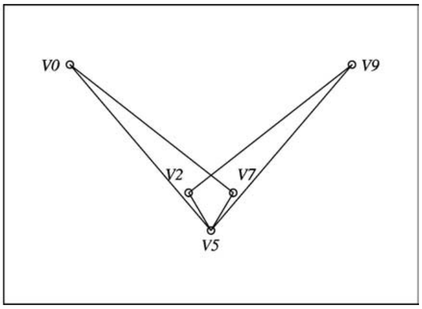


Fig. 6. Segments combination and the corresponding graph-2.

Fig. 6 shows the refined graph and curve segments for those in Fig. 1.

3. Segment approximation

In this section, the resulted segments from curve segmentation are to be approximated with rational quadratic Bézier curves. The approximation algorithm consists of two steps: the shoulder point computation and the segment approximation.

3.1. Rational quadratic Bézier curve

A *rational quadratic Bézier curve* has the following form

$$P(t)=\frac{P_0\phi_0(t)+\omega P_1\phi_1(t)+P_2\phi_2(t)}{\phi_0(t)+\omega\phi_1(t)+\phi_2(t)},\quad 0\leqslant t\leqslant 1, \tag{3}$$

where $\omega\in\mathbb{R}$, $P_i\in\mathbb{R}^2$ and $\phi_0=(1-t)^2$, $\phi_1=2t(1-t)$, $\phi_2=t^2$.

The rational quadratic Bézier curve (3) has the following properties (Lee, 1985; Farin, 1989; Pottmann, 1991).

(P1) $P(t)$ lies in its control triangle $\triangle P_0P_1P_2$ for $\omega>0$, and is triangle convex.
(P2) $P(t)$ passes through the endpoints $\triangle P_0$, P_2 with the corresponding tangent directions $P'(0)$ and $P'(1)$ parallel to P_0P_1 and P_1P_2.
(P3) If the tangent lines at the endpoints are parallel, the curve can be written as

$$P(t)=\frac{P_0\phi_0(t)+\omega T\phi_1(t)+P_2\phi_2(t)}{\phi_0(t)+\phi_2(t)};\quad 0\leqslant t\leqslant 1, \tag{4}$$

where T is the tangent vector at the endpoint P_0.
(P4) The point $S_P=P(\frac{1}{2})$ is called the *shoulder point* of $P(t)$. We have $S_P=\frac{1}{2}(Q_0+Q_2)$, where $Q_0=\frac{P_0+\omega P_1}{1+\omega}$, $Q_2=\frac{\omega P_1+P_2}{1+\omega}$ or $Q_0=P_0+\omega T$, $Q_2=\omega T+P_2$ when (4) is used. S_P is the unique point in the curve segment $P(t)$, $0\leqslant t\leqslant 1$, that has the maximum distance to line P_0P_2.

We usually rewrite $P(t)$ in (3) or (4) as $P(\omega,t)$ to show its dependence on ω. Let $S[P_0,T_0,P_2,T_2]$ be a triangle convex segment, and P_1 be the intersection point of the tangent lines at P_0 and P_2 if it

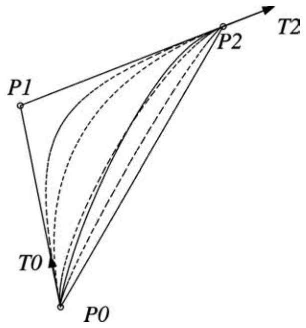


Fig. 7. Approximation curve family.

exists. The curve family $P(\omega,t)$ with $\omega>0$ interpolates points P_0, P_2 and has the tangent directions T_0, T_2 at P_0, P_2 respectively, and thus provides a G^1 approximation of S. Suppose that the solid curve in Fig. 7 is the curve segment S to be approximated and the dotted curves are the quadratic curve family $P(\omega,t)$. A proper value must be set for ω such that it has an optimal approximation to the segment S. The selection of the weight ω might lead to some optimization problems similar to the following:

$$\min_{\omega>0}\bigl(s\bigl(S, P(\omega,t)\bigr)\bigr), \quad \min_{\omega>0}\Bigl(\max_{0\leqslant t\leqslant 1}\bigl(d^2(\omega,t)\bigr)\Bigr),$$

where $s(S, P(\omega,t))$ is the area bounded by S and $P(\omega,t)$, $0\leqslant t\leqslant 1$ and $d(\omega,t)$ is some distance expression from a point $P(t)$ to S (Chuang and Hoffmann, 1989; Pottmann et al., 2002). However such expressions might involve complicated computations and are quite impractical.

In the next section, we will give another approximation method using the shoulder points. The shoulder points of S and $P(\omega,t)$ are to be pushed as near as possible, leading to an optimal approximation of the two segments in certain sense. This algorithm is therefore called *shoulder points approximation*. Experiments show that high accuracy of approximation may be achieved with a small number of conics.

3.2. Shoulder point computation

From the definition of the shoulder point, we can see that the gradient of $\mathcal{C}$ at the shoulder point S_P of $S[P_0, P_2]$, written as $\nabla f(S_P)$, is perpendicular to $P_2 - P_0$. The following equations system is therefore to be solved to get S_P.

$$F(x,y): \begin{cases} f(x,y)=0, \\ h(x,y)=\nabla f(x,y)\cdot(P_2-P_0)=0. \end{cases} \tag{5}$$

However, it is not trivial to determine which one of these solutions corresponds to the unique shoulder point of S. The following algorithm based on the Newton–Ralphson method provides an efficient method to obtain the shoulder point.

Algorithm 8 (*Shoulder point computation*). The input is a triangle convex segment $S[P_0, T_0, P_2, T_2]$. The output is the shoulder point S_P of S if it is found.

(1) Select an initial point I_0.

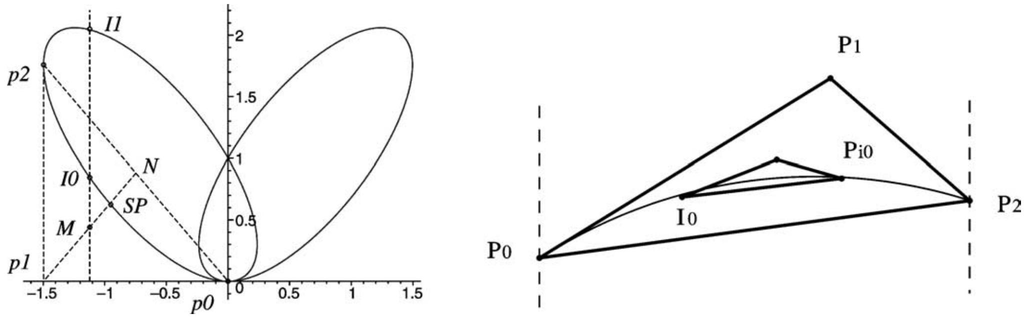


Fig. 8. Shoulder point computations.

As shown in the left part of Fig. 8, suppose $P_0P_1P_2$ is the control triangle of S and N is the midpoint of P_0, P_2. If T_0 is not parallel to T_2, let M be the midpoint of N and P_1; otherwise let $M = N$. Suppose $M = (M_x, M_y)$. Find the intersection point $I_0 = (M_x, I_y)$ on S from Algorithm 2 and set I_0 as the initial point.

It should be noticed that we do not take the initial point I_0 as the intersection point of the line P_1N with the segment S, which seems to be a better choice, since such an intersection point is not easy to be obtained.

(2) Find the shoulder point with the Newton–Ralphson method.

Starting at $p_0 = I_0$, repeat the following process until $\|\Delta p_k\| < \delta$.

- Let $J(x, y)$ be the Jacobian matrix of $F(x, y)$ defined in (5).
- Solve the system of the linear equations $J(p_k)\Delta p_k = -F(p_k)$.
- Let $p_{k+1} = p_k + \Delta p_k$. If p_{k+1} lies in the control triangle of S, go to the preceding step and repeat. Otherwise or if $k = 10$, the algorithm fails.

(3) If the above step ends in a successful way and p_{k+1} is neither a singular point nor an endpoint of S, output $S_P = p_{k+1}$. Otherwise, the algorithm fails.

In the left part of Fig. 8, I_0 is the initial point and S_P is the shoulder point computed with the algorithm.

The above algorithm can be used to compute the shoulder point in most cases. If it fails, e.g., when computed in the example curve $\mathcal{C}_5$ in Section 5, the following algorithm tries to refine the initial point of the Newton–Ralphson method until the shoulder point is obtained.

Algorithm 9 (*Refined shoulder point computation*). The input and output are the same with that of Algorithm 8 and suppose $P_i = (x_i, y_i)$, $i = 0, 2$.

(1) Use Algorithm 8 to find a shoulder point. If it fails, go to the next step.
(2) Let $I_0 = (I_x, I_y)$ be the initial point used in the preceding step, and $l(x, y) = 0$ the line passing through I_0 and parallel to P_0P_2 with the form $y = I_y + \frac{y_2-y_0}{x_2-x_0}(x - I_x)$ (right figure in Fig. 8). Substitute y into $f(x, y) = 0$ and we get a univariate equation $g(x) = 0$.

(3) Find all the solutions $\bar{x}_0 < \cdots < \bar{x}_m$ of $g(x) = 0$ in the interval (x_0, x_2). Let

$$\overline{P_i} = \left(\bar{x}_i, I_y + \frac{y_2 - y_0}{x_2 - x_0}(\bar{x}_i - I_x)\right), \quad 0 \leqslant i \leqslant m.$$

(4) Select a unique point $\overline{P_{i_0}}$, $0 \leqslant i_0 \leqslant m$, with Algorithm 3 such that it is on S. Set $P_0 = I_0$, $P_2 = \overline{P_{i_0}}$ and go to step 1.

Since the initial positions converge to the shoulder point, the process will end successfully.

3.3. *Segment approximation*

In geometry, the approximation error should be defined as the following Hausdorff distance between the segment S and its approximation S_a,

$$e(S, S_a) = \text{dis}(S, S_a) = \max_{P \in S} \min_{P' \in S_a} d(P, P').$$

However such a distance is difficult to compute and there is no need to compute it in most cases. As an implement, we take the distance from the parametric curve $P(t) = (x(t), y(t))$, $0 \leqslant t \leqslant 1$ to an implicit defined curve $\mathcal{C} : f(x, y) = 0$ in the following form, which is called the *error function* (Chuang and Hoffmann, 1989),

$$e(t) = \frac{f(x(t), y(t))}{[f_x(x(t), y(t))^2 + f_y(x(t), y(t))^2]^{1/2}}. \tag{6}$$

The *approximation error* between $P(t)$ and $\mathcal{C}$ is set as an optimization problem $e(P(t), \mathcal{C}) = \max_{0 \leqslant t \leqslant 1}(e(t))$. In practice, we sample t as $t_i = \frac{i}{n}$, $0 \leqslant i \leqslant n$, for a proper value of n and take the approximation error $e(P(t), \mathcal{C})$ as $\max_i(|e(t_i)|)$.

Algorithm 10 (*Segment approximation*). The inputs are a triangle convex curve segment $S[P_0, T_0, P_2, T_2]$ and the error bound δ. The output is a piecewise rational quadratic Bézier curves with G^1 continuity such that it give an approximation to S with approximation error less than δ.

(1) According to the interpolating requirements at the endpoints of $P(\omega, t)$, set $P(\omega, t)$ as (3), or (4) if T_0 and T_2 are parallel.

(2) Find the shoulder point $S_P = (P_x, P_y)$ on S with Algorithm 9.

(3) Let the shoulder point of $P(\omega, t)$ be $S(\omega)$. A specific value ω_0 will be determined such that $S(\omega_0)$ has a minimum distance to the shoulder point S_P. If T_0 is not parallel to T_2, suppose $P_i = (x_i, y_i)$, $i = 0, 1, 2$, then we have

$$S(\omega) = (S_x, S_y) = \frac{P_0 + 2\omega P_1 + P_2}{2(1+\omega)} = \left(\frac{x_0 + 2\omega x_1 + x_2}{2(1+\omega)}, \frac{y_0 + 2\omega y_1 + y_2}{2(1+\omega)}\right).$$

Solving the equation $\frac{\partial d^2(P, S(\omega))}{\partial \omega} = 0$, where $d^2(S_P, S(\omega)) = (P_x - S_x)^2 + (P_y - S_y)^2$, we get

$$\omega_0 = \frac{1}{2} \cdot \frac{(x_0 + x_2 - 2P_x) + \alpha(y_0 + y_2 - 2P_y)}{(P_x - x_1) + \alpha(P_y - y_1)},$$

where $\alpha = \frac{y_0 + y_2 - 2y_1}{x_0 + x_2 - 2x_1}$.

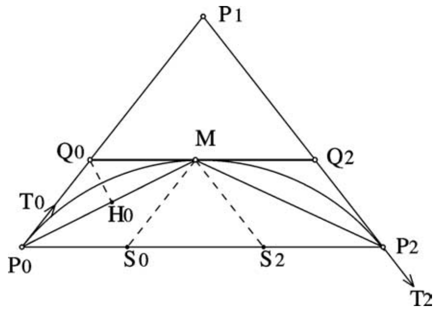


Fig. 9. Error control.

If $T_0 = (T_x, T_y)$ is parallel to T_2, we get in a similar way

$$\omega_0 = \frac{(2S_x - x_0 - x_2) + (2S_y - y_0 - y_2)}{2(T_x^2 + T_y^2)}.$$

(4) If the approximation error $e(P(\omega, t), S) < \delta$, output the Bézier curve. Otherwise, divide the segment into two parts at the shoulder point S_P and repeat the approximation method for them until the approximation error is less than δ.

We may assume that there always exists a control triangle for an approximated curve segment S, since if its tangent directions are parallel at the endpoints, we may do one step of subdivision at its shoulder point. We can give the following theorem.

Theorem 4. *With Algorithm* 10*, the approximation error is convergent to zero. More precisely, let s be the area of the control triangle for the approximated curve segment. After k steps of recursive subdivisions, the Hausdorff distance between S and its approximation is less than $\sqrt{s}/2^k$.*

Proof. Let $P(t)$ be the approximation curve of S after one step of approximation. Then $P(t)$ and S are contained in the same control triangle $\triangle P_0P_1P_2$ (Fig. 9). After one step of subdivision, the resulted segments are contained in triangles $\triangle P_0Q_0M$ and $\triangle P_2Q_2M$ respectively. Let S_0 and S_2 be points on P_0P_2 such that $MS_0 \parallel P_0Q_0$ and $MS_2 \parallel P_2Q_2$, s_0, s_1 and s_2 the areas of triangles $\triangle P_0Q_0M$, $\triangle S_0S_2M$ and $\triangle P_2Q_2M$ respectively. Then we have $(s_0 + s_2)/s_1 = Q_0Q_2/S_0S_2$ and $s_1/s = (S_0S_2/P_0P_2)^2$. As a consequence,

$$\frac{s_0 + s_2}{s} = \frac{Q_0Q_2 \cdot S_0S_2}{P_0P_2^2} \leqslant \frac{(Q_0Q_2 + S_0S_2)^2}{4P_0P_2^2} = \frac{1}{4}. \tag{7}$$

Due to the subdivision procedure, the angles $\angle P_1P_0P_2$ and $\angle P_1P_2P_0$ must be acute angles and hence the angles $\angle P_0Q_0M$ and $\angle P_2Q_2M$ must be obtuse angles. Let the altitudes of the triangles $\triangle P_0Q_0M$, $\triangle P_2Q_2M$ corresponding to P_0M, P_2M be $h_{10} = Q_0H_0$ and h_{12} respectively. We have $(Q_0H_0)^2 \leqslant P_0H_0 \cdot H_0M \leqslant (P_0H_0 + H_0M)^2/4 = (P_0M)^2/4$. That is $h_{10} \leqslant P_0M/2$. Acting in a similar way, we get $h_{12} \leqslant P_2M/2$. From (7), we have that

$$h_{10}^2 + h_{12}^2 \leqslant h_{10} \cdot \frac{P_0M}{2} + h_{12} \cdot \frac{P_2M}{2} = s_0 + s_2 \leqslant \frac{s}{4}.$$

In particular, we have $h_{10}^2 \leqslant s/4$ and $h_{12}^2 \leqslant s/4$. Repeat the process and it is easy to see that after k steps of subdivisions, we have $h_{k0}^2 \leqslant s/2^{2k}$ and $h_{k2}^2 \leqslant s/2^{2k}$. Thus $h_{k0} \leqslant \sqrt{s}/2^k$ and $h_{k2}^2 \leqslant \sqrt{s}/2^k$. $\square$

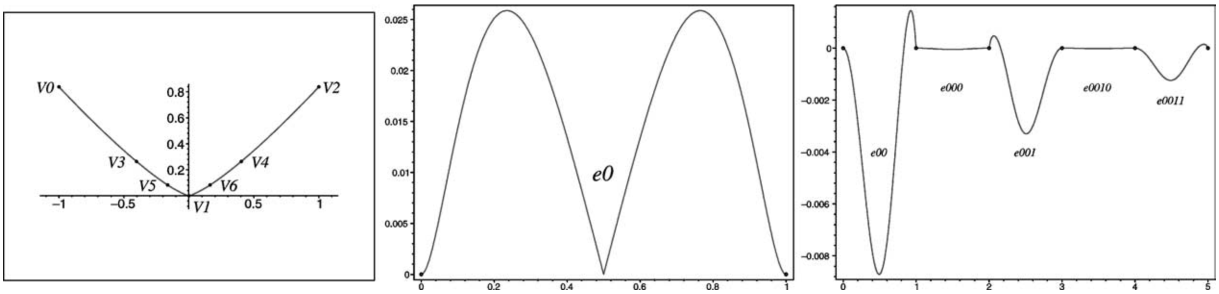


Fig. 10. Approximation and the error function of $\mathcal{C}_1$.

The first part of Fig. 10 shows the approximation of the following curve

$$\mathcal{C}_1 : (x^2 + y^2)^3 - 4x^2 y^2 = 0.$$

The approximation process is taken as follows. $\mathcal{C}_1$ is first approximated with one piece of conics with the error function $e_0(t)$ (6) plotted in the second part of Fig. 10. $\mathcal{C}_1$ is then divided at its shoulder point V_1 and the resulted two segments are approximated with error functions $e_{00}(t)$ and $e_{01}(t)$. Due to the symmetry of $\mathcal{C}_0$, we only show $e_{00}(t)$ in the third part of Fig. 10. In a similar way, $e_{000}(t)$, $e_{001}(t)$, $e_{0010}(t)$, $e_{0011}(t)$ are obtained and plotted in the third part of Fig. 10.

4. Curve tracing

The following concepts from graph theory will be used (Bondy and Murty, 1976). A *walk* in a graph $\mathcal{G}$ is a finite non-null sequence $W = v_0 e_1 v_1 e_2 v_2 \ldots e_k v_k$, whose terms are alternately vertices and edges, such that for $1 \leqslant i \leqslant k$, the ends of e_i are v_{i-1} and v_i. The integer k is the *length* of W. If the edges $e_1, e_2, \ldots, e_k$ are further distinct, W is called a *trail*. A trail that traverses every edge of $\mathcal{G}$ is called an *Euler trail* of $\mathcal{G}$. A walk is *closed* if it has positive length and its origin and terminus are the same. A closed Euler trail is called an *Euler tour*.

Theorem 5 (Bondy and Murty, 1976). *A connected graph has an Euler trail if and only if it has at most two vertices of odd degree.*

To guarantee the G^1 continuity, two segments $S[P_1, T_1, P_2, T_2]$ and $S[P_3, T_3, P_4, T_4]$ can be connected in a trail only if $P_2 = P_3$ and $T_2 = \lambda T_3$ for a non-zero number λ. A walk in the topology graph of a curve is called a *branch* if the corresponding segments of every neighboring edges in the trail satisfy the above property.

Based on the theory of plane algebraic curves, for example (Walker, 1978; Cheng et al., 2004), we first have the following theorems about the structure of a real plane algebraic curve $\mathcal{C}$.

Theorem 6. *Every point on a real plane algebraic curve $\mathcal{C}$ has an even number of segments originating from it. Furthermore, we can divide the segments into pairs such that two segments in the same pair share the same tangent direction.*

Theorem 7. *For a real plane algebraic curve $\mathcal{C}$, the number of the branches approaching to infinity is even.*

From Lemma 2 and the above two theorems, the vertices in the graph $\mathcal{G}_{\vec{C}}$ with odd degrees must be the intersections of the boundaries of $\mathcal{B}$ with the curve branches of $\mathcal{C}_{\mathcal{B}}$ approaching to infinity. Such points are called *boundary points* and its degree in $\mathcal{G}_{\vec{C}}$ must be one.

Algorithm 11 (*Curve tracing*). The input is a graph $\mathcal{G}_{\vec{C}}$ for a plane curve $\mathcal{C}_{\mathcal{B}}$. The outputs are edge-disjoint branches T_i such that $\bigcup_{i=1}^{r} T_i = E(\mathcal{G}_{\vec{C}})$.

(1) For all the singular points P in V_S do the following steps.
(2) For any edge $e_1 = (P, P_1)$, find another edge $e_2 = (P, P_2)$ sharing the same tangent direction at P with e_1. If there exist more than one such edges, we consider all the possible cases and select the one resulting the smallest number of branches r. This step is always possible by Theorem 6.
(3) Update $\mathcal{G}_{\vec{C}}$ as follows: (a) add a new vertex V_P and two new edges $e'_1 = (P_1, V_P)$, $e'_2 = (V_P, P_2)$ to $\mathcal{G}_{\vec{C}}$; (b) remove the edges e_1 and e_2 from $\mathcal{G}_{\vec{C}}$. Repeat this step until the degree of P is equal to two.
(4) Divide the updated graph $\mathcal{G}_{\vec{C}}$ into some connected subgraphs $\mathcal{G}^i_{\vec{C}}$, $1 \leqslant i \leqslant r$.
(5) The degree of each vertex in $\mathcal{G}^i_{\vec{C}}$ is two except the boundary points. The degree of a boundary point is one. With this property, we can then generate naturally a Euler trail T_i of $\mathcal{G}^i_{\vec{C}}$.

There only exist two possible cases for each $\mathcal{G}^i_{\vec{C}}$. One case is that $V(\mathcal{G}^i_{\vec{C}})$ contains no boundary points, then the resulted T_i is a Euler tour. The other case is that $V(\mathcal{G}^i_{\vec{C}})$ contains just two boundary points and T_i is then a Euler trail from one boundary point to the other one.

This algorithm not only gives a tracing order for $\mathcal{C}_{\mathcal{B}}$ but also gives a clear explanation for the number of the resulted branches. From Theorem 5, if the number of the boundary points on $\mathcal{C}_{\mathcal{B}}$ is $2k$, $k \geqslant 1$, there always exist k edge-disjoint walks T_i, $1 \leqslant i \leqslant k$, of $\mathcal{G}_{\vec{C}}$ such that $\bigcup_{i=1}^{k} T_i = E(\mathcal{G}_{\vec{C}})$. We can then conclude that $r \geqslant k$.

We give the generation process of the tracing order of $\mathcal{C}_0$ in Fig. 11. Vertices v_4 and v_5 are split into two points respectively, giving a Euler tour $v_0v_5v_9v'_4v_2v'_5v_7v_4v_0$ for the graph. Geometrically, the vertices v'_4 and v'_5 are v_4 and v_5 respectively.

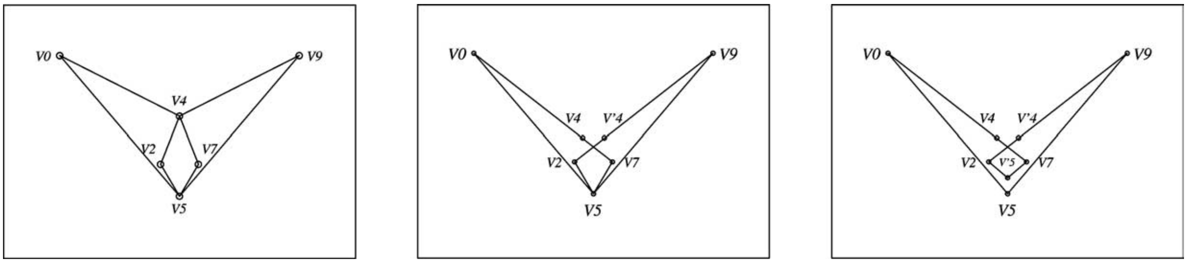


Fig. 11. The generation of the tracing order of $\mathcal{C}_0$.

5. Main algorithm and experimental results

Algorithm 12. The inputs are $\mathcal{C}: f(x, y) = 0$ and an error bound $\delta > 0$. The outputs are a bounding box $\mathcal{B}$ and rational quadratic B-spline curves $B_i(t)$, $1 \leqslant i \leqslant r$, such that they give a C^1 approximation to $\mathcal{C}_\mathcal{B}$ with $e(B_i(t), \mathcal{C}) < \delta$.

(1) *Topology determination.* Determine the bounding box $\mathcal{B}$ and the topology of $\mathcal{C}_\mathcal{B}$ with Algorithm 1. Let the resulted segments be $\mathcal{S}_T$ and let $\mathcal{G}_T = \mathcal{U}(\mathcal{S}_T)$.
(2) *Flex computation.* Compute the set of flexes V_F of $\mathcal{C}_\mathcal{B}$ as shown in Section 2.3. Divide those segments in $\mathcal{S}_T$ containing flexes with Algorithm 4 to obtain a new set of triangle convex segments $\mathcal{S}_F$ and let $\mathcal{G}_F = \mathcal{U}(\mathcal{S}_F)$.
(3) *Tangent computation.* Compute the tangent directions of each segment in $\mathcal{S}_F$ at its endpoints with Algorithm 5 to obtain $\mathcal{G}_{\vec{F}}$ and $\mathcal{S}_{\vec{F}}$.
(4) *Segments combination.* Combine some edges in the graph $\mathcal{G}_{\vec{F}}$ with Algorithm 6 or 7 to obtain a new graph $\mathcal{G}_{\vec{C}}$ and $\mathcal{S}_{\vec{C}}$.
(5) *Segment approximation.* Approximate each segment in $\mathcal{S}_{\vec{C}}$ with piecewise rational quadratic Bézier curves with Algorithm 10.
(6) *Curve tracing.* Find r edge-disjoint branches T_i, $1 \leqslant i \leqslant r$, in $\mathcal{G}_{\vec{C}}$ with Algorithm 11. Let $E_i = \mathcal{U}^{-1}(T_i)$ be the corresponding curve branches in $\mathcal{C}_\mathcal{B}$.
(7) *B-spline conversion.* Convert these approximation rational quadratic Bézier curves for the segments in E_i into a B-spline curve B_i with a proper knot selection (Piegl and Tiller, 1987). B_i provides a C^1 continuous approximation to branch B_i, $1 \leqslant i \leqslant r$.

The method reported is implemented in Maple. The benchmark curves $\mathcal{C}_0, \mathcal{C}_1, \mathcal{C}_2, \mathcal{C}_3, \mathcal{C}_4$ are from (Walker, 1978). Curves $\mathcal{C}_5$ and $\mathcal{C}_6$ are taken from (Gonzalez-Vega and Necula, 2002).

$$\begin{aligned}
&\mathcal{C}_2: x^4 + x^2y^2 - 2x^2y - xy^2 + y^2 = 0,\\
&\mathcal{C}_3: (x^2+y^2)^2 + 3x^2y - y^3,\\
&\mathcal{C}_4: (x^2+y^2)^3 - 4x^2y^2 = 0,\\
&\mathcal{C}_5: y^8 + y^7 - (8+7x)y^6 - (7-21x^2)y^5 - (-20-35x+35x^3)y^4\\
&\qquad - (-14+70x^2-35x^4)y^3 - (16+42x-70x^3+21x^5)y^2\\
&\qquad - (7-42x^2+35x^4-7x^6)y + 7x - 14x^3 + 7x^5 - x^7 = 0,\\
&\mathcal{C}_6: -3 + 12y^2 + 2y^4 - 12y^6 + y^8 + 12x^2 - 28y^2x^2 + 12y^4x^2\\
&\qquad + 4y^6x^2 - 18x^4 + 20y^2x^4 + 2y^4x^4 + 12x^6 - 4x^6y^2 - 3x^8 = 0.
\end{aligned}$$

In the figures in this section, the left figures show the approximation spline curves and the right figures show the plots of the corresponding error functions. Specifically, the curve in the right figures defined in the interval $(i, i+1)$ corresponds to the $(i+1)$th curve segment with the following tracing orders in the left figures.

Tracing order for $\mathcal{C}_0$ (Fig. 12): $v_0v_5v_9v_4v_2v_5v_7v_4v_0$.

Tracing order for $\mathcal{C}_2$ (Fig. 14): $v_0v_1v_2v_3v_4v_0$.

Tracing order for $\mathcal{C}_3$ (Fig. 15): $v_0v_1v_2v_3v_1v_4v_5v_1v_6v_0$.

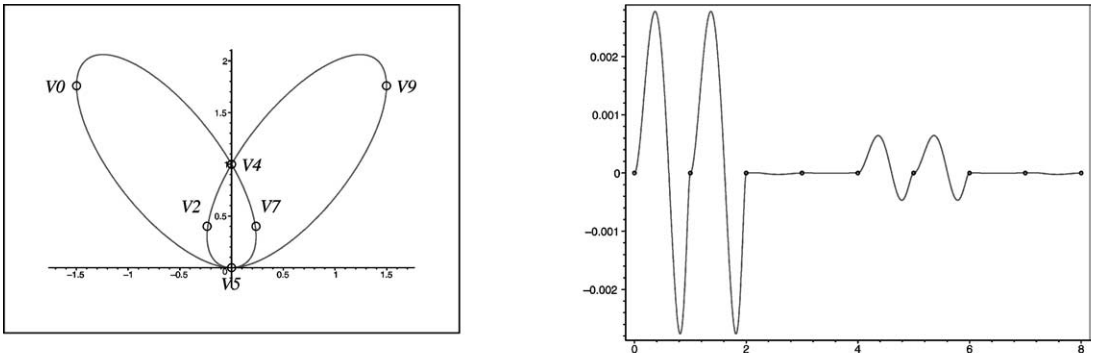


Fig. 12. Approximation and the error function of $\mathcal{C}_0$.

Fig. 13. Approximation of $\mathcal{C}_0$ in (Bajaj and Xu, 1997).

Table 1
Comparison of our results with results in BX (Bajaj and Xu, 1997)

		Our results			BX		
Ex.	deg	d-app	error	p-num	d-app	error	p-num
$\mathcal{C}_0$	4	(2,2)	0.003	8	(2,1)	0.1	34
$\mathcal{C}_2$	4	(2,2)	0.005	5	(3,3)	0.1	12
$\mathcal{C}_3$	4	(2,2)	0.005	9	(2,1)	0.09	27
$\mathcal{C}_4$	6	(2,2)	0.003	12	(2,1)	0.1	28

Tracing order for $\mathcal{C}_4$ (Fig. 16): $v_0v_1v_2v_3v_1v_4v_5v_1v_6v_0$.

Tracing order for $\mathcal{C}_5$ (Fig. 17): $v_0v_1v_2v_3v_4v_5v_6v_7v_8$.

Tracing order for $\mathcal{C}_6$ (Fig. 18): $v_0v_1v_2v_3v_4v_5v_6v_7v_8$ and $v_9v_{10}v_{13}v_{11}v_{12}v_9$.

As a comparison, we list some approximation results obtained by our algorithm and that obtained in (Bajaj and Xu, 1997) in Table 1. In the table, *deg* is the degree of curve $\mathcal{C}_i$; *d-app* $= (m, n)$, where m, n are the degrees of $P(t)$, $Q(t)$ in the approximation rational curve $\frac{P(t)}{Q(t)}$; *error* is the error bound; *p-num* is the number of approximation curve segments of degree p-app. It can be seen from the table that our new method may achieve better approximation bound with less approximation segments than that in (Bajaj and Xu, 1997). The approximation of $\mathcal{C}_0$ obtained in (Bajaj and Xu, 1997) are also shown in Fig. 13 for an intuitive comparison.

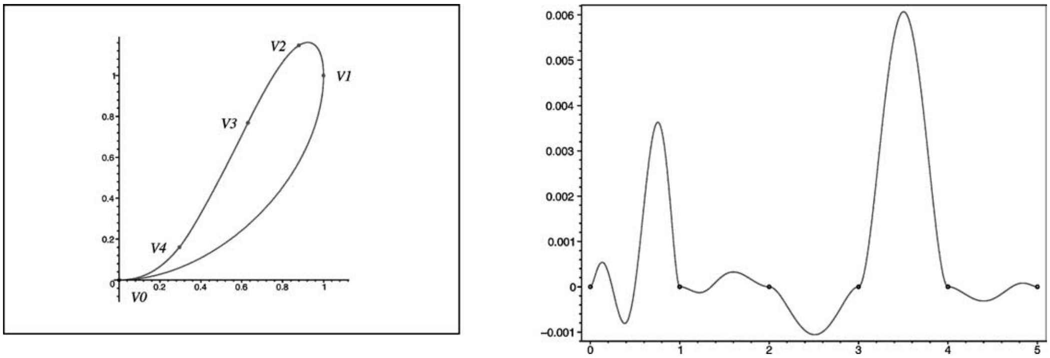


Fig. 14. Approximation and the error function of $\mathcal{C}_2$.

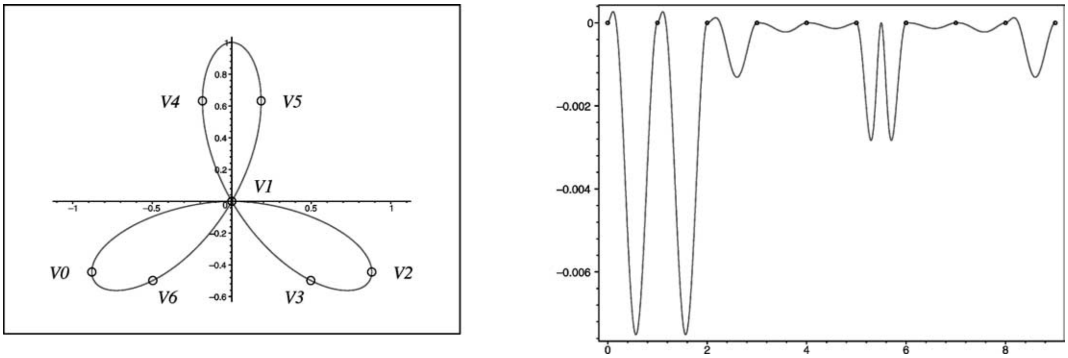


Fig. 15. Approximation and the error function of $\mathcal{C}_3$.

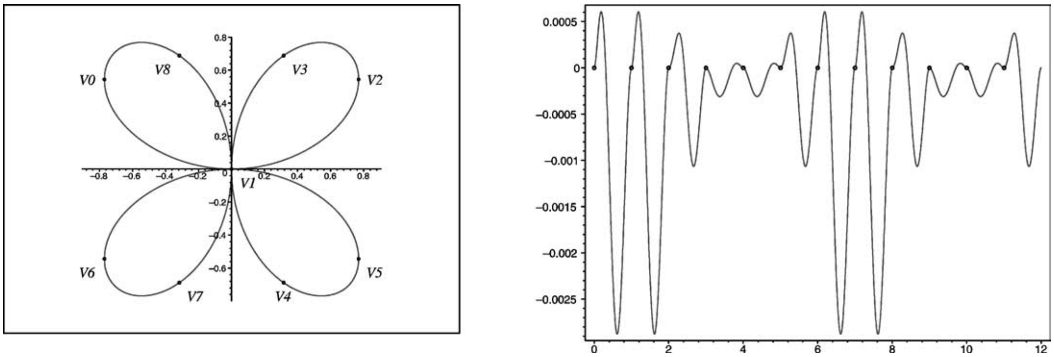


Fig. 16. Approximation and the error function of $\mathcal{C}_4$.

6. Approximation of spatial curves

Suppose that an irreducible spatial curve $\mathcal{C}_S$ is defined by the intersection of two implicitly defined algebraic surfaces $g(\mathbf{v}) = 0$ and $h(\mathbf{v}) = 0$, where $\mathbf{v} = (x, y, z)$. It is known how to decide whether the curve $\mathcal{C}_S$ is irreducible or not (Gao and Chou, 1992). Let A be a 3×3 matrix, $\bar{\mathbf{v}} = (\bar{x}, \bar{y}, \bar{z}) = \mathbf{v} \cdot A$, $\bar{g}(\bar{\mathbf{v}}) = g(\mathbf{v} \cdot A)$, $\bar{h}(\bar{\mathbf{v}}) = h(\mathbf{v} \cdot A)$. We first have the following result (Gao and Chou, 1992).

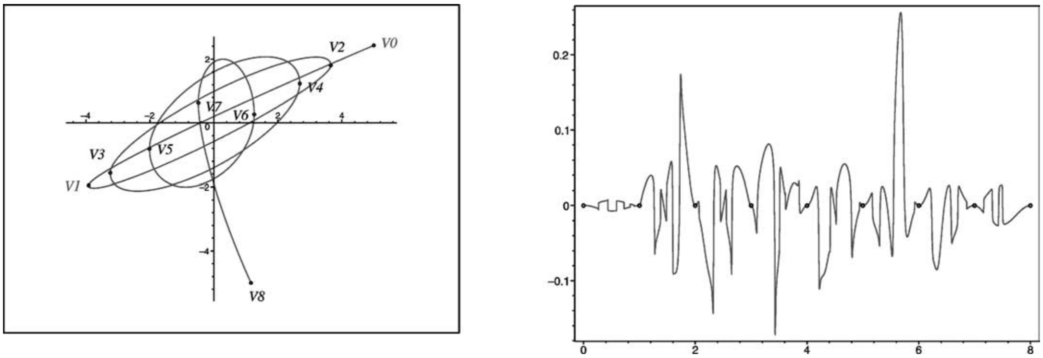


Fig. 17. Approximation and the error function of $\mathcal{C}_5$.

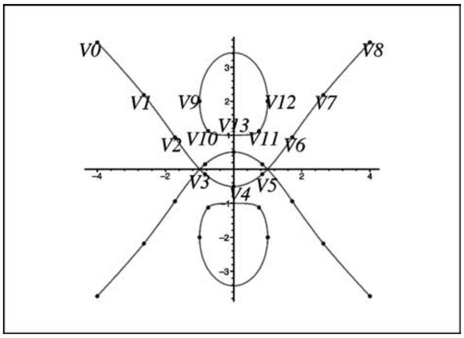


Fig. 18. Approximation of $\mathcal{C}_6$.

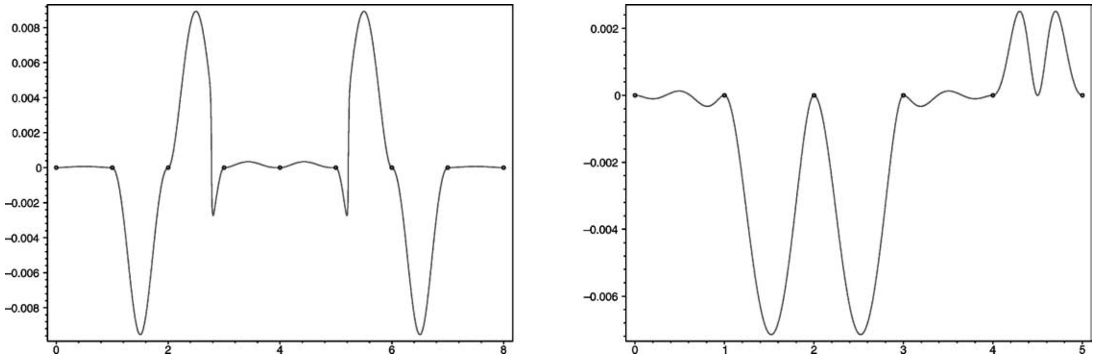


Fig. 19. The error function of $\mathcal{C}_6$. $\mathcal{C}_6$ has four branches and is symmetric with the x axis. We only show the result of the error function for two of the branches.

Theorem 8. *Let $\mathcal{C}_S$ be an irreducible spatial curve defined by $g(\mathbf{v}) = 0$ and $h(\mathbf{v}) = 0$. We may always find a rotational matrix A such that the new spatial curve $\bar{\mathcal{C}}_S$ defined by $\bar{g}(\bar{\mathbf{v}}) = \bar{h}(\bar{\mathbf{v}}) = 0$ is birational to an algebraic plane curve $\mathcal{C} : R(\bar{x}, \bar{y}) = 0$. We may also find a birational map from $\mathcal{C}$ to $\mathcal{C}_S$ as follows $(\bar{x}, \bar{y}) \to (\bar{x}, \bar{y}, H(\bar{x}, \bar{y}))$, where H is a rational function in $\bar{x}, \bar{y}$.*

Before giving our approximation method to $\mathcal{C}_S$, we will first propose an algorithm to give an approximation for a function $R(t)$ with a rational quadratic function.

Algorithm 13. The input is a rational function $R(t)$, $t_0 \leqslant t \leqslant t_1$, and the output is an approximation rational quadratic function $R_a(t)$, $t_0 \leqslant t \leqslant t_1$, of $R(t)$ such that

$$R_a(t_0) = R(t_0), \quad R_a(t_1) = R(t_1); \qquad R_a'(t_0) = R'(t_0), \quad R_a'(t_1) = R'(t_0).$$

(1) Without loss of generality, we may assume that $t_0 = 0$, $t_1 = 1$ for simplicity.
(2) Suppose that the approximation function of $R(t)$ is

$$R_a(t) = \frac{\omega_0 R_0 \phi_0(t) + \omega_1 R_1 \phi_1(t) + \omega_2 R_2 \phi_2(t)}{\omega_0 \phi_0(t) + \omega_1 \phi_1(t) + \omega_2 \phi_2(t)}; \quad 0 \leqslant t \leqslant 1,\ \omega_i, R_i \in \mathbb{R},\ i = 0, 1, 2,$$

where $\phi_i(t)$ is defined in (3).
(3) It can be easily seen that $R_0 = R(0)$ and $R_2 = R(1)$. From

$$R_a'(0) = \frac{2\omega_1}{\omega_0}(R_1 - R_0) = T_0, \quad R_a'(1) = \frac{2\omega_1}{\omega_2}(R_2 - R_1) = T_2,$$

we get

$$\begin{array}{ll} \omega_0 = \frac{2(R_1-R_0)}{\omega_1 T_0},\ \omega_2 = \frac{2(R_2-R_1)}{\omega_1 T_2}, & \text{if } T_0T_2 \neq 0; \\ \omega_0 = \frac{2(R_1-R_0)}{\omega_1 T_0},\ R_2 = R_1, & \text{if } T_0 \neq 0,\ T_2 = 0; \\ \omega_1 = 0, & \text{if } T_0 = T_2 = 0. \end{array}$$

(4) Let $R_a(\frac{1}{2}) = M = R(\frac{1}{2})$, we have

$$\begin{array}{ll} \omega_1 = 1,\ R_1 = \frac{R_0^2T_2 - R_2^2T_0 - (R_0T_2 - R_2T_0 - T_0T_2)M}{R_0T_2 - R_2T_0 + T_0T_2 + (T_0 - T_2)M}, & \text{if } T_0T_2 \neq 0; \\ \omega_1 = 1,\ \omega_2 = \frac{2(R_0^2 - R_0R_2 - R_2T_0 + (R_2 - R_0 + T_0)M)}{T_0(R_2 - M)}, & \text{if } T_0 \neq 0, T_2 = 0; \\ \omega_0 = 1,\ \omega_2 = -\frac{R_0 - M}{R_2 - M},\ R_1 = 1, & \text{if } T_0 = T_2 = 0. \end{array}$$

It is evident that the approximation error of $R_a(t)$ to $R(t)$ is convergent to zero.

For an approximation $\mathcal{C}_S^a(t)$ of $\mathcal{C}_S$, we define their approximation error function as

$$e(t) = \max\bigl(e(g,t), e(h,t)\bigr), \tag{8}$$

where

$$e(g,t) = \frac{g(\mathcal{C}_S^a(t))}{[g_x(\mathcal{C}_S^a(t))^2 + g_y(\mathcal{C}_S^a(t))^2 + g_z(\mathcal{C}_S^a(t))^2]^{1/2}}.$$

Function $e(h,t)$ is defined in a similar way as in (Chuang and Hoffmann, 1989). The approximation error $e(\mathcal{C}_S, \mathcal{C}_S^a)$ is then taken as $\max_{0 \leqslant t \leqslant 1} e(t)$. The following algorithm is taken to give a rational quadratic approximation of $\mathcal{C}_S$. It first gives an approximation to the projection of $\mathcal{C}_S$ into the xy plane with Algorithm 12. And then it approximates $\mathcal{C}_S$ in the z direction with Algorithm 13.

Algorithm 14. The inputs are a spatial curve $\mathcal{C}_S$ defined by $g = h = 0$ and an error bound $\delta > 0$. The outputs are a bounding box $\mathcal{B}_S = \{(x, y, z){:}\ x_0 \leqslant x \leqslant x_1,\ y_0 \leqslant y \leqslant y_1,\ z_0 \leqslant z \leqslant z_1\}$ and rational quadratic spatial spline curves $E_i(t)$, $1 \leqslant i \leqslant r$, such that they give a C^1 approximation of $\mathcal{C}_S$ within the bounding box $\mathcal{B}_S$ and $e(\mathcal{C}_S, E_i^S) < \delta$.

(1) Decide whether $g = h = 0$ defines an irreducible spatial curve as done in (Gao and Chou, 1992). If $\mathcal{C}_S$ is not irreducible, end the algorithm. Otherwise, let $R(x, y)$ and $H(x, y)$ be the polynomial and rational function obtained in Theorem 8.
(2) For the plane curve $\mathcal{C} : R(x, y) = 0$, determine a bounding box $\mathcal{B}_1 = \{(x, y)\colon x_0 \leqslant x \leqslant x_1,\ y_0 \leqslant y \leqslant y_1\}$ and its approximation B-spline curves $B_i(t)$, $0 \leqslant i \leqslant r$ with Algorithm 12. Similarly, we may determine a bounding box $\mathcal{B}_2 = \{(x, z)\colon \bar{x}_0 \leqslant x \leqslant \bar{x}_1,\ z_0 \leqslant y \leqslant z_1\}$. Let $\mathcal{B}_S = \{(x, y, z)\colon \min(x_0, \bar{x}_0) \leqslant x \leqslant \max(x_1, \bar{x}_1),\ y_0 \leqslant y \leqslant y_1,\ z_0 \leqslant y \leqslant z_1\}$. Then $\mathcal{C}_S$ and the part of $\mathcal{C}_S$ inside $\mathcal{B}_S$ have the same topology.
(3) Let $E_P(t) = (E_x(t), E_y(t))$, $t_0 \leqslant t \leqslant t_1$ be one quadratic segment in $B_i(t)$. Suppose S and $S_S[P_0, P_1]$ are the corresponding segments to $E_P(t)$ in $\mathcal{C}$ and $\mathcal{C}_S$ respectively. $S_S[P_0, P_1]$ is a spatial curve segment with endpoints $P_i = (x_i, y_i, z_i)$, $i = 0, 1$. Take the following steps to give a rational quadratic approximation of S_S.
    - Let $\bar{E}_z(t) = H(E_x(t), E_y(t))$, $t_0 \leqslant t \leqslant t_1$. From the interpolation property of $E_P(t)$ at its endpoints, we have $\bar{E}_z(t_i) = H(E_x(t_i), E_y(t_i)) = H(x_i, y_i) = z_i$, $i = 0, 1$.
    - Give a rational quadratic approximation function $E_z(t)$ for $\bar{E}_z(t)$ such that $E_z(t_i) = \bar{E}_z(t_i)$, $E'_z(t_i) = \bar{E}'_z(t_i)$, $i = 0, 1$, with Algorithm 13. Let $E(t) = (E_x(t), E_y(t), E_z(t))$, $t_0 \leqslant t \leqslant t_1$. Then it is a spatial quadratic curve segment.
    - If the approximation error $e(E(t), S_S) < \delta$, end this procedure. Otherwise compute the shoulder point $S_P = (x_p, y_p)$ of S and let $z_p = H(x_p, y_p)$. Divide S_S at (x_p, y_p, z_p) and repeat this procedure until $e(E(t), S_S) < \delta$.
(4) The resulted spatial quadratic curve segments naturally form a spline curve with C^1 continuity. Denote them as $E_i(t)$, $1 \leqslant i \leqslant r$.

Theorem 9. *With Algorithm* 14*, each resulted spline curve $E_i(t)$, $0 \leqslant i \leqslant r$, is C^1-continuous and the approximation $e(\mathcal{C}_S, E_i(t))$ is convergent to zero after a sufficient number of subdivisions.*

Proof. Suppose that $\bar{E}_i(t) = (x_i(t), y_i(t), z_i(t))$, $i = 0, 1$, are two adjacent quadratic segments (conics) in $E_i(t)$ sharing a common knot $t = t_1$. From the approximation of the plane curve $\mathcal{C}$, we have that $(x_0(t_1), y_0(t_1)) = (x_1(t_1), y_1(t_1))$ and $(x'_0(t_1), y'_0(t_1)) = (x'_1(t_1), y'_1(t_1))$. Then we get that $z_0(t_1) = H(x_0(t_1), y_0(t_1)) = H(x_1(t_1), y_1(t_1)) = z_1(t_1)$ and therefore $\bar{E}_0(t_1) = \bar{E}_1(t_1)$. Furthermore, from $z'_i(t) = H_x(x_i(t), y_i(t))x'_i(t) + H_y(x_i(t), y_i(t))y'_i(t)$, we get $z'_0(t_1) = z'_1(t_1)$ and therefore $\bar{E}'_0(t) = \bar{E}'_1(t)$. This proves that each $E_i(t)$ is C^1 continuous.

Suppose that $P_0 = (x_0, y_0, z_0)$ is a point on $\mathcal{C}_S$. Then we have $R(x_0, y_0) = 0$ and $z_0 = H(x_0, y_0)$. From the approximation of $E_P(t)$ to $\mathcal{C} : R(x, y) = 0$, we have a point $E_P(t_0)$ with the minimum Euclidean distance $D(E_P(t_0), (x_0, y_0))$ to (x_0, y_0). Let $H(t_0) = H(E_P(t_0))$, $\tilde{P}_0 = (E_P(t_0), H(t_0))$ and $\overline{P}_0 = (x_0, y_0, H(t_0))$ and we have

$$D\big(E(t_0), P_0\big) \leqslant D\big(E(t_0), \tilde{P}_0\big) + D(\tilde{P}_0, \overline{P}_0) + D(\overline{P}_0, P_0).$$

From the approximation of $E_z(t)$ to $H(t)$, $D(E(t_0), \tilde{P}_0)$ is convergent to zero. From the approximation of $E_P(t)$ to $\mathcal{C}$, $D(\tilde{P}_0, \overline{P}_0)$ converges to zero. With the continuity of $H(x, y)$, $D(\overline{P}_0, P_0) = D((x_0, y_0, H(E_P(t_0))), (x_0, y_0, H(x_0, y_0)))$ is convergent to zero. Then $D(E(t_0), P_0)$ is convergent to zero and so does $e(\mathcal{C}_S, E_i(t))$. $\square$

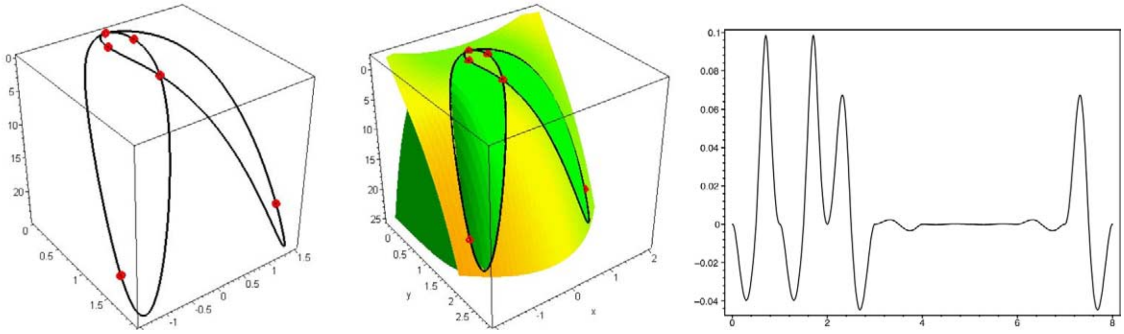


Fig. 20. Approximation of a spatial curve defined as the intersection of two surfaces.

Let $\mathcal{C}_S$ be a spatial curve defined as below (Bajaj et al., 1988) (Fig. 20)

$$g(x, y, z) = z - 2x^4 - y^4 = 0, \qquad h(x, y, z) = z - 3x^2y + y^2 - 2y^3 = 0,$$

and let $R(x, y)$ be the resultant of $g(x, y, z)$ and $h(x, y, z)$ with respect to z. Then the plane curve $\mathcal{C}: R(x, y) = 2x^4 - 3x^2y + y^2 - 2y^3 + y^4 = 0$ is birational to $\mathcal{C}_S$ with $H(x, y) = 2x^4 + y^4$. In fact $\mathcal{C}$ is just $\mathcal{C}_0$ as defined in (1). Using the approximation result of $\mathcal{C}_0$ in Section 5, we obtain the approximation curve for $\mathcal{C}_S$, which still consists of eight curve segments as plotted in the left figure in Fig. 20. The plot of the approximation error function as defined in (8) is shown as the right figure in Fig. 20.

7. Conclusion

In this paper, we give a simple and intuitive approximation of the plane algebraic curve with rational quadratic curves. The basic idea is to divide the curve into triangle convex segments which can be nicely approximated with quadratic Bézier curves and to connect the segments into certain maximal branches which can be globally approximated by quadratic B-splines. We also extend the method to give approximation to spatial curves.

Experiments show that we can achieve high precision approximation with few segments. Instead of giving a power series for each approximated segment, the endpoint information and shoulder points are mainly used to express the segment. Since the geometric information is considered, the algorithm is easy to understand and many geometric characters of the approximated curve are kept.

References

Abhyankar, S., Bajaj, C., 1988. Automatic parameterization of rational curves and surfaces, III: algebraic plane curves. Computer Aided Geometric Design 5 (4), 309–321.

Bajaj, C., Hoffmann, C., Hopcroft, J., Lynch, R., 1988. Tracing surface intersections. Computer Aided Geometric Design 5 (4), 285–307.

Bajaj, C., Xu, G.L., 1997. Piecewise rational approximations of real algebraic curves. J. Comput. Math. 15 (1), 55–71.

Bondy, J., Murty, U., 1976. Graph Theory with Applications. Macmillan Press.

Chang, G.Z., Sederberg, T., 1997. Over and Over Again. American Mathematical Association.

Chen, F.L., Wang, W.P., 2003. Computing real inflection points of cubic algebraic curves. Computer Aided Geometric Design 20 (2), 101–117.

Chen, F.L., Deng, L., 2003. Interval parametrization of planar algebraic curves. In: Li, Z.M., Sit, W. (Eds.), Proceeding of the 6th Asian Symposium on Computer Mathematics. World Scientific, Singapore, pp. 64–76.

Cheng, J., Gao, X.-S, Li, M., 2004. Topology determination of real projective plane algebraic curves. Preprint.

Chuang, J., Hoffmann, C., 1989. On local implicit approximation and its application. ACM Trans. Graph. 8 (4), 298–324.

Farin, G., 1989. Curvature continuity and offsets for piecewise conics. ACM Trans. Graph. 8 (2), 89–99.

Farouki, R., 1989. Hierarchical segmentations of algebraic curves and some applications. In: Lyche, T., Schumaker, L.L. (Eds.), Math. Methods in Comp. Aided Geom. Design. Academic Press, Boston, MA, pp. 239–248.

Gao, X.S., Chou, S.C., 1992. On the parameterization of algebraic curves. Applicable Algebra in Elementary Communication and Computing 3, 27–38.

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

Hong, H., 1996. An efficient method for analyzing the topology of plane real algebraic curves. Math. Comput. Simulation 42 (4–6), 571–582.

Ihm, I., Naylor, B., 1991. Piecewise linear approximations of digitized space curves with applications. In: Patrikalakis, N. (Ed.), Scientific Visualization of Physical Phenomena. Springer-Verlag, Berlin, pp. 545–569.

Lee, E., 1985. Rational Bézier representation for conics. In: Farin, G. (Ed.), Geometric Modeling: Algorithm and New Trends. SIAM, Philadelphia, PA, pp. 3–19.

Montaudouin, Y., Tiller, W., Vold, H., 1986. Application of power series in computational geometry. Computer-Aided Design 18 (10), 93–108.

Piegl, L., Tiller, W., 1987. Curve and surface constructions using rational B-spline. Computer-Aided Design 19 (9), 485–498.

Pottmann, H., 1991. Locally controllable conic splines with curvature continuity. ACM Trans. Graph. 10 (4), 366–377.

Pottmann, H., Leopoldseder, S., Hofer, M., 2002. Approximation with active B-spline curves and surfaces. In: Coquillart, S., Shum, H., Hu, S.M. (Eds.), Proc. of Pacific Graphics 2002. IEEE Press, Los Alamitos, CA, pp. 8–25.

Sederberg, T., Zhao, J., Zundel, A., 1989. Rational approximation of algebraic curves. In: Strasser, W., Seidel, H. (Eds.), Theory and Practice of Geometric Modeling. Springer-Verlag, Berlin, pp. 33–54.

Sederberg, T., Zheng, J., 2002. Algebraic methods for computer aided geometric design. In: Farin, G., Hoschek, J., Kim, M.S. (Eds.), Handbook of Computer Aided Geometric Design. North-Holland, Amsterdam.

Sendra, J., Winkler, F., 1991. Symbolic parameterization curves. J. Symbolic Comput. 12, 607–632.

Walker, R., 1978. Algebraic Curves. Springer-Verlag, New York.

Yang, H.P., Wang, W.P., Sun, J.G., 2004. Control points adjustment for B-spline curve approximation. Computer-Aided Design 36 (7), 639–652.