MM Research Preprints, 52–63
MMRC, AMSS, Academia, Sinica, Beijing
No. 21, December 2002

Approximate Implicitization of Planar Parametric Curves using Quadratic Splines[1)]

Xiao-Shan Gao and Ming Li
Institute of System Sciences, AMSS, Academia Sinica
Beijing 100080, China
(xgao,mli)@mmrc.iss.ac.cn

Abstract. A quadratic Bézier spline with G^1-continuity is given to approximate a planar parametric curve. In its construction, the parametric curve is first divided into several segments. A rational quadratic Bézier curve is then applied to approximate each segment and recursive subdivision for the segment may be used if the approximation error is out of the given range. With this method, we not only keep the shape but also the direction of the parametric curve with simple computation using quadratic curves. High accuracy of the approximation is also achieved with relatively small number of conics.

Keywords: approximation, Bézier curve, implicitization, parametric form

1. Introduction

Parametric and implicit representations are two main forms for planar curves in $CAGD$. Each of the two representations has its own advantages. With the parametric representation, $x = x(t)/d(t), y = y(t)/d(t)$, where $x(t), y(t), d(t)$ are often polynomials (i.e. the rational form), it is easy to get every point in the curve and plot it. While for the implicit form $f(x, y) = 0$, it's easy to determine whether a point is on the curve. The availability of both often results in simpler computation. For example, with the two forms available, we can get the intersection of two planar curves by solving a one-variable equation.

For any rational parametric curves, we can always convert it into implicit form. The process is called implicitization. While the converse, called parametrization, is not always true. The implicitization techniques include some classic algebraic geometry theories, such as the resultant based method, the Groebner basis method, the characteristic set method, and others such as moving lines and surfaces. For a survey, please consult [13].

However, it is not necessary to give an exact implicit form for a parametric curve. This is in part due to the fact that the implicitization always involves relatively complicated process and the result of the implicit form might have large number of coefficients. Another important reason is that an *exact* implicitization form may have self-intersections or some unwanted branches. Figure 1 is a cubic Bézier curve but the implicit form it in Figure 2 is a curve with self-intersection. Figure 3 shows a quartic Bézier curve and Figure 4, while its implicit form as shown in Figure 4, includes an extraneous component in its neighborhood.

[1)]The work is supported in part by Chinese NSF under an outstanding youth grant (NO. 69725002) and by a National Key Basic Research Project of China (NO. G1998030600).

All these unwanted properties of the implicitization of a curve limited the applications of the exact implicitization in $CAGD$.

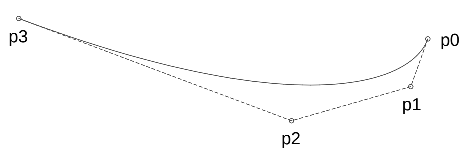

Fig. 1. A cubic Bézier curve

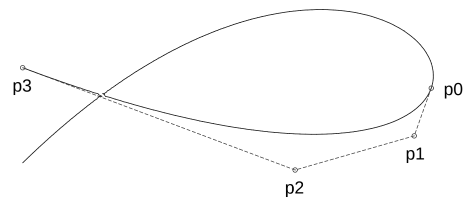

Fig. 2. Self-intersection

Due to these reasons, approximate implicitation has been proposed. Montaudouin and Tiller considered a power series to obtain local explicit approximations, about a singular point, to polynomials curves and surfaces [10]. Chuang and Hoffmann extended this method to give a local approximation to a parametric curve or surface [2]. And the method was capable of exact implicitization when the approximation was of the same degree as the implicit form. Dokken proposed a new way to approximate the parametric curve and surface globally in the sense that the approximation was valid within the whole domain of the curve segment or surface patch [3]. Sederberg etc. used monoid curves and surfaces to find an approximate implicit equation and an approximate inversion map of a planar rational parametric curve or surface [12]. Juttler etc. generated a quadratic B-spline approximation via orthogonal projection in Sobolev spaces, resulting in a C^1 spline of degree four [7]. Using active B-splines, Pottmann presented an active contour model for parametric curve and surface. The active curve or surface adapted to the model shape to be approximated in an optimization algorithm [11].

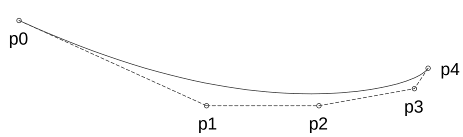

Fig. 3. A quartic Bézier curve

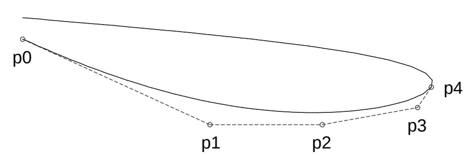

Fig. 4. Unwanted branch

This paper considers an approximate implicitization method with piecewise quadratic splines in Bézier form. The quadratic curve in Bézier form is used since its many familiar and classic property for us. In our procedure, we first divide the parametric curve into several parts. Then a quadratic rational Bézier curve is adapted to approximate each segment. It also has the same endpoints and corresponding tangent directions with the approximated segment, ensuring a spline generated with G^1-continuity. Recursive subdivision for each segment may be used if the approximation error is out of the control range.

The rest of the paper is organized as follows. We introduce the rational quadratic Bézier curve and give some of its properties in section 2.. Our main result, the approximate implicitization method, is illustrated in section 3.. After some examples with error analyzes shown in section 4., we conclude this paper with future works in section 5..

2. Rational Quadratic Bézier Curve

Conics (i.e. a conics segment) particularly expressed in Bézier form, are widely used in $CAGD$ due to their familiar and classical properties [4, 6, 9]. A general properties of conics in Bézier representation were discussed in [8]. Farin used it to discuss the problem of the construction of curvature continuous, planar curves [4]. In his master thesis, Ming Li used the Bézier form to give a quadratic spline with G^1-continuity that interpolated a set of scattered data with associated normal directions [9].

Let ϕ_i denote the quadratic Bernstein basis

$$\phi_0 = (1-t)^2, \phi_1 = 2t(1-t), \phi_2 = t^2$$

and let

$$P(t) = \frac{\omega_0 P_0 \phi_0(t) + \omega_1 P_1 \phi_1(t) + \omega_2 P_2 \phi_2(t)}{\omega_0 \phi_0(t) + \omega_1 \phi_1(t) + \omega_2 \phi_2(t)}; \quad 0 \le t \le 1, P_i \in R^2. \tag{1}$$

be a rational quadratic curve in *Bézier form*, in which P_i are called *control points*, ω_i the *weights* and P_i form the so-called *control polygon*. We can also write (1) in its *barycentric coordinates* with respect to the triangle $P_0,\ P_1,\ P_2$ as,

$$P(t) = \sum P_i \psi_i(t),$$

with

$$\psi_i(t) = \frac{\omega_i \phi_i(t)}{\phi(t)},\ \phi(t) = \sum \omega_i \phi_i(t). \tag{2}$$

Here we list some useful properties of rational quadratics [4, 8]:

(P1) *Weights Uniqueness:* Since $\omega_0\omega_2/4\omega_1^2$ remains unchanged for a rational quadratic, without loss of generality we may assume that $\omega_0 = 1, \omega_2 = 1$ for any nondegenerate conics. We can then rewrite (1) as:

$$P(t) = \frac{P_0 \phi_0(t) + \omega P_1 \phi_1(t) + P_2 \phi_2(t)}{\phi_0(t) + \omega \phi_1(t) + \phi_2(t)}; \quad 0 \le t \le 1, P_i \in R^2. \tag{3}$$

We call this form of rational quadratic its *standard form* and we'll use this form throughout.

(P2) *Convex Hull:* The segment $P(t),\ 0 \le t \le 1$ lies in the convex hull of the control polygon for $\omega > 0$.

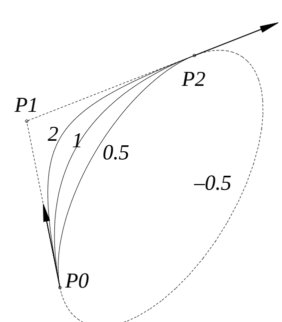

Fig. 5. Type Parameter

(P3) *Endpoints Interpolation:* It is easy to see that,

$$P(0) = P_0, \qquad P(1) = P_2,$$
$$P'(0) = 2\omega(P_1 - P_0), \quad P'(1) = 2\omega(P_2 - P_1).$$

We can see that $P(t)$ goes through the endpoints P_0, P_2 and if we extend from P_0 at its tangent direction and P_2 at reverse, they'll meet at P_1.

(P4) *Control Point at Infinity:* For a given conics, if the tangent lines at the endpoints are parallel, we can treat them as meeting at infinity and write it in Bézier form as

$$P(t) = \frac{P_0\phi_0(t) + \omega\mathbf{d}\phi_1(t) + P_2\phi_2(t)}{\phi_0(t) + \phi_2(t)}; \quad 0 \le t \le 1, P_i \in R^2. \tag{4}$$

where **d** is the tangent direction at either of the endpoints.

(P5) *Type Parameter:* The type of the conic is uniquely determined by the weight ω. Specifically, for *omega* < 1, we obtain an ellipse segment; for $\omega = 1$, a parabola; and for $\omega > 1$, a hyperbola. If we change the sign of ω, we'll get a complementary segment of the conics, the other part of the quadratic curve outside the control polygon. See Figure 5.

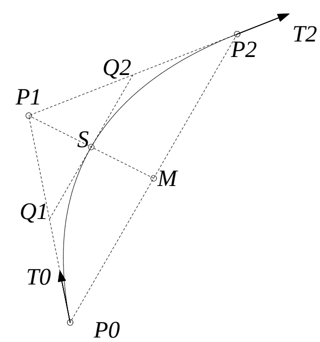

Fig. 6. Shoulder point

(P6) *Shoulder Point:* The point $S = P(\frac{1}{2})$ is called the *shoulder point*. It can be computed from

$$S = \frac{1}{2}(Q_0 + Q_1), \tag{5}$$

where,

$$Q_0 = \frac{P_0 + \omega P_1}{1+\omega}, \quad Q_1 = \frac{\omega P_1 + P_2}{1+\omega}$$

The tangent line at the shoulder point is spanned by Q_0 and Q_1 and is parallel to the line P_0P_2. S is the unique point in the curve $P(t)$, $0 \le t \le 1$, that has the maximum distance to line P_0P_2. See also Figure 6.

(P7) *Points Evaluation [5]:* A rational quadratic may be evaluated by the *de casteljau Algorithm*,

$$P_i^r(t) = (1-t)\frac{\omega_i^{r-1}}{\omega_i^r}P_i^{r-1} + t\frac{\omega_{i+1}^{r-1}}{\omega_i^r}P_{i+1}^{r-1}(t); \quad r = 1,2,\ i = 0,\dots,2-r \tag{6}$$

with

$$\omega_i^r(t) = (1-t)\omega_i^{r-1}(t) + t\omega_{i+1}^{r-1}(t),$$

where, $P_i^0(t) = P_i$ and $\omega_i^0(t) = \omega_i$.

(P8) *Implicit form:* The implicit form of a rational quadratic is given by

$$\psi_1^2 = 4\omega^2\psi_0\psi_2, \tag{7}$$

where ψ_i are barycentric coordinates of a point on the rational quadratic with respect to the triangle $P_0,\ P_1,\ P_2$ as described in (2).

3. Approximation of Planar Parametric Curves

Using the properties of rational quadratic Bézier curves described in section 2., we now can give our approximate implicitization method in this section. We'll mainly go deep into some aspects of curve segmentation, segment approximation, including disposal of cusp points each. An the end of this section, the discussion of the error control is given.

3.1. Curve Segmentation

The parametric curve will be first divided into several *triangle convex segments.* It is based on three types of points, *the cusp points, the inflection points* and *the vertical points.*

A curve segment $c(t) = (x(t), y(t)),\ 0 \le t \le 1$ is said to be *convex* if it lies on the same side of the tangent line at each point on it. Suppose $c(t)$ a convex segment, write d_i a vector starting from $c(i)$ with direction $c'(i),\ i = 0, 1$. If d_0 has an intersection with d_1, we'll call it a *triangle convex curve* in this paper. The case that $d_0,\ d_1$ parallel is also included. The quadratic (3) is a triangle convex curve while $\omega > 0$ and not while $\omega < 0$. You can get it from Figure 6.

A *cusp point* on $c(t)$ is a point with zero derivatives $(x'(t), y'(t))$.

An *inflecion point* of a curve is a point at which the concavity of the curve changes. For a parametric curve $c(t)$, we can usually convert the acquirement of the inflection points to the solvement of an equation $(y'(t)/x'(t))' = 0$, when $x'(t) \neq 0$, with t.

We define a *vertical point* in this paper a non-cusp point with $x'(t)$ vanishes at it.

Figure 7 shows some types of points in two curves.

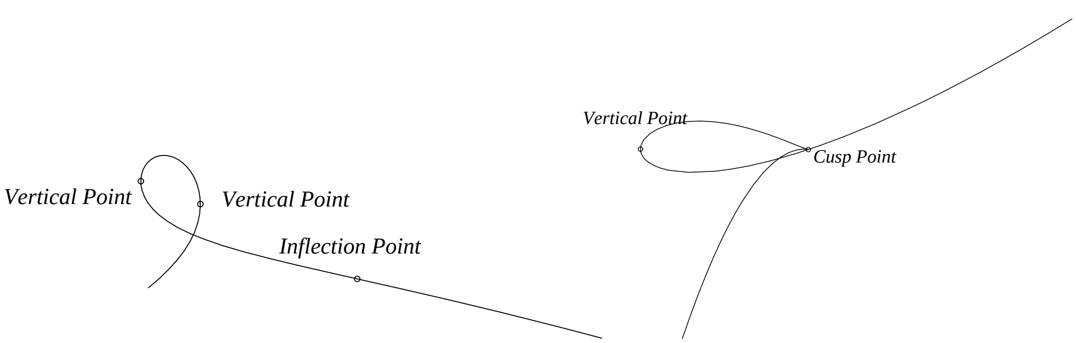

Fig. 7. Point Type

In our segmentation procedure, we'll divide $c(t)$ at the inflection points and vertical points. For a cusp point $C = c(t_0)$ on $c(t)$, we'll add two points $S_0,\ S_1$ on the curve at each side of C. The curve is not divided at c but at S_0, S_1. It's must be satisfied for the selection of the two points that the distance of C to line S_0S_1 is less than δ, the error range. We usually set $S_0 = c(t_0 - \epsilon)$ and $S_1 = c(t_0 + \epsilon)$ with a small value ϵ.

3.2. Segment Approximation

We'll show how to approximate a triangle convex segment here. For a given triangle convex segment $r(t),\ 0 \le t \le 1$, suppose P_0, P_2 its endpoints and T_0, T_2 the associated tangent directions. Write the intersection point of T_0, T_2 as P_1. Then any rational quadratic $P(t)$ interpolating points P_0, P_2 with the tangent directions T_0, T_2 must lie in the group below,

$$P(\omega, t) = \frac{P_0\phi_0(t) + \omega P_1\phi_1(t) + P_2\phi_2(t)}{\phi_0(t) + \omega\phi_1(t) + \phi_2(t)}; \quad 0 \le t \le 1. \tag{8}$$

where the weight $\omega \ge 0$. If T_0 is parallel to T_2, the form with infinite control point is taken.

$$P(\omega, t) = \frac{P_0\phi_0(t) + \omega T_0\phi_1(t) + P_2\phi_2(t)}{\phi_0(t) + \phi_2(t)}; \quad 0 \le t \le 1. \tag{9}$$

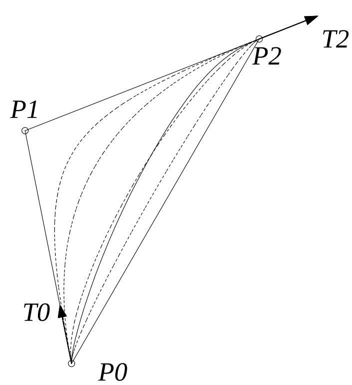

Fig. 8. Approximation Curve Group

Figure 8 shows some curves in the quardratic group. A proper value must be set for ω so that it has a better approximation to $c(t) = (x(t), y(t))$. Suppose the quadratic curve group $P(\omega, t) = (Px(\omega, t), Py(\omega, t))$, the selection of the weights might lead to one of the following optimization problems:

$$\begin{gathered} \min_{\omega}(\max_{t}(d(\omega, t))), \\ \min_{\omega}(\max_{t}(d(\omega, \sigma(t)))), \\ \min_{\omega}(\textstyle\int_0^1 |Py(\omega, t) - y(t)| d(Px(\omega, t) - x(t))dt), \end{gathered} \tag{10}$$

where, $d(\omega, t)$ is the distance between $r(t)$ and $P(\omega, t)$ and $d(\omega, \sigma(t))$ is the distance between $r(t)$ and $P(\omega, \sigma(t))$, the point on $P(\omega, t)$ with minimum distance to $r(t)$.

But all these expressions involve complex computations and they are hard to practice. We'll give another approximation method with the properties of the shoulder point in the following.

Remark 3.1 *With ω definite, we can get the error between two curves with either of the expression in (10). In fact, we'll use the error estimate with the second expression in our procedure.*

Regarding that the shoulder point in a conics has the maximum distance to the line determined by its two endpoints, we obtain the algorithm below.

Algorithm of approximation of a triangle convex segment:

1. According to the interpolating requirements at the endpoints of $P(\omega, t)$, set $P(\omega, t)$ as (8), or (9) for the case that T_0 and T_2 parallel.

2. Select a point $M = (Mx, My)$ in $r(t)$ with the maximum distance to line P_0P_2. Set d the direction of P_0P_2, the acquirement of M can be converted to the solvement of the following equation in t.
$$P'(t) \cdot d = 0.$$
The equation can be solved by Newton's Iterate Method with the initial value of $t = 0.5$. Write the solution $t = t_0$.

3. Let the shoulder point $S(\omega)$ of $P(\omega, t)$, as expressed in (5), have a minmum distance to M. Write $P_i = (x_i, y_i),\ i = 0, 1, 2$ and assume $d(M, S)$ the distance between M and $S(\omega)$. We have
$$S(\omega) = (Sx, Sy) = \frac{P_0 + 2\omega P_1 + P_2}{2(1+\omega)} = (\frac{x_0 + 2\omega x_1 + x_2}{2(1+\omega)}, \frac{y_0 + 2\omega y_1 + y_2}{2(1+\omega)}).$$
Let
$$\frac{\partial d^2(M, S)}{\partial t} = 0,$$
where,
$$d^2(M, S) = (Mx - Sx)^2 + (My - Sy)^2,$$
In the end,we get
$$\omega = \frac{x_0 - \alpha y_0 + x_2 - \alpha y_2 - 2(Mx - \alpha My)}{2(x - \alpha My - x_1 + \alpha y_1)}$$
where,
$$\alpha = \frac{2y_1 - y_0 - y_2}{2x_1 - x_0 - x_2}.$$

4. If the approximation error between $P(t)$ and $c(t)$ is less than δ, the error range, end this procedure. If not, divide the segment into two parts at point $r(t_0)$ and reparametrizate two the segments into $[0, 1]$. Repeat the approximation method until the error is less than δ.

Remark 3.2 *In a geometry view, we just want to make the area error between $P(t)$ and $c(t)$ have a less value with the shoulder approximation method.*

Fig. 9. Approximate segments

Remark 3.3 *For the segment with cusp on, which is usaully not a triangle convex segment. We set* ω *to* 1 *for this case, since we can see that the approximation error must be less than* δ *for any* $\omega > 0$ *from the segmentation around cusp points.*

We show in Figure 9 a spline with the number of segments $1, 2$ and 3 to approximate it respectively.

$$c_1(t) = (t^6 + t^5 - 2t^3 + 3t^2 + 12t, t^6 - t^5 + t^4 - 4t^3 - 2t^2 + 24t), \ -1 \le t \le 1.$$

The meaning of the content of the table is,

1. Range: the range of t of the corresponding segment.
2. Error: the error between the segment and its approximate quadratic.
3. Weight:the weight with respect to the approximate quadratic.

Analyze of a Triangle Convex Segment c_1

	s_0	s_{00}	s_{01}	s_{000}	s_{001}
Range	(-1,1)	(-1,.219957)	(.219957,1)	(-1,-.344147)	(-.344147,.219957)
Error	.728422	.480050	.058097	.036679	.001518
Weight	.265228	.243797	1.305564	.549220	.950157

Table 1

3.3. Algorithm of Approximate Implicitization

Approximate Implicitization Algorithm:

1. Segment the curve with the curve segmentation method.

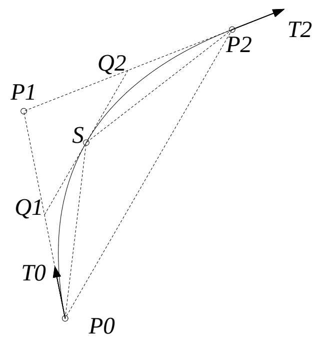

Fig. 10. Error Control

2. For each segment without cusp points, approximate it with the segment approximation method. For a segment with cusp point, select the corresponding weight of it 1.

3. Implicitizate each segment of the quadratic approximate spline with (7).

The approximtion error is ensured to be controlled to any degree from the following theorem.

Theorem 3.4 *With the approximate implicitization algorithm, the approximation error is convergent to zero.*

We give an geometrical explanation for this theorem. See also Figure 10. We have that $P(t)$ and $c(t)$ are contained in the same triangle $P_0P_1P_2$. If we look at M and S as the same point. After one step of recursive segmentation, the curve are contained in triangle $P_0Q_0P_1$ and $P_1Q_1P_2$ respectively. The area between the two curve is restricted by the area of the control triangle. The distance error of the two curve must be less than the height of the triangle also Fig.10. Each of the two properties can ensure the existance of the theorem.

4. Examples

We present examples here, you can find all the curves' definitions in our examples in [2].

$$c_2(t) = (5t^3 + 2t^2, t^4 - 3t^3 + 2t^2),$$
$$c_3(t) = (3t^6 + t^5 - 2t^4 + 38t^3 - 5t^2 - 14t, t^6 - 12t^5 - 2t^4 + 2t^3 - 7t^2 + 13t),$$
$$c_4(t) = ((5t^5 - 16t^4 + 10t^3 + 4t^2)/d(t), (t^5 + t^4 + 2t^3 - 16t^2)/d(t)).$$

where

$$d(t) = 0.1t^3 + 0.1t^2 - 2t + 12.5.$$

The curves of $c_2(t), c_3(t)$ with t in $[-1, 1]$ and $c_4(t)$ with t in $[-1, 2.25]$ and their approximate splines are shown in Fig.11,12,13. Their error analyse are shown in table 2,3,4 respectively.

Fig. 11. c_2 and its approximate splines

Error Analyze of Curve c_2

	s_0	s_1	s_{00}	s_{01}	s_{10}	s_{11}
Range	(-1,-.01)	(.01,1)	(-1, -.652605)	(-.652605,-0.01)	(.01,.609701)	(.609701,1)
Error	.017633	.020224	.000158	.009626	.0062282465	.002725
Weight	.881178	.881178	.773046	1.021974	.7802260357	1.086915

Table 2

Fig. 12. c_3 and its approximate splines

Error Analyze of Curve c_3

	s_0	s_1	s_2
Range	(-1,-.817709)	(-.817709,-.181788)	(-.181788,.616079)
Error	.019212	.099744	. 063933
Weight	.569856	1.263689	.337851

	s_3	s_{10}	s_{11}
Range	(.616079,1)	(-.817709,-.928236)	(-.928236,-.181788)
Error	.009611	.017402	.007267
Weight	.992093	1.14852	.939973

Table 3

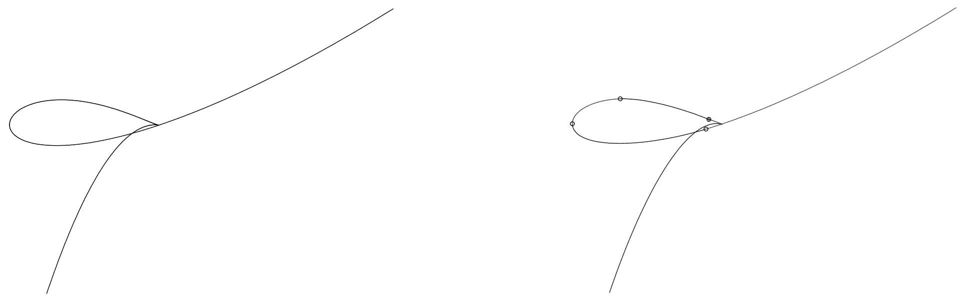

Fig. 13. c_2 and its approximate splines

Error Analyze of Curve c_4

	s_0	s_1	s_2
Range	(-1,-0.01)	(-0.01,0.01)	(0.01,1.466996)
Error	.009271	.000085	1.675201
Weight	.865405	.005014	11.729589

	s_3	s_{20}	s_{21}
Range	(1.466996,9/4)	(0.01,.973527)	(.973527,1.466995)
Error	.063562	.002565	0.110261
Weight	.484975	.700975	3.251226

Table 4

5. Conclusion and Future Work

We describle an algoritms to give an approximate implicitization method for a planar curve. It mainly involves a computation to solve two nonlinear equations for each segment of the approximation splines. With this method, we not only keep the shape but also the direction of the parametric curve with simple computation using quadratic curves. It's a chanleging problem for us to extend this method so that it can give an approximate parametrization for an implicit curve even to a surface.

References

[1] Bajaj, C., Local Parameterization, Implicitization and Inversion of Real Algebraic Curves, Tech. Rep. 89-863, Dept. of Computer Science, Purdue University, West Lafayette, Ind, 1989.

[2] Chuang, J.H. and Hoffmann, C.M., On local implicit approximation and its application, ACM Trans. Graphics 8, 4 ,298-324, 1989.

[3] Dokken, T., Approximate implicitization, in *Mathematical Methods in CAGD*, Lyche T. , L.L. Schumaker(eds), Vanderbilt University Press, Nashville & London, 2001.

[4] Farin, G., Curvature continuity and offsets for piecewise conics. *ACM Trans. on graphics*, 8, 2, 89-99, 1989.

[5] Farin, G., Curves and surfaces for Computer Aided Geometric Design: A Practical Guide, 4th edt. Academic Press, 1997.

[6] Fudos I. and Hoffmann, C.M., Constraint-based parametric conics for CAD, *Computer-Aided Design*, 28(2), pp91-100,1996.

[7] Juttler, B., Schicho, J., Shalaby M., Spline Implicitization of Planar Curves, talk at *ADG'2002*, Linz, Auatria, October, 2002.

[8] Lee,E. T. Y., The rational Bézier representation for conics, Geometric Modeling: Algorithm and New Trends (G Garin,Ed.),pp3-19,SIAM,Philadelphia,1985.

[9] Li, M., Algebraic curves and surfaces blending with given normal direction, master thesis, JiLin University, 2001.

[10] de Montaudouin, Y., Tiller, W. and Vold., H., Application of power series in computerional geometry. Computer-Aided Design, 18(10):93-108, 1986.

[11] Pottmann, H., Leopoldseder, S. Hofer, M., Approximation with Active B-spline Curves and Surfaces, *Pacific Graphics 2002 Proceedings*, Coquillart S. H.Y. Shum, S.M. Hu (eds), 8-25, IEEE computer Society Los Alamitos, CA, 2002.

[12] Sederberg, T.W., Zheng, J. Klimaszewski, K. and Dokken, T., Approximate Implicitization Using Monoid Curves and Surfaces, *Graphical Models and Images processing*, 61,177-198, 1999.

[13] Sederberg, T.W. and Zheng, J., Algebraic Methods for Computer Aided Geometric Design, in *Hanndbook of Computer Aided Geometric Design*, eds. G. Farin, J. Hoschek and M-S Kim, North-Holland, Amsterdam, 2002.