Noname manuscript No.
(will be inserted by the editor)

A new approach for approximate implicitization of parametric curves

Jinming Wu · Xiaolei Zhang

Received: 19 January 2013 / Revised: 30 June 2013/ Accepted: 07 July 2013

Abstract In this paper, we present a new approach to solve approximate implicitization of parametric curves. The basic idea is to divide the normal parametric curve into several curve segments at three types of critical points and then use multiquadric quasi-interpolation to approximate each curve segment. Meanwhile, we interpolate two endpoints of each segment by using compactly supported radial basis functions in order to maintain the continuity of the adjacent curve segments. The resulting implicit curves possess certain shape preserving and good approximation behaviors.

Keywords Approximate implicitization · Parametric curves · MQ quasi-interpolation · Radial basis functions

Mathematics Subject Classification (2000) 65D17 · 41A05

1 Introduction

Shape representation based on parametric forms has been studied extensively, and the use of parametric representations remains dominant in Computer

This work was supported by the National Natural Science Foundation of China (Nos. 11101366, 11226329 and 11271328), the Natural Science Foundation of Zhejiang Province (No. LQ13A010004) and the Foundation of Zhejiang Educational Committee (No. Y201222928).

Jinming Wu
College of Statistics and Mathematics, Zhejiang Gongshang University, Hangzhou, 310018, China
E-mail: wujm97@mail.zjgsu.edu.cn

Xiaolei Zhang
College of Statistics and Mathematics, Zhejiang Gongshang University, Hangzhou, 310018, China

Aided Geometry Design and Geometric Modeling. However, algebraic models have certain mathematical and computational advantages complementary to the parametric forms, and they are receiving increased attention. Actually, parametric curves/surfaces and implicit curves/surfaces are both important. For example, with the parametric form, it is easy to generate points on a general curve/surface and plot it. On the other hand, it is convenient to determine whether a point is on, inside, or outside a given solid with the implicit treatments [1,2].

For a general parametric curve/surface, we usually cannot compute its exact implicit form. Even though its exact implicit form can be computed, the curve/surface implicitization always involves relatively complicated computation and the degree of the implicit curves/surfaces is high. Another difficulty is that implicit curves/surfaces may have unexpected components and self-intersections which lead to computational instability and topological inconsistency in geometric modeling. All these unsatisfied properties restrict the applications of the exact implicitization in practical use.

Due to these reasons, finding approximate implicitization of parametric curves/surfaces becomes a practical problem. In the past twenty years, many scholars are studying on this issue. The earlier work on approximate implicitization was done by Velho et al.[3], who presented an approximate implicitization scheme from parametric surfaces to implicit surfaces based on wavelet analysis. In 1999, Sederberg et al.[4] proposed a method to solve approximate implicitization by using monoid curves and surfaces, which was made more available in Dokken's work [5,6]. In 2004, Chen and Deng [7] presented the concept of interval implicitization of rational curves and developed the corresponding optimization algorithm. In 2006, Li et al.[8] discussed the approximate implicitization of planar parametric curves by using the piecewise quadratic Bézier spline curves with G^1 continuity. Since the parametric curves/surfaces discussed in these papers are rational, it is required to design different approaches to deal with approximate implicitization of arbitrary parametric curves/surfaces. In 2007, Wang and Wu [9] discussed the approximate implicitization of some regular parametric curves based on radial basis function networks and multiquadric (MQ) quasi-interpolation. In 2008, Wu and Wang [10] presented an algorithm to solve the approximate implicitization of parametric surfaces based on multivariate interpolation with the use of normal constraint points. Very recently, Zhang and Wu [11] proposed another way to solve approximate implicitization of parametric curves by using cubic algebraic splines.

Since the methods in papers [9,11] are carried out independently and each method has their own unique advantages, we may combine their advantages together and develop a simple algorithm. The basic idea is to divide the original normal parametric curves into several curve segments at three types of critical points and then use MQ quasi-interpolation operator to approximate each curve segment. Meanwhile, we interpolate these separated points by using compactly supported radial basis functions (CS-RBFs) in order to maintain the continuity of the adjacent curve segments.

Our proposed algorithm has the following advantages: Firstly, this is a great and essential improvement of our previous work [9]. In fact, all the subsequent numerical examples provided in this paper can not be tackled directly by the method in [9]. Secondly, our proposed method is simple and easy to implement compared with the cubic algebraic spline method in [11]. More importantly, the resulting implicit curves have certain shape preserving.

2 Preliminaries

2.1 Radial basis function

A radial basis function (RBF) is a relatively simple multivariate function generated by a univariate function. Nowadays, the radial basis function has become an effective tool for multivariate scattered data interpolation [12]-[15].

The process of RBF interpolation is as follows: For given scattered data $\{X_j\}_{i=1}^N \subset \mathbf{R^n}$ and the corresponding function values $\{f(X_i)\}_{i=1}^N$, we can construct an interpolant of the form

$$s(X) = \sum_{j=1}^{N} \lambda_j \phi(\|X - X_j\|),\ X = (x_1, x_2, \cdots, x_n), \tag{1}$$

where $\|\cdot\|$ denotes the Euclidean norm, $\phi : \mathbf{R}_+ \to \mathbf{R}$ is a given RBF. The coefficients λ_j can be determined by solving the linear system

$$s(X_i) = \sum_{j=1}^{N} \lambda_j \phi(\|X_i - X_j\|) = f_i,\ i = 1, \cdots, N.$$

The positive definiteness of guarantees that the above interpolation problem possesses a unique solution. If ϕ has compact support, then the positive definite linear system is sparse and reduces computational cost greatly. Thus, we bypass this problem by restricting ϕ to have compact support.

CS-RBFs have only recently been constructed. Wu first constructed a broad variety of CS-RBFs [16]. Very recently, Wendland constructed these functions such that they possess the lowest degree among all CS-RBFs which are positive definite for given space dimension and prescribed order of smoothness [17]. They are radial basis functions which are positive definite on $\mathbf{R^d}$ for a given space dimension d, belong to a prescribed smoothness class, are compactly supported and easy to evaluate.

We find this a useful property in practice and thus provide a good selection of Wendland's functions. Nowadays, CS-RBFs have become a popular tool for multivariate interpolation of large scattered data, implicit surface reconstruction and so on [12].

In order to adapt the interpolation to scattered data of different densities, it is necessary to be able to scale the support of ϕ. So from now on we assume the radius α of support of ϕ is one and replace ϕ by

$$\phi_\alpha(\cdot) = \phi(\cdot/\alpha),\ for\ \alpha > 0.$$

2.2 MQ quasi-interpolation operator $\mathcal{L}_D$

The univariate multiquadric quasi-interpolation of a function $f : [a, b] \to \mathbf{R}$ at the scattered points

$$a = x_0 < x_1 < \cdots < x_n = b,$$

has the form

$$\mathcal{L}f(x) := \sum_{i=0}^{n} f(x_i)\psi_i(x),$$

where each $\psi_i(x)$ is the linear combination of the MQ basis

$$\varphi_i(x) = \sqrt{(x - x_i)^2 + c^2}$$

and c is a shape parameter.

In the summarized paper [18], Franke pointed out that MQ interpolation was best in terms of timing, storage, accuracy, visual pleasantness of surface, and ease of implementation. Although the MQ interpolation (with appended constant) is always solvable, the resulting matrix from using MQ quickly becomes ill-conditioned as the number of points increases. Thus, the research focus on the MQ quasi-interpolation. In the early 1992, Beaton and Powell [19] proposed three kinds of univariate multiquadric quasi-interpolation schemes, namely, $\mathcal{L}_A$, $\mathcal{L}_B$ and $\mathcal{L}_C$, to approximate the function $\{f(x) : x \in [x_0, x_N]\}$ from the space that is spanned by the multiquadrics and linear functions. Afterwards, Wu and Schaback [20] presented another quasi-interpolation formula $\mathcal{L}_D$ on $[x_0, x_N]$ without using the derivative values at the endpoints.

Theorem 1 [20] *MQ quasi-interpolation operator $\mathcal{L}_D$ preserves linear reproduction, monotonicity, convexity and variation-diminishing.*

They proved it can have an $\mathcal{O}(h^2|\log h|)$ error only if at least the shape parameter $c = \mathcal{O}(h)$. Meanwhile, how to select a good value for the parameter c in multiquadric interpolation is well studied by Carlson and Foley [21].

2.3 Curve segment

In order to approximate the parametric curve efficiently, a basic idea is to divide it into several segments. We introduce the following notations and definitions (for details, readers may refer [8]).

The parametric curve $C(t) = \{(x(t), y(t))|\ t \in [a, b]\}$ is said to be a regular curve if for any $a \le t \le b$, $x'(t), y'(t), x''(t), y''(t)$ always exist.

A natural idea is to divide the regular parametric curve into several curve segments possessing relatively good shape, separated by the following three types of critical points.

- A point $C(t_0)$ is called a cusp point of parametric curve $C(t)$ if $x'(t_0) = y'(t_0) = 0$. A cusp point is usually a sharp point on the curve.

– A point $C(t_0)$ is called an inflection point of $C(t)$ if $x'(t_0)y''(t_0)-x''(t_0)y'(t_0) = 0$ and $x'(t_0) \neq 0$. An inflection point of a curve is a point at which the convexity of the curve changes.
– A point $C(t_0)$ is called a vertical point of $C(t)$ if $x'(t_0) = 0$ and $y'(t_0) \neq 0$.

A parametric curve $C(t)$ is called normal if it has a finite number of critical points. It is clear that a rational parametric curve is always normal.

A curve segment $C(t) = (x(t), y(t)),\ t \in [t_1, t_2]$ is said to be triangle convex if the left tangent line and right tangent line meet at v_{12} and the line segment $p_1p_2(p_1 = C(t_1), p_2 = C(t_2))$ and the curve segment $C(t),\ t \in [t_1, t_2]$ form a convex region inside the triangle $[p_1p_2v_{12}]$.

Find the vertical points, the cusp points, and the inflection points by solving the following univariate equations: $x'(t) = 0, x'(t) = y'(t) = 0$ and $x'(t)y''(t) - x''(t)y'(t) = 0$. Let the solutions be $t_i,\ i = 1, \cdots, n-1$. We assume that $a = t_0 < t_1 < \cdots < t_n = b$. Then, $C(t)$ is a triangle convex segment in $[t_i, t_{i+1}], i = 0, \cdots, n-1$.

Remark 1 The command **Solve** in Mathematica 6.0 is used to solve the univariate equation directly, which can find all the real solutions with prescribed accuracy. It is noted that the approximation behavior is not sensitive to the accuracy of the solutions of these systems from the numerical experiments.

3 Approximate implicitization of a parametric curve

From now on, the normal parametric curve $C(t)$ is divided into n triangle convex segments $C(t) = (x(t), y(t)),\ t \in [t_i, t_{i+1}],\ i = 0, 1, \cdots, n-1$. Our discussed problem can be addressed as

Problem 1 Given the curve segment $C(t),\ t \in [t_i, t_{i+1}]$, the approximate implicitization of this curve segment is to find an implicit curve segment such that it not only possesses certain shape preserving but also passes through the two endpoints $C(t_i)$ and $C(t_{i+1})$.

It is pointed out that the certain shape preserving describes the following four possible properties: monotonicity, convexity, smoothness and fairness [22].

We will explain our proposed algorithm in detail. It is primarily based on MQ quasi-interpolation operator $\mathcal{L}_D$ and multivariate scattered data interpolation using CS-RBFs.

Firstly, a set of N_i sampling points is chosen, i.e.,

$$\mathrm{SP}(i) = \Big\{ x_{i_j} = x(t_{i_j}) : \ t_{i_j} \in [t_i, t_{i+1}],\ j = 1, 2, \cdots, N_i \Big\}.$$

We assume, if not specified, $x_{i_1} < x_{i_2} < \cdots < x_{i_{N_i}}$. In fact, this is true if and only if $t_{i_1} < t_{i_2} < \cdots < t_{i_{N_i}}$ or $t_{i_1} > t_{i_2} > \cdots > t_{i_{N_i}}$. Otherwise, there exists $t_{k_0} \in [t_i, t_{i+1}]$ such that $x'(t_{k_0}) = 0$ which is a contradiction.

Secondly, the set of data $\left\{(x(t_{i_j}), y(t_{i_j}))\right\}_{j=1}^{N_i}$ is approximated by MQ quasi-interpolation operator $\mathcal{L}_{\mathcal{D}}y^{(i)}(x)$ as follows

$$(\mathcal{L}_{\mathcal{D}}y)^{(i)}(x) = \sum_{j=1}^{N_i} y(t_{i_j})\psi_j^{(i)}(x), \tag{2}$$

where,

$$\begin{aligned}
\psi_1^{(i)}(x) &= \frac{1}{2} + \frac{\varphi_2^{(i)}(x) - (x - x_{i_1})}{2(x_{i_2} - x_{i_1})},\\
\psi_2^{(i)}(x) &= \frac{\varphi_3^{(i)}(x) - \varphi_2^{(i)}(x)}{2(x_{i_3} - x_{i_2})} - \frac{\varphi_2^{(i)}(x) - (x - x_{i_1})}{2(x_{i_2} - x_{i_1})},\\
\psi_j^{(i)}(x) &= \frac{\varphi_{j+1}^{(i)}(x) - \varphi_j^{(i)}(x)}{2(x_{i_{j+1}} - x_{i_j})} - \frac{\varphi_j^{(i)}(x) - \varphi_{j-1}^{(i)}(x)}{2(x_{i_j} - x_{i_{j-1}})},\ j = 3, \cdots, N_i - 2,\\
\psi_{N_i-1}^{(i)}(x) &= \frac{(x_{i_{N_i}} - x) - \varphi_{N_i-1}^{(i)}(x)}{2(x_{i_{N_i}} - x_{i_{N_i-1}})} - \frac{\varphi_{N_i-1}^{(i)}(x) - \varphi_{N_i-2}^{(i)}(x)}{2(x_{i_{N_i-1}} - x_{i_{N_i-2}})},\\
\psi_{N_i}^{(i)}(x) &= \frac{1}{2} + \frac{\varphi_{N_i-1}^{(i)}(x) - (x_{i_{N_i}} - x)}{2(x_{i_{N_i}} - x_{i_{N_i-1}})},\\
\varphi_j^{(i)}(x) &= \sqrt{(x - x_{i_j})^2 + c^2}
\end{aligned}$$

Obviously, $(\mathcal{L}_{\mathcal{D}}y)^{(i)}(x)$ is nothing more than Wu-Schaback's operator $\mathcal{L}_{\mathcal{D}}$ which is restricted on $[x(t_i), x(t_{i+1})]$.

Thirdly, we define an error function $e^{(i)}(x,y) = y - (\mathcal{L}_{\mathcal{D}}y)^{(i)}(x)$ and compute

$$e_l^{(i)} = e^{(i)}(X_i),\ e_r^{(i)} = e^{(i)}(X_{i+1}),$$

where $X_i = (x(t_i), y(t_i))$ and $X_{i+1} = (x(t_{i+1}), y(t_{i+1}))$.

Moreover, we construct the following interpolant

$$\varepsilon^{(i)}(x,y) = \lambda_l^{(i)}\phi_{\alpha_{l_i}}(\|X - X_i\|) + \lambda_r^{(i)}\phi_{\alpha_{r_i}}(\|X - X_{i+1}\|) \tag{3}$$

satisfying the interpolation conditions

$$\varepsilon^{(i)}(X_i) = e_l^{(i)},\ \varepsilon^{(i)}(X_{i+1}) = e_r^{(i)},$$

where, $\phi_{\alpha_{l_i}}(\|X - X_i\|)$ and $\phi_{\alpha_{r_i}}(\|X - X_{i+1}\|)$ are two CS-RBFs with the support of radius α_{l_i} and α_{r_i}, respectively.

Finally, we obtain a formula in the form:

$$y = (\mathcal{L}_{\mathcal{D}}y)^{(i)}(x) + \varepsilon^{(i)}(x,y). \tag{4}$$

If we set

$$F^{(i)}(x,y) = y - (\mathcal{L}_{\mathcal{D}}y)^{(i)}(x) - \varepsilon^{(i)}(x,y), \tag{5}$$

then

$$\mathcal{C}_i = \Big\{(x, y)|\ F^{(i)}(x, y) = 0,\ x \in [x(t_i), x(t_{i+1})]\Big\} \tag{6}$$

is our implicit curve segment. It is used to approximate the parametric curve segment $C(t),\ t \in [t_i, t_{i+1}]$.

It is easy to prove that

$$F^{(i)}(X_i) = 0,\ F^{(i)}(X_{i+1}) = 0.$$

The first term of the right-hand side of (4) is considered to be a base approximation to the parametric curve segment. While, the second term represents local modification in order to have the property of endpoint interpolation.

If we let

$$F(x, y) = \begin{cases} F^{(0)}(x, y), & (x, y) \in \mathcal{C}_0; \\ \vdots \\ F^{(n-1)}(x, y), & (x, y) \in \mathcal{C}_{n-1}; \end{cases} \tag{7}$$

then

$$\mathcal{C} = \{(x, y)|\ F(x, y) = 0\} \tag{8}$$

is our final implicit curve. It can approximate the parametric curve $C(t),\ t \in [a, b]$. Actually, $F(x, y)$ is a piecewise and continuous function.

It is inevitable that two natural questions will arise. The first question is how to select the set of sampling points on each parametric curve segment. We find that the proposed algorithm is not sensitive to the number of the selected sampling points. Meanwhile, relatively good approximation behavior can be achieved when a small amount of sampling points are chosen uniformly. This fact will be confirmed in the numerical examples presented later on. In sum, we have a great freedom to select the number and positions of sampling points and we do not discuss it detail in this paper. As to the adaptive sampling of parametric curves, we recommend that the readers can refer the paper [23].

Another question is how to compute the two coefficients $\lambda_l^{(i)}$ and $\lambda_r^{(i)}$. Since the $\varepsilon^{(i)}(x, y)$ is considered to be the local detail in order to interpolate two endpoints, it is not sensitive to the radius of CS-RBFs. Therefore, if we set $h_i = \|X_i - X_{i+1}\|$ and assume $\alpha_{l_i}, \alpha_{r_i} \le h_i$, then we have $\lambda_l^{(i)} = e_l^{(i)}$ and $\lambda_r^{(i)} = e_r^{(i)}$. The interpolation function can be simply computed in the form

$$\varepsilon^{(i)}(x, y) = e_l^{(i)} \phi_{\alpha_{l_i}}(\|X - X_i\|) + e_r^{(i)} \phi_{\alpha_{r_i}}(\|X - X_{i+1}\|). \tag{9}$$

Remark 2 We use Command **ContourPlot** in Mathematica 6.0 directly to plot the implicit curve segment $\mathcal{C}_i$ on $[x(t_i), x(t_{i+1})]$. Certainly, other components are not useful and should be deleted. We put together all the curve segments $\{\mathcal{C}_i\}_{i=0}^{n-1}$ to obtain the final implicit curve $\mathcal{C}$.

We now show how to estimate the approximation error between the parametric curve and its corresponding implicit curve. The distance from $C(t),\ t \in$

$[t_i, t_{i+1}]$ to implicit curve $\mathcal{C}_i$ can be evaluated by the following approximation error function [8]

$$e(F^{(i)}, t) = \frac{F^{(i)}(x(t), y(t))}{[F_x^{(i)}(x(t), y(t))^2 + F_y^{(i)}(x(t), y(t))^2]^{\frac{1}{2}}}.$$

Since it involves the partial derivative in denominator and is not convenient in practical use, we replace $e(F^{(i)}, t)$ with $\tilde{e}(F^{(i)}, t)$:

$$\tilde{e}(F^{(i)}, t) = F^{(i)}(x(t), y(t)). \tag{10}$$

The approximation error between $C(t),\ t \in [t_i, t_{i+1}]$ and $\mathcal{C}_i$ is defined as follows

$$\tilde{e}(F^{(i)}) = \max_{t_i \le t \le t_{i+1}} |\tilde{e}(F^{(i)}, t)|. \tag{11}$$

In practice, we discrete t as $t_{i_j} = t_i + \frac{j}{M_i}(t_{i+1} - t_i),\ j = 0, \cdots, M_i$ for a proper value of M_i, say $M_i = 100$, and evaluate the approximation error as $\max\limits_{0 \le j \le M_i} |\tilde{e}(F^{(i)}, t_{i_j})|$.

Therefore, the approximation error between $C(t),\ t \in [a, b]$ and $\mathcal{C}$ is defined

$$\tilde{e}(F) = \max_{0 \le i \le n-1} \tilde{e}(F^{(i)}). \tag{12}$$

4 Main algorithm

With the above preparations, the algorithm of approximate implicitization of a normal parametric curve is outlined as follows.

Algorithm 2 *Approximate implicitization of a normal parametric curve*

Input *A normal parametric curve* $C(t) = (x(t), y(t)),\ a \le t \le b$, *and a small threshold* δ.

Output *An implicit curve* $\mathcal{C} = \{(x, y)|\ F(x, y) = 0\}$ *satisfying each* $E(\tilde{e}(F)) \le \delta$.

Step 1 *Divide the normal parametric curve into several triangle convex segments and let* $a = t_0 < t_1 < \cdots < t_{n-1} < t_n = b$ *be the parametric values corresponding to the critical points and two endpoints. Set* $i = 0$.

Step 2 *Choose sampling points on* $C(t),\ t \in [t_i, t_{i+1}]$ *and construct the implicit curve* $\mathcal{C}_i = \{(x, y)|\ F^{(i)}(x, y) = 0\}$ *on* $[t_i, t_{i+1}]$, *where* $F^{(i)}(x, y)$ *is defined by (5).*

Step 3 . *If* $E(\tilde{e}(F^{(i)})) \le \delta$, *then this procedure terminates output* $F^{(i)}(x, y)$. *Otherwise, if* $E(\tilde{e}(F^{(i)})) > \delta$, *then we subdivide the interval* $[t_i, t_{i+1}]$ *at midpoint and go to Step 2 on each subinterval.*

Step 4 . *Set* $i = i+1$. *If* $i \le n-1$ *then go to Step 2; Otherwise, stop and output the implicit curve* $\mathcal{C} = \{(x, y)|\ F(x, y) = 0\}$, *where* $F(x, y)$ *is defined by (7).*

The following theorem is a direct consequence from the properties of $\mathcal{L}_\mathcal{D}$ and interpolation at the separated points.

Theorem 3 *With the above* **Algorithm 2**, *one obtains a global continuous implicit curve which keeps the cusp points and possesses certain shape preserving of the original parametric curve .*

5 Numerical examples

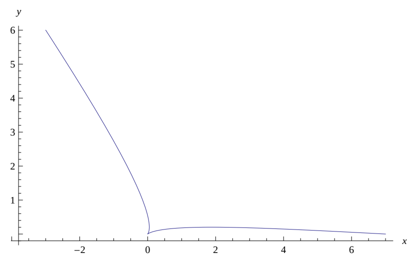
In this section, some numerical examples are provided to illustrate the proposed method is flexible and effective.

Example 1 Consider the following parametric curves from [8,11]

$$\begin{aligned}
C_1(t) &= (5t^3+2t^2, t^4-3t^3+2t^2),\\
C_2(t) &= (3t^6+t^5-2t^4+38t^3-5t^2-14t, t^6-12t^5-2t^4+2t^3-7t^2+13t),\\
C_3(t) &= (\frac{5t^5-16t^4+10t^3+4t^2}{0.1t^3+0.1t^2-2t+12.5}, \frac{t^5+t^4+2t^3-16t^2}{0.1t^3+0.1t^2-2t+12.5}),\\
C_4(t) &= (\sin(2t)+\ln(5t^4+2)+3t^2, 3e^{t^2-1}+\cos(t/5)+2t^7).
\end{aligned}$$

The parameters for curves of $C_1(t)$, $C_2(t), C_3(t)$ and $C_4(t)$ take values in $[-1,1]$, $[-1,1]$, $[-1,2.25]$, and $[-1,1]$. The original parametric curves denoted by black lines are shown in the left of Fig.1–Fig.4.

Their implicit curves together with the separated points, denoted by red lines and thick dots, are shown in the right of Fig.1-Fig.4. Moreover, the maximum errors and variances which are evaluated at arbitrary 100 testing points are listed in Tables 1, 2, 3 and 4, respectively.



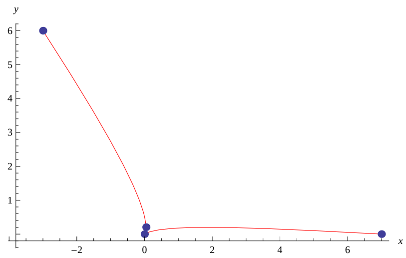


Fig. 1 $C_1(t)$ and its approximate implicit curve

Table 1 Error analysis of curve $C_1(t)$

Range of t	(-1,-0.27)	(-0.27,0)	(0,1)
Approximation error	0.054	0.053	0.0047
Variance	1.89×10^{-4}	3.49×10^{-4}	4.71×10^{-3}

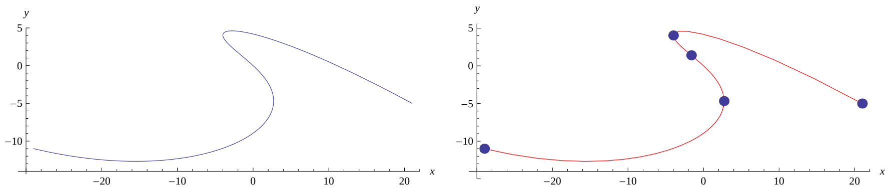


Fig. 2 $C_2(t)$ and its approximate implicit curve

Table 2 Error analysis of curve $C_2(t)$

Range of t	(-1,-0.31)	(-0.31,0.12)	(0.12,0.4)	(0.4,1)
Approximation error	0.017	0.019	0.052	0.11
Variance	7.24×10^{-4}	2.01×10^{-3}	1.46×10^{-4}	4.85×10^{-4}

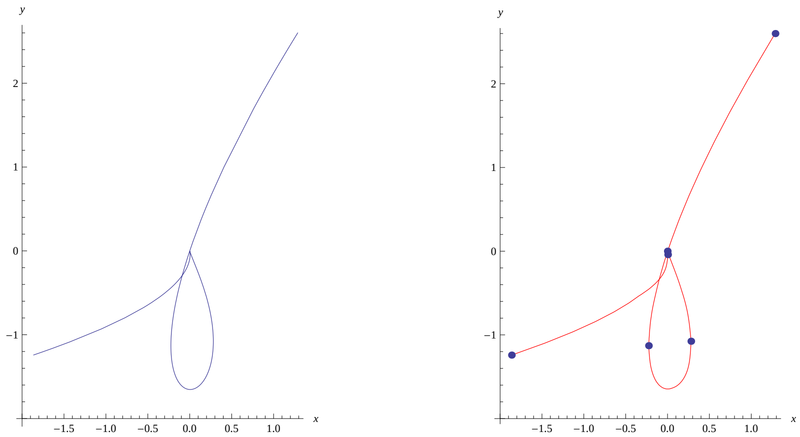


Fig. 3 $C_3(t)$ and its approximate implicit curve

Table 3 Error analysis of curve $C_3(t)$

Range of t	(-1,-0.19)	(-0.19,0)	(0,0.97)	(0.97,1.79)	(1.79,2.25)
Approximation error	0.041	0.017	0.069	0.055	0.050
Variance	1.98×10^{-4}	3.47×10^{-5}	3.77×10^{-4}	1.78×10^{-4}	1.32×10^{-4}

Table 4 Error analysis of curve $C_4(t)$

Range of t	(-1,-0.26)	(-0.26,1)
Approximation error	0.017	0.029
Variance	1.54×10^{-5}	3.15×10^{-5}

In fact, we can compute the exact implicit form of the first three curves with a Gröbner bases method. However, the expressions for them are polynomials of high degree with large numbers of coefficients. Moreover, curve $C_4(t)$ does

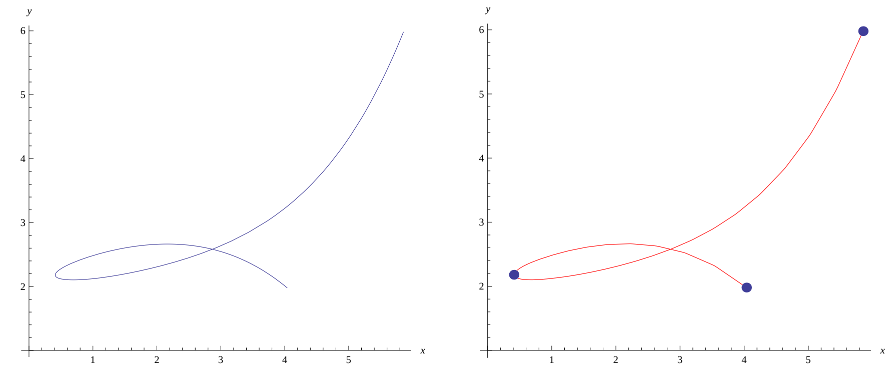


Fig. 4 $C_4(t)$ and its approximate implicit curve

not have an exact implicit form. Thus, it is convinced that the approximate implicitization of parametric curves is practical and useful.

6 Conclusion

We describe an algorithm to find an approximate implicitization of a given normal parametric curve. With the proposed algorithm, we obtain a continuous implicit curve which possesses certain good shape preserving. The proposed method is simple and easy to implement compared to the other existing methods.

However, the proposed algorithm is hard to be generalized to tackle the case of parametric surfaces directly. Our ultimate target is to discuss approximate implicitization of the parametric surfaces. The essential generalization remains to be our future work.

Acknowledgements The authors would like to thank the anonymous referees for their valuable comments and suggestions on the paper, which have considerably improved the original paper.

References

1. Bloomenthal J(1997) Introduction to Implicit Surfaces. Morgan Kaufmann Publishers Inc, San Francisco
2. Gomes J et al.(2009) Implicit Curves and Surfaces: Mathematics, Data Structures and Algorithms. Springer, London
3. Velho L, Gomes J (1996) Approximate conversion from parametric to implicit surfaces. Computer Graphics Forum 15: 327-337
4. Sederberg TW, Zheng J, Klimaszewski K, Dokken T(1999) Approximate implicit using monoid curves and surfaces. Graphical Model and Image Processing 61: 177-198
5. Dokken T(2001), Approximate implicitization, Mathematical Methods for Curves and Surfaces, Oslo 2000, T.Lyche and L.L.Schumaker(eds), Vanderbilt University Press, 81-102

6. Dokken T, Thomassen JB(2003) Overview of approximate implicitization, in Topics in Algebraic Geometry and Geometric Modeling. AMS Cont. Math 334:169-184
7. Chen FL, Deng L (2004) Interval implicitization of rational curves. CAGD 21:401-415
8. Li M, Gao XS, Chou SC(2006) Quadratic approximation to plane parametric curves and its application in approximate implicitization. Visual Computers 22: 906-917
9. Wang RH, Wu JM (2007) Approximate implicitization based on RBF networks and MQ quasi-interpolation. J Comput Math 25: 97-103
10. Wu JM, Wang RH (2008) Approximate implicitization of parametric surfaces by using compactly supported radial basis functions. Comput Math Appl 56: 3064-3069
11. Zhang XL, Wu JM (2009) Approximate implicitization of parametric curves using cubic algebraic splines. Math Probl Eng vol. 2009, Article ID 327457, 12 pages, 2009. doi:10.1155/2009/327457
12. Buhmann M (2000) Radial Basis Function: Theory and Implementations. Cambridge University Press, Cambridge
13. Schaback R (2005) Multivariate interpolation by polynomials and radial basis functions. Constr Approx 21: 293-317.
14. Wendland H (2005) Scattered Data Approximation. Cambridge University Press, Cambridge
15. Fasshauer G (2007)Meshfree Approximation Methods with Matlab. World Scientific Publishers, Singapore
16. Wu ZM (1995) Multivariate compactly supported positive definite radial functions. Adv Comput Math 4: 283-292
17. Wendland H (1995) Piecewise polynomial, positive definite and compactly supported radial basis functions of minimal degree. Adv Comput Math 4: 389-396
18. Franke R (1982) Scattered data interpolation: test of some methods. Math Comput 38: 181-200
19. Beaton RK, Powell MJ (1992) Univariate multiquadraic approximation: quasi-interpolation to scattered data. Constr Approx 8: 275-288
20. Wu ZM, Schaback R (1994) Shape preserving interpolation with radial basis function. Acta Math Appl Sin 10: 441-446
21. Carlson RE, Foley TA (1991) The parameter $\mathbf{R^2}$ in multiquadric interpolation. Comput Math Appl 21: 29-42
22. Wu JM (2011) A new approach for shape preserving interpolating curves. Comput Math Appl 61: 1425-1430
23. de Figueiredo LH (1995) Adaptive Sampling of Parametric Curves. Academic Press, Brasil