Proceedings of the 6th WSEAS International Conference on Robotics, Control and Manufacturing Technology, Hangzhou, China, April 16-18, 2006 (pp265-270)

Cubic Polynomial Curves with a Shape Parameter

MO GUOLIANG, ZHAO YANAN
Information and Computational Science
Zhejiang University City College
Huzhou Street, Hangzhou, 310015
PEOPLE'S REPUBLIC OF CHINA

Abstract: A class of cubic polynomial blending functions with a shape parameter is presented. It is an extension of cubic uniform B-spline basis functions. Piecewise polynomial curves with a shape parameter are constructed from these blending functions. The generated curves have second geometric continuity for any fixed shape parameter and have the same terminal properties as the cubic uniform B-spline curves. If the value of the shape parameter is changed, the approaching degree of the curves to their control polygon is adjusted accordingly and the curves are manipulated to approximate the cubic uniform B-spline curve from its both sides. In comparison with the existing results, the degree of blending functions is lower and the domain of the shape parameter is larger in this paper. A new method using cubic polynomial curves with a shape parameter is also proposed to solve interpolation problem without solving global systems of equations. Finally some computing examples of the curve design are also given.

Key-Words: Blending function, Shape parameter, GC^2 continuous, Cubic spline, Singular

1 Introduction

As we all know, polynomial B-spline curves, especially quadratic and cubic B-spline curves, are widely used in computer aided geometric design (CAGD). Piecewise continuous curves with four control points for each curve segment are flexible and can be used conveniently [1]. To improve the shape of a curve or the extent which a curve approaches its control polygon, the method of constructing curves by using tension parameter was presented in CAGD [2-4]. The weights of rational Bézier curves and rational B-spline curves have the function to adjust the shapes of curves [1,5]. The shapes of some other high-order rational curves can also be adjusted to meet different needs [6,7].

It is good to provide a more utilizable method of generating curves with a shape parameter. A C-B spline curve [8] is one of the curves with a shape parameter, and its basis functions contain triangle functions. The shape parameter is represented as triangle functions, but using triangle functions is not as easy as using polynomial functions. When values of the shape parameter are changed, C-B spline curves are away from the control polygon. Quadratic and cubic trigonometric polynomial curves with a shape parameter were proposed successively [9,10]. The range of the shape parameter is the interval $[-1,1]$. Another method of generating a cubic uniform B-spline curve with a shape parameter was introduced [11]. The degree of its four blending functions is 4, and the blending functions are not basis functions. The parameter value can change on the interval $[-8,1]$. The curve generated is C^2 continuous, but it can't approximate its control polygon arbitrarily. The method was also generalized to nth$(n \ge 1)$ order uniform B-spline curve case.

Analogously to the cubic B-spline curves, the purpose of this paper is to present practical piecewise polynomial curves. The method of constructing a cubic polynomial curve with a shape parameter is obtained, and the degree of its four blending functions is 3. The value of its shape parameter can change on the interval $[-8,+\infty)$. The curve generated by this method is GC^2 continuous, and it can approximate its control polygon arbitrarily by increasing the value of its shape parameter. The curve is an extension of a cubic uniform B-spline curve. It has some good properties as a cubic uniform B-spline curve does. This paper also presents new interpolation method that can produce C^2 continuous cubic polynomial curves with a shape parameter without solving global systems of equations [12-14], while providing slackness control capabilities. With the low-degree polynomials and direct computation of control vertices, the method is computationally simple, and thus useful for interactive interpolation shape design and computer graphics applications.

The paper is organized as follows. In Section 2, the blending functions are proposed and the properties are described. In Section 3, the cubic polynomial curve with a shape parameter is given. Open and closed piecewise cubic polynomial curves are described. The approximation of the GC^2 continuous polynomial curve to a cubic B-spline curve and to the given control polygon are shown in Section 4. A new interpolation method is also described in Section 4. Some conclusions can be drawn in Section 5.

2 Cubic basis functions with a shape parameter

Definition 1. Supposed $t \in [0,1], \lambda \in \mathrm{R}$, the associated blending functions are defined to be the following functions:

$$\begin{cases} \phi_0(t;\lambda) = \dfrac{2}{12+\lambda}(1-t)^3, \\ \phi_1(t;\lambda) = \dfrac{1}{12+\lambda}\left[2(3+\lambda)t^3 - 3(4+\lambda)t^2 + 8 + \lambda\right], \\ \phi_2(t;\lambda) = \dfrac{1}{12+\lambda}\left[-2(3+\lambda)t^3 + 3(2+\lambda)t^2 + 6t + 2\right], \\ \phi_3(t;\lambda) = \dfrac{2}{12+\lambda}t^3, \end{cases}$$

$t \in [0,1]$.

Theorem 1. The blending functions have the following properties:

(1) $\phi_0(t;\lambda) = \phi_3(1-t;\lambda), \phi_1(t;\lambda) = \phi_2(1-t;\lambda)$.

(2) $\sum_{j=0}^{3} \phi_j(t;\lambda) = 1, t \in [0,1], \lambda \ge -8$.

(3) $\phi_j(t;\lambda) \ge 0, (j = 0,1,2,3), t \in [0,1], \lambda \ge -8$.

(4)

$$\begin{pmatrix} \phi_0(t;\lambda) \\ \phi_1(t;\lambda) \\ \phi_2(t;\lambda) \\ \phi_3(t;\lambda) \end{pmatrix} = \frac{1}{12+\lambda}\begin{pmatrix} 2 & 0 & 0 & 0 \\ 8+\lambda & 8+\lambda & 4 & 2 \\ 2 & 4 & 8+\lambda & 8+\lambda \\ 0 & 0 & 0 & 2 \end{pmatrix}\begin{pmatrix} B_0^3(t) \\ B_1^3(t) \\ B_2^3(t) \\ B_3^3(t) \end{pmatrix}$$

$B_j^3 = \binom{3}{j}(1-t)^{3-j}t^j\,(j = 0,1,2,3)$ are Bernstein polynomial basis functions.

Proof:

(i) Property (1), (2) and (4) are straightforward.

(ii)Property (3) is proved according to (4). For any j, $\phi_j(t;\lambda)$ can be represented as convex linear combinations of $B_i^3(t)(i = 0,1,2,3)$, and the combination coefficients are non-negative, so $\phi_j(t;\lambda)\ (j = 0,1,2,3)$ are non-negative.

The proof is completed.

If $\lambda = 0$, $\phi_j(t;0)(j = 0,1,2,3), t \in [0,1]$ are the basis functions of cubic uniform B-spline curve on the interval $[0,1]$. The blending functions are the generalized basis functions of cubic uniform B-spline. Fig. 1 shows the images of the basis functions for different shape parameters.

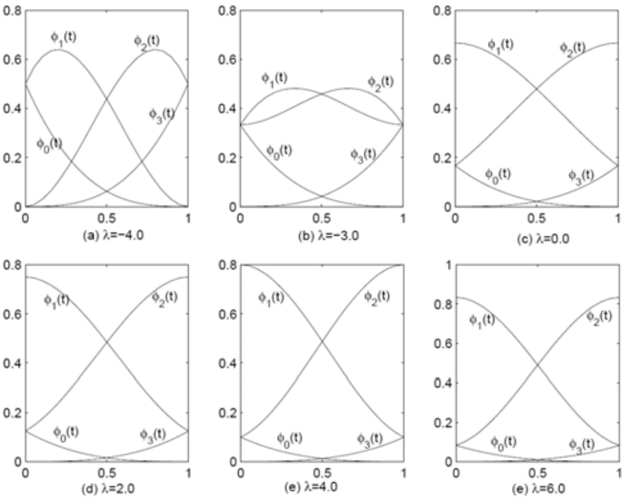


Fig.1 The image of the basis functions with different shape parameters λ

3 Cubic spline curve with a shape parameter

Definition 2. Given control points $\boldsymbol{P}_i(i = 0,1,\cdots,n)$, $(n \ge 3)$ a knot vector $t_0 < t_1 < \cdots < t_{n+1}$, and a parameter $\lambda(\lambda \ge -4)$, then for any $t \in [t_k, t_{k+1}]$, $k = 3,4,\cdots,n$, the curve segment of polynomial is defined as follows:

$$\boldsymbol{C}_k(t;\lambda) = \sum_{i=0}^{3} \boldsymbol{P}_{i+k-3}\phi_i\left(\frac{t-t_k}{t_{k+1}-t_k};\lambda\right).$$

The piecewise cubic polynomial curve is defined as follows:

$$\boldsymbol{C}(t;\lambda) = \boldsymbol{C}_k(t;\lambda), t \in [t_k, t_{k+1}], k = 3,4,\cdots,n,$$

The curve $\boldsymbol{C}(t;\lambda)$ is a piecewise cubic polynomial curve with a shape parameter defined on the interval $[t_3, t_{n+1}]$, and it is generalized from the cubic uniform B-spline curve. For any $t \in [t_k, t_{k+1}), k = 3,4,\cdots,n$, the curve segment lies in the convex hull H_k of $\boldsymbol{P}_{k-3}, \boldsymbol{P}_{k-2}, \boldsymbol{P}_{k-1}, \boldsymbol{P}_k$. The total curve $\boldsymbol{C}(t;\lambda)$ lies in $\bigcup_{k=3}^{n} H_k$. The shape of $\boldsymbol{C}(t;\lambda)$ can be adjusted when the parameter value λ is changed.

For $t \in [t_k, t_{k+1}), k = 3, 4, \cdots, n$, $\boldsymbol{C}_k(t;\lambda)$ can be transformed into a Bézier curve with its variable parameter $u = \dfrac{t - t_k}{t_{k+1} - t_k}$ and its control points are $\boldsymbol{b}_{k-3}, \boldsymbol{b}_{k-2}, \boldsymbol{b}_{k-1}, \boldsymbol{b}_k$, where

$$\begin{pmatrix} \boldsymbol{b}_{k-3} \\ \boldsymbol{b}_{k-2} \\ \boldsymbol{b}_{k-1} \\ \boldsymbol{b}_k \end{pmatrix} = \frac{1}{12+\lambda} \begin{pmatrix} 2\boldsymbol{P}_{k-3} + (8+\lambda)\boldsymbol{P}_{k-2} + 2\boldsymbol{P}_{k-1} \\ (8+\lambda)\boldsymbol{P}_{k-2} + 4\boldsymbol{P}_{k-1} \\ 4\boldsymbol{P}_{k-2} + (8+\lambda)\boldsymbol{P}_{k-1} \\ 2\boldsymbol{P}_{k-2} + (8+\lambda)\boldsymbol{P}_{k-1} + 2\boldsymbol{P}_k \end{pmatrix}$$

So de Casteljau corner cutting algorithm of a Bézier curve can be applied to the computation of the curve $\boldsymbol{C}(t;\lambda)$ for speed and stable computation.

Theorem 2. The curve $\boldsymbol{C}(t;\lambda)$ has 2nd geometric continuity, i.e., it is a GC^2 continuous curve.

Proof: Let $h_k = t_{k+1} - t_k, k = 3, 4, \cdots, n.$ Straightforward computation is shown as:

$$\boldsymbol{C}_k(t_k+;\lambda) = \frac{1}{12+\lambda}\left[2\boldsymbol{P}_{k-3} + (8+\lambda)\boldsymbol{P}_{k-2} + 2\boldsymbol{P}_{k-1}\right],$$

$$\boldsymbol{C}_k(t_{k+1}-;\lambda) = \frac{1}{12+\lambda}\left[2\boldsymbol{P}_{k-2} + (8+\lambda)\boldsymbol{P}_{k-1} + 2\boldsymbol{P}_{k-2}\right],$$

$$\boldsymbol{C}'_k(t_k+;\lambda) = \frac{6}{h_k(12+\lambda)}\left(\boldsymbol{P}_{k-1} - \boldsymbol{P}_{k-3}\right),$$

$$\boldsymbol{C}'_k(t_{k+1}-;\lambda) = \frac{6}{h_k(12+\lambda)}\left(\boldsymbol{P}_k - \boldsymbol{P}_{k-2}\right),$$

$$\boldsymbol{C}''_k(t_k+;\lambda) = \frac{6}{h_k^2(12+\lambda)}\left[2\boldsymbol{P}_{k-3} - (4+\lambda)\boldsymbol{P}_{k-2} + (2+\lambda)\boldsymbol{P}_{k-1}\right],$$

$$\boldsymbol{C}''_k(t_{k+1}-;\lambda) = \frac{6}{h_k^2(12+\lambda)}\left[(2+\lambda)\boldsymbol{P}_{k-2} - (4+\lambda)\boldsymbol{P}_{k-1} + 2\boldsymbol{P}_k\right].$$

For $k = 4, 5, \cdots, n$, the following can be obtained

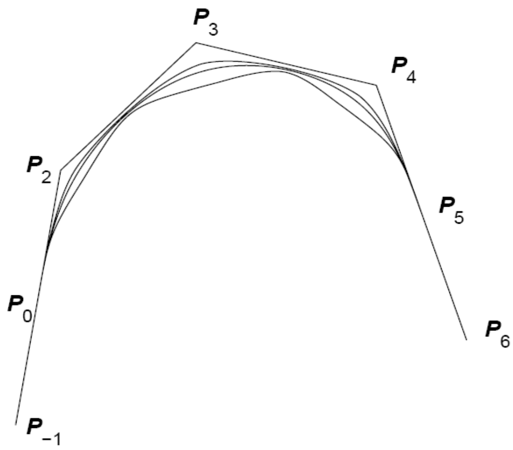
$$\boldsymbol{C}(t_k+;\lambda) = \boldsymbol{C}(t_k-;\lambda),$$

$$\boldsymbol{C}'(t_k+;\lambda) = \frac{h_{k-1}}{h_k}\boldsymbol{C}'(t_k-;\lambda),$$

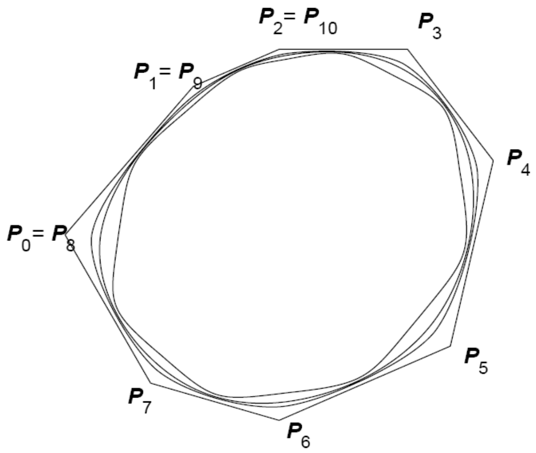
$$\boldsymbol{C}''(t_k+;\lambda) = \left(\frac{h_{k-1}}{h_k}\right)^2 \boldsymbol{C}''(t_k-;\lambda) + \lambda\frac{h_{k-1}}{h_k^2}\boldsymbol{C}'(t_k;\lambda).$$

According to the proof of Theorem 2, the curve $\boldsymbol{C}(t;\lambda)$ and the cubic uniform B-spline curve have similar properties on the endpoint location and endpoint tangent vector.

If let $\boldsymbol{P}_{-1} = 2\boldsymbol{P}_0 - \boldsymbol{P}_1$ and $\boldsymbol{P}_{n+1} = 2\boldsymbol{P}_n - \boldsymbol{P}_{n-1}$ and $\boldsymbol{P}_{-1}\boldsymbol{P}_0 \cdots \boldsymbol{P}_n\boldsymbol{P}_{n+1}$ be the control polygon of the curve $\boldsymbol{C}(t;\lambda)$, then $\boldsymbol{P}_0$ and $\boldsymbol{P}_n$ are its start and terminal points respectively. $\dfrac{12}{12+\lambda}\left(\boldsymbol{P}_1 - \boldsymbol{P}_0\right)$ and $\dfrac{12}{12+\lambda}\left(\boldsymbol{P}_n - \boldsymbol{P}_{n-1}\right)$ are its tangent vectors at $\boldsymbol{P}_0$ and $\boldsymbol{P}_n$ respectively. If let $\boldsymbol{P}_n = \boldsymbol{P}_0$, $\boldsymbol{P}_{n+1} = \boldsymbol{P}_1$, $\boldsymbol{P}_{n+2} = \boldsymbol{P}_n$ and $\boldsymbol{P}_0\boldsymbol{P}_1 \cdots \boldsymbol{P}_{n+1}\boldsymbol{P}_{n+2}$ be the control polygon of the curve $\boldsymbol{C}(t;\lambda)$, then the curve $\boldsymbol{C}(t;\lambda)$ is a closed curve. Fig.2 depicts open and closed cubic polynomial curves for $\lambda = -1, 0, 1$.



(a) from the bottom to the top



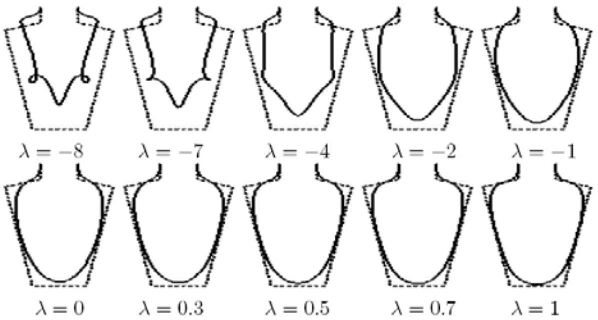
(b) from the inner to the outter

Fig.2 Open and closed GC^2 continuous cubic spline curves with $\lambda = -1, 0, 1$

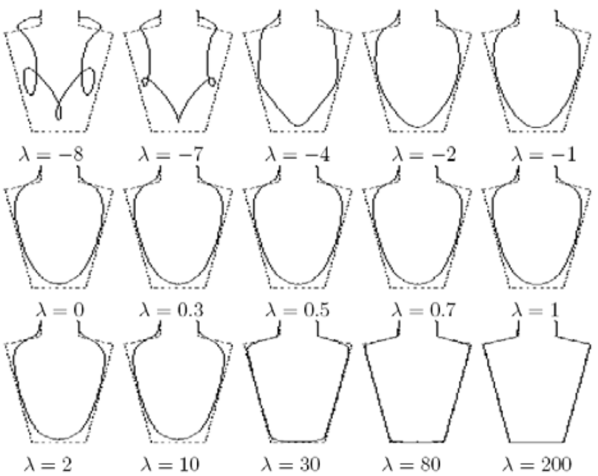
In Fig.2, the curve $\boldsymbol{C}(t;\lambda)$ gradually approximates its control polygon if the shape parameter λ gradually increases. The property is common. In fact, if t is fixed, $\phi_0(t)$ and $\phi_3(t)$ are monotone decreasing with respect to λ, so for any $\boldsymbol{C}_k(t;\lambda)$, the weights of points $\boldsymbol{P}_{k-3}$ and $\boldsymbol{P}_k$ become smaller if λ becomes larger. The curve segment $\boldsymbol{C}_k(t;\lambda)$ approximates the line segment $\boldsymbol{P}_{k-2}\boldsymbol{P}_{k-1}$ because the weights of points $\boldsymbol{P}_{k-2}$ and $\boldsymbol{P}_{k-1}$ increase.

4 Application of cubic spline curve with a shape parameter

λ can be used as a shape parameter in order to construct curves locating at different positions. The range for the shape parameter in this paper is the interval $[-8,+\infty)$. In References [11], the shape parameter is the interval $[-8,1]$. The curves generated by the two methods can approximate their control polygon when the values of their shape parameter increase. So our method has more abundant modeling functions than that in References [11]. It is fatal that the blending functions constructed in References [11] are degree 4 and not basis functions. Fig. 3 and Fig. 4 show examples of curve design.

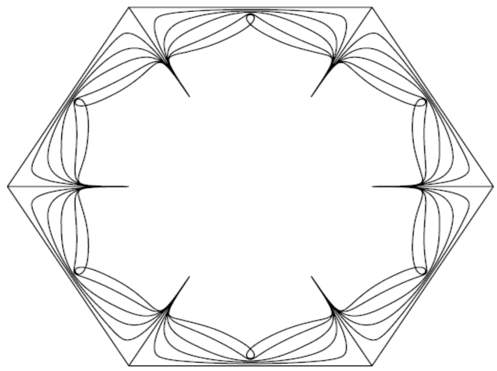


(a) Using method in Reference [11]

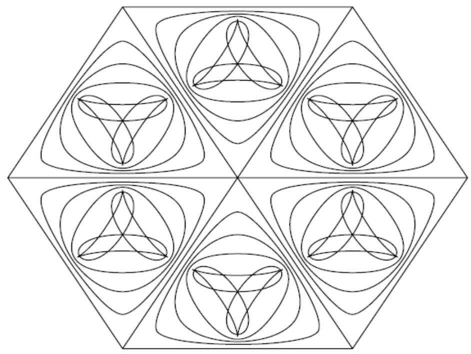


(b) Using our method

Fig.3 Open GC^2 continuous cubic spline curves interpolating endpoints with different λ



(a) petaling



(b) flower pattern

Fig.4 Open and closed GC^2 continuous cubic spline curves with $\lambda = -3, -2.5, -1.5, 0, 4$

If $\lambda \in [-4,0]$, the curve $\boldsymbol{C}(t;\lambda)$ approaches more to the cubic uniform B-spline having the same control polygon with $\boldsymbol{C}(t;\lambda)$. If $\lambda \in [0,+\infty)$, the curve approaches more to the control polygon of $\boldsymbol{C}(t;\lambda)$.

The interpolating problem is considered in the following. Given a set of data points $\{\boldsymbol{P}_i\}_{i=0}^{n} \in \mathrm{R}^3, n \geq 4$, our goal is to construct a low-degree interpolating polynomial curve without solving any global system to find its control vertices and its global slackness controlled. In order to interpolate the two endpoints, we introduce two auxiliary data points $\boldsymbol{P}_{-1} = 2\boldsymbol{P}_0 - \boldsymbol{P}_1$ and $\boldsymbol{P}_{n+1} = 2\boldsymbol{P}_n - \boldsymbol{P}_{n-1}$ if the control polygon is open $(\boldsymbol{P}_0 \neq \boldsymbol{P}_n)$, or $\boldsymbol{P}_{-1} = \boldsymbol{P}_{n-1}$ and $\boldsymbol{P}_{n+1} = \boldsymbol{P}_1$ if the control polygon is closed $(\boldsymbol{P}_0 = \boldsymbol{P}_n)$. The knot vector is obtained by accumulating chord-length.

The cubic singular blending function [14] is defined to be

$$S_k(t)=\begin{cases}\dfrac{9}{2}\left(\dfrac{t-t_k}{t_{k+1}-t_k}\right)^3, t_k \le t \le \dfrac{2}{3}t_k+\dfrac{1}{3}t_{k+1}, \\ \dfrac{9}{2}\left[\left(\dfrac{t-t_k}{t_{k+1}-t_k}\right)^3-3\left(\dfrac{t-t_k}{t_{k+1}-t_k}-\dfrac{1}{3}\right)^3\right], \\ \dfrac{2}{3}t_k+\dfrac{1}{3}t_{k+1} \le t \le \dfrac{1}{3}t_k+\dfrac{2}{3}t_{k+1}, \\ 1-\dfrac{9}{2}\left(\dfrac{t_{k+1}-t}{t_{k+1}-t_k}\right)^3, \dfrac{1}{3}t_k+\dfrac{2}{3}t_{k+1} \le t \le t_{k+1};\end{cases}$$

$$t_k < t_{k+1}, k=2,3,\cdots,n,n+1.$$

It is easy to prove that $S_k(t)$ are singular functions satisfying that

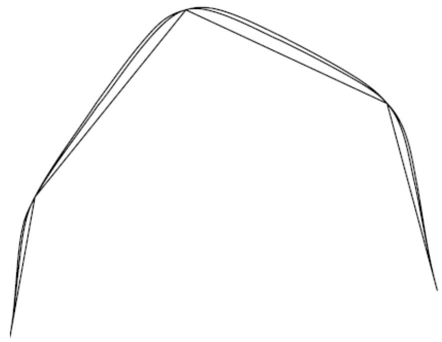
$$S_k(t_k)=0, S_k(t_{k+1})=1,$$
$$S_k'(t_k)=S_k''(t_k)=0,$$
$$S_k'(t_{k+1})=S_k''(t_{k+1})=0.$$

The cubic α -spline curve is defined to be

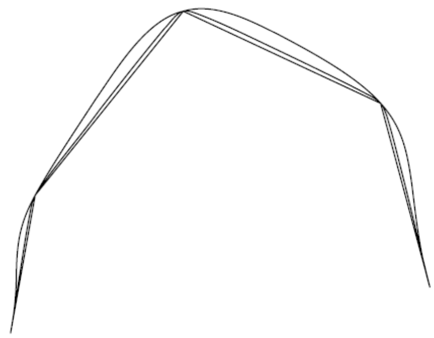
$$\begin{aligned}&\boldsymbol{Q}_k(t;\lambda,\alpha)\\ &=(1-\alpha)\boldsymbol{C}_k(t;\lambda)+\left(1-S_k(t)\right)\boldsymbol{P}_{k-2}+S_k(t)\boldsymbol{P}_{k-1}\\ &\quad-\frac{1-\alpha}{6+\lambda}\left[\boldsymbol{P}_{k-3}+\left(4+\lambda\right)\boldsymbol{P}_{k-2}+\boldsymbol{P}_{k-1}\right]\\ &+\frac{S_k(t)\left(1-\alpha\right)}{6+\lambda}\left[\boldsymbol{P}_{k-3}+(3+\lambda)\boldsymbol{P}_{k-2}-(3+\lambda)\boldsymbol{P}_{k-1}-\boldsymbol{P}_k\right],\\ &t_k \le t \le t_{k+1}, k=2,3,\cdots,n+1.\end{aligned}$$

$$\boldsymbol{Q}(t;\lambda,\alpha)=\boldsymbol{Q}_k(t;\lambda,\alpha), t_k \le t \le t_{k+1}, k=2,3,\cdots,n+1,$$

It is clear that, for any blending factor α, the resulting curve $\boldsymbol{Q}(t;\lambda,\alpha)$ is GC^2 continuous and $\boldsymbol{Q}(t_k;\lambda,\alpha)=\boldsymbol{P}_{k-2}, k=2,3,\cdots,n+2.$ So the constructed curve $\boldsymbol{Q}(t;\lambda,\alpha)$ interpolates the data points $\{\boldsymbol{P}_i\}_{i=0}^{n}$. For the purpose of practical shape modeling, we shall restrict $\alpha \in [0,1]$, and call these curves the standard α cubic spline curve with a shape parameter λ. If $\alpha=0$, the α cubic spline curve is reduced to the cubic spline curve $\boldsymbol{C}(t;\lambda)$. The blending factor α can be used to control the slackness of the α cubic spline curve $\boldsymbol{Q}(t;\lambda,\alpha)$. Note that changing the blending factor α does not affect the continuity of $\boldsymbol{Q}(t;\lambda,\alpha)$ and the interpolation features at the data points. Thus there are two parameters that can influence the shape of the cubic spline curve. Fig.5 and Fig.6 describe open and closed α cubic spline curves with two different parameters.

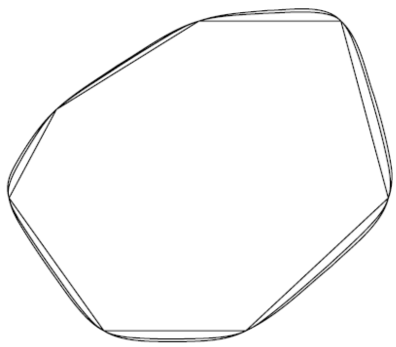


(a) α=0.5, from the top to the bottom,λ=−1.5,0

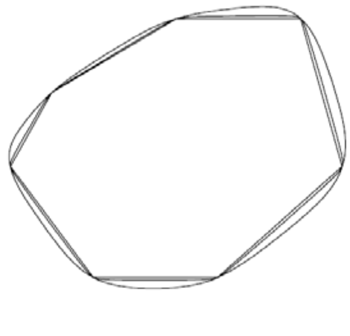


(b) λ=0, from the top to the bottom,α=0.2,0.8

Fig.5 Open curves interpolating data points with different shape parameters and blending factors



(a) α=0.5, from the outer to the inner λ=−1.5,0



(b) λ=0, from the outer to the inner α=0.2,0.8

Fig.6 Closed curves interpolating data points with different shape parameters and blending factors

5 Conclusions

In this paper, a type of cubic piecewise polynomial curve with a shape parameter is presented. The cubic uniform B-spline curve is its special case, and this cubic curve can locate different places on both sides of the cubic uniform B-spline curve. By changing the value of the shape parameter, we can adjust the approaching degree of the curves to their control polygon and manipulate the curves to approximate the cubic uniform B-spline curves from both sides. The curve is GC^2 continuous, and it can arbitrarily approximate its control polygon by increasing the value of its shape parameter. It can also be used to solve an interpolation problem. The constructed α cubic spline curve interpolates the given data points. Some computing examples show that the method of constructing curves has important use value.

References:

[1] J. Hoschek, D. Lasser, *Fundamentals of Computer Aided Geometric Design*, Wellesley, MA (translated by L.L. Schumaker), 1993.

[2] B. A. Barsky, *Computer graphics and geometric modelling using Beta-splines*, Springer-Verlag, Hei-delberg, 1988.

[3] J. A. Gregory, M. Sarfraz, A rational cubic spline with tension, *Computer Aided Geometric Design*, Vol.7, 1990, pp.1-13.

[4] B. Joe, Multiple-knot and rational cubic Beta-splines, *ACM Transactions on Graphics*, Vol.8, No.2, 1989, pp.100-120.

[5] L. Piegl, W. Tiller, *The NURBS Book*, Springer, New York, 1995.

[6] T. N. T. Goodman, Constructing piecewise rational curves with Frenet frame continuity, *Computer Aided Geometric Design*, Vol.7, 1990, pp.15-31.

[7] B. Joe, Quartic Beta-splines, *ACM Transactions on Graphics*, Vol.9, No.3, 1990, pp.301-337.

[8] Jiwen Zhang, C-curves: An extension of cubic curves, *Computer Aided Geometric Design*, Vol.13, No.3, 1996, pp.199-217.

[9] Xuli Han, Quadratic trigonometric polynomial curves with a shape parameter, *Computer Aided Geometric Design*, Vol.19, No.7, 2002, pp. 503-512.

[10] Xuli Han, Cubic trigonometric polynomial curves with a shape parameter, *Computer Aided Geometric Design*, Vol.21, No.6, 2004, pp.535-548.

[11] Xuli Han, Shengjun Liu, An extension of the cubic uniform B-spline curves, *Journal of Computer-Aided Design & Computer Graphics*, Vol.15, No.5, 2003, pp.576-578.

[12] B. A. Barsky, D. P. Greenberg, Determining a set of B-spline control vertices to generate an interpolating surface, *Computer Graph and Image Process*, Vol.14, No.3, 1980, pp.203-226.

[13] K. F. Loe, α -B-spline: a linear singular blending B-spline. *The Visual Computer*, Vol.12, 1996, pp.18-25.

[14] C.L. Tai, G.J. Wang, Interpolation with slackness and continuity control and convexity-preservation using singular blending. *Journal of Computational & Applied Mathematics*, Vol.172, No.2, 2004, pp.337-361.