Constructing iterative non-uniform B-spline curve and surface to fit data points

Article *in* Science in China Series F Information Sciences · June 2004

DOI: 10.1360/02yf0529 · Source: DBLP

Hongwei Lin
Zhejiang University

Constructing iterative non-uniform B-spline curve and surface to fit data points

LIN Hongwei, WANG Guojin & DONG Chenshi

Department of Mathematics, Zhejiang University, Hangzhou, China, 310027; State Key Laboratory of CAD&CG, Zhejiang University, Hangzhou 310027, China

Correspondence should be addressed to Wang Guojin (email: wgj@math.zju.edu.cn)

Received May 6, 2003

Abstract In this paper, based on the idea of *profit and loss modification*, we present the iterative non-uniform B-spline curve and surface to settle a key problem in computer aided geometric design and reverse engineering, that is, constructing the curve (surface) fitting (interpolating) a given ordered point set without solving a linear system. We start with a piece of initial non-uniform B-spline curve (surface) which takes the given point set as its control point set. Then by adjusting its control points gradually with iterative formula, we can get a group of non-uniform B-spline curves (surfaces) with gradually higher precision. In this paper, using modern matrix theory, we strictly prove that the limit curve (surface) of the iteration interpolates the given point set. The non-uniform B-spline curves (surfaces) generated with the iteration have many advantages, such as satisfying the NURBS standard, having explicit expression, gaining locality, and convexity preserving, etc.

Keywords: fitting, iteration, non-uniform B-spline curve and surface, convexity preserving.

DOI: 10.1360/02yf0529

There are lots of problems on interpolating or fitting with B-spline in Computer Aided Design (CAD) and reverse engineering. In order to get the B-spline curve (surface) interpolating a given ordered point set, a linear system usually needs to be solved for calculating the vertices of its control polygon (net)[1,2]. Although such curves (surfaces) can interpolate the given point set, there are some disadvantages of the method. First, because the linear system depends only on the interpolated points, and contains no convexity-preserving condition, it cannot be guaranteed that the interpolating curve (surface) calculated with the method is convexity preserving. Second, the interpolating curve (surface) cannot be explicitly expressed with the given point set, so it loses the explicit expression held by the original B-spline. Third, the method of solving a linear system is global, and modification only to one interpolated point will lead to resolving the linear system and then modifying the whole interpolating curve (surface); hence it also loses the advantage of locality held by B-spline curve (surface).

In order to overcome the disadvantages of the method, which solves a linear system, lots of scholars have done their best to search for a method for calculating the interpolating curve (surface) directly. Loe[3] proposed a method which constructs an α-B-spline curve (surface) with a kind of singular blending function. It is not necessary for this method to solve a linear system for the control points. But, in fact, it is a kind of generalized B-spline[4—6], and does not satisfy the NURBS (Non-Uniform Rational B-Spline) standard. Because the NURBS standard is adopted in CAD system currently, it is more significant to search for a method that gets interpolating (fitting) B-spline curve (surface) without solving the global linear system. Early in 1975, a Chinese mathematician, Prof. Qi[7], presented the idea of *profit and loss modification* for spline fitting. After four years, a very famous American expert in function theory, Prof. de Boor[8], also set forth the idea in an invited lecture independently. They happened to have the same view. In 1991, Qi[9] briefly published the results discussed with de Boor at the beginning of the 1980s. That is, he presented the iterative format for uniform B-spline curve and listed the outlines of proof for its convergence. But he did not propose the convexity-preserving algorithm and give the proof at the same time. More importantly, in shape design area, the more commonly used geometric model is non-uniform B-spline surface, but not the uniform B-spline curve. So, at the current era in which the CAD technology is developing with high speed, the research for the convergence and convexity-preserving property of the iterative non-uniform B-spline curve (surface) is an urgent matter. It has important significance to the reverse engineering.

In reserve engineering, for a given ordered point set, a group of fitting curves (surfaces) with different precisions often needs to be calculated. The problem cannot be settled with the method of solving global linear system, α-B-spline method, or the least square method. In this paper, based on the *profit and loss modification* idea, we present the iterative non-uniform B-spline curve (surface), which settles the problem successfully. The so-called iterative non-uniform B-spline curve (surface), starts with a piece of initial non-uniform B-spline curve (surface) taking the given ordered point set as its control point set, and by gradually adjusting the positions of its control points with iterative formula, finally approaches the given point set step by step. By means of modern matrix theory, we strictly prove that the sequence of the iterative non-uniform B-spline curves (surfaces) converges to the curve (surface) which interpolates the given point set. Altering the iterative format slightly, we also prove that the limit curve is convexity preserving. Compared with previous methods, the iterative non-uniform B-spline curve (surface) method obviously holds the advantages as follows. First, without solving the global system for calculating the control points, we can get the fitting curve (surface) satisfying NURBS standard and holding advantages such as explicit expression and locality, etc. Second, the method is convexity preserving. Third, with iterative non-uniform B-spline curve (surface), we settle the problem in our work. That is, fitting a given ordered point set using a group of curves (surfaces) with different precisions.

1 Iterative non-uniform B-spline curve

1.1 Deducing of the iterative format

We want to design an iterative sequence of non-uniform B-spline curves for fitting a given ordered point set. Each curve of the sequence has the same non-uniform B-spline bases defined on the same knot vector, while its control points are adjusted according to an iterative format. The limit curve of the iterative sequence interpolates the given point set, when the iterative time tends to infinity.

Suppose that an ordered point set $\{\boldsymbol{P}_i\}_{i=1}^{n} \in \mathbb{R}^3$ has been given. In the following iteration, the non-uniform B-spline base $B_i^4(t)$ is defined on the knot vector $\{t_i\}_{i=0}^{n+5}$, here

$$t_i = \sum_{j=2}^{i-2} \left\| \boldsymbol{P}_j - \boldsymbol{P}_{j-1} \right\|, \ i = 4,\ 5,\ \cdots,\ n+2, \quad t_0 = t_1 = t_2 = t_3 = 0,\ t_{n+2} = t_{n+3} = t_{n+4} = t_{n+5}. \tag{1}$$

At the beginning of the iteration, let

$$\boldsymbol{P}_i^0 = \boldsymbol{P}_i,\ i = 1,\ 2,\ \cdots,\ n; \quad \boldsymbol{P}_0^0 = \boldsymbol{P}_1^0,\ \boldsymbol{P}_{n+1}^0 = \boldsymbol{P}_n^0. \tag{2}$$

First, with $\{\boldsymbol{P}_i^0\}_{i=0}^{n+1}$ as the control point set, we construct a piece of degree 3 non-uniform B-spline curve

$$\boldsymbol{r}^0(t) = \sum_{i=0}^{n+1} \boldsymbol{P}_i^0 B_i^4(t), \qquad t \in [t_3,\ t_{n+2}]. \tag{3}$$

Further, write the first adjusting vector of the j-th control point as

$$\boldsymbol{\varDelta}_j^0 = \boldsymbol{P}_j - \boldsymbol{r}^0(t_{j+2}),\ \ j=1,\ 2,\ \cdots,\ n,$$

let

$$\boldsymbol{P}_j^1 = \boldsymbol{P}_j^0 + \boldsymbol{\varDelta}_j^0,\ j = 1,\ 2,\ \cdots,\ n; \quad \boldsymbol{P}_0^1 = \boldsymbol{P}_1^1,\ \boldsymbol{P}_{n+1}^1 = \boldsymbol{P}_n^1;$$

and take $\{\boldsymbol{P}_j^1\}_{j=0}^{n+1}$ as the control point set. We have

$$\boldsymbol{r}^1(t) = \sum_{i=0}^{n+1} \boldsymbol{P}_i^1 B_i^4(t), \qquad t \in [t_3,\ t_{n+2}],$$

the degree 3 non-uniform B-spline curve after the first iteration.

Similarly, if the non-uniform B-spline curve $\boldsymbol{r}^k(t)$ after the k-th ($k = 0,\ 1,\ \cdots$) iteration has been gotten, the (k+1)-th adjusting vector of the j-th control point can be written as

$$\varDelta_j^k = \boldsymbol{P}_j - \boldsymbol{r}^k(t_{j+2}),\ j=1,\ 2,\ \cdots,\ n. \tag{4}$$

Let

$$\boldsymbol{P}_j^{k+1} = \boldsymbol{P}_j^k + \varDelta_j^k,\ j=1,2,\ \cdots,n,\quad \boldsymbol{P}_0^{k+1}=\boldsymbol{P}_1^{k+1}, \boldsymbol{P}_{n+1}^{k+1}=\boldsymbol{P}_n^{k+1}; \tag{5}$$

and take $\{\boldsymbol{P}_j^{k+1}\}_{j=0}^{n+1}$ as control point set. We can get

$$\boldsymbol{r}^{k+1}(t) = \sum_{i=0}^{n+1} \boldsymbol{P}_i^{k+1} B_i^4(t), \qquad t \in [t_3, t_{n+2}], \tag{6}$$

the non-uniform B-spline curve (see fig. 1) after (k+1)-th iteration.

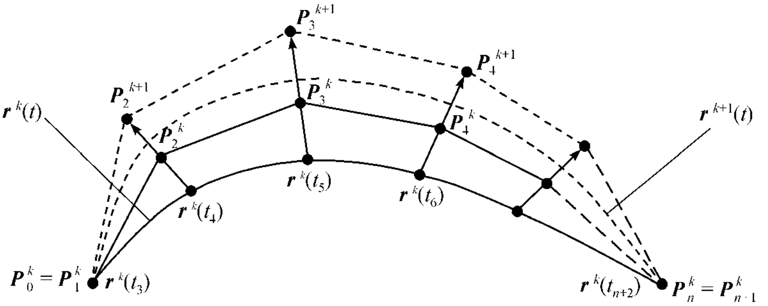

Fig. 1. Iterative non-uniform B-spline curve, from $\boldsymbol{r}^k(t)$ to $\boldsymbol{r}^{k+1}(t)$.

Such non-uniform B-spline curve sequence $\{\boldsymbol{r}^k(t)|k=1,\ 2,\ \cdots\}$is called the iterative non-uniform B-spline curve.

Because

$$\begin{aligned}
\varDelta_j^{k+1} &= \boldsymbol{P}_j - \boldsymbol{r}^{k+1}(t_{j+2}) = \boldsymbol{P}_j - \sum_{i=0}^{n+1}(\boldsymbol{P}_i^k + \varDelta_i^k)B_i^4(t_{j+2})\\
&= \boldsymbol{P}_j - (\boldsymbol{P}_{j-1}^k + \varDelta_{j-1}^k)B_{j-1}^4(t_{j+2}) - (\boldsymbol{P}_j^k + \varDelta_j^k)B_j^4(t_{j+2}) - (\boldsymbol{P}_{j+1}^k + \varDelta_{j+1}^k)B_{j+1}^4(t_{j+2})\\
&= (\boldsymbol{P}_j - \boldsymbol{r}^k(t_{j+2})) - \varDelta_{j-1}^k B_{j-1}^4(t_{j+2}) - \varDelta_j^k B_j^4(t_{j+2}) - \varDelta_{j+1}^k B_{j+1}^4(t_{j+2})\\
&= -B_{j-1}^4(t_{j+2})\varDelta_{j-1}^k + (1 - B_j^4(t_{j+2}))\varDelta_j^k - B_{j+1}^4(t_{j+2})\varDelta_{j+1}^k,\\
&\qquad k = 0,\ 1,\ \cdots;\ j = 2, 3,\ \cdots, n-1,
\end{aligned}$$

noting that $\varDelta_1^k = \varDelta_n^k = \boldsymbol{0}$, $k = 0,\ 1,\ \cdots$ for both the two extreme points of the interval [t_3, t_{n+2}] are four multiplicities, we have the iterative format of the adjusting vectors of the control point set in matrix

$$\left[\varDelta_2^{k+1}, \varDelta_3^{k+1}, \cdots, \varDelta_{n-1}^{k+1}\right]^{\mathrm{T}} = \boldsymbol{C}\left[\varDelta_2^k, \varDelta_3^k, \cdots, \varDelta_{n-1}^k\right]^{\mathrm{T}},\ \boldsymbol{C} = \boldsymbol{I}-\boldsymbol{B};\ k = 0,\ 1,\ \cdots; \tag{7}$$

here, $\boldsymbol{I}$ is n−2 rank identity matrix,

$$\boldsymbol{B}=\begin{bmatrix} B_2^4(t_4) & B_3^4(t_4) & & & & \\ B_2^4(t_5) & B_3^4(t_5) & B_4^4(t_5) & & & \\ & \ddots & \ddots & \ddots & & \\ & & B_{n-3}^4(t_n) & B_{n-2}^4(t_n) & B_{n-1}^4(t_n) \\ & & & B_{n-2}^4(t_{n+1}) & B_{n-1}^4(t_{n+1}) \end{bmatrix}. \tag{8}$$

1.2 Convergence of the iterative non-uniform B-spline curve

Suppose the tri-diagonal matrix

$$\boldsymbol{A}=\begin{bmatrix} a_1 & b_1 & & & \\ c_1 & a_2 & b_2 & & \\ & \ddots & \ddots & \ddots & \\ & & c_{n-2} & a_{n-1} & b_{n-1} \\ & & & c_{n-1} & a_n \end{bmatrix}\in \mathbb{R}_{n\times n},\quad b_ic_i>0,\ i=1,\ 2,\ \cdots,\ n-1, \tag{9}$$

sgn(.) is the sign function, and the real tri-diagonal symmetric matrix

$$\tilde{A}=\begin{bmatrix} a_1 & \operatorname{sgn}(b_1)\sqrt{b_1c_1} & & & \\ \operatorname{sgn}(c_1)\sqrt{b_1c_1} & a_2 & \operatorname{sgn}(b_2)\sqrt{b_2c_2} & & \\ \ddots & \ddots & \ddots & & \\ & & \operatorname{sgn}(c_{n-2})\sqrt{b_{n-2}c_{n-2}} & a_{n-1} & \operatorname{sgn}(b_{n-1})\sqrt{b_{n-1}c_{n-1}} \\ & & & \operatorname{sgn}(c_{n-1})\sqrt{b_{n-1}c_{n-1}} & a_n \end{bmatrix}. \tag{10}$$

Then we have three lemmas as follows.

Lemma 1[10]. The matrix $\boldsymbol{A}$ is similar to the matrix $\tilde{\boldsymbol{A}}$.

Proof. Take the diagonal matrix

$$\boldsymbol{D}=\operatorname{diag}(d_1,\ d_2,\ \cdots,\ d_n),\ \text{ here, }\ d_1=1,\ d_i=d_{i-1}\sqrt{b_{i-1}/c_{i-1}},\ i=2,\ 3,\ \cdots,\ n,$$

and we have $\boldsymbol{DAD}^{-1}=\tilde{\boldsymbol{A}}$.

Lemma 2[10]. Let $\{d_k\}_{k=0}^{n}$ is the ordered principal minors of the matrix $\boldsymbol{A}$, with $d_0 = 1$, $d_1 = a_1$. Then $d_k=a_kd_{k-1}-b_{k-1}c_{k-1}d_{k-2}$.

Lemma 3. All the characteristic values of the matrix $\boldsymbol{A}$ are positive real numbers if and only if all the ordered principal minors of the matrix $\boldsymbol{A}$ are greater than 0.

Proof. By Lemma 1, that all the characteristic values of $\boldsymbol{A}$ are positive real numbers is equivalent to that the matrix $\tilde{\boldsymbol{A}}$ is a positive definite matrix. It is equivalent to all the ordered principal minors of the matrix $\tilde{\boldsymbol{A}}$ are greater than 0. By Lemma 2, the ordered principal minors of the matrix $\boldsymbol{A}$ and the matrix $\tilde{\boldsymbol{A}}$ are the same.

In the following we will discuss the convergence of the iterative format of the non-uniform B-spline curve. In the discussion, we will write the characteristic values of the $m \times m$ matrix $\boldsymbol{M}$ as $\lambda_i(\boldsymbol{M})$, $i=1, 2, \cdots m$, and the spectral radius of the matrix $\boldsymbol{M}$ as $\rho(\boldsymbol{M})$.

Theorem 1. The iterative format (7) of the non-uniform B-spline curve is convergent.

Proof. Since the subsequence $\{t_i\}_{i=4}^{n+1}$ of (1) is strictly increasing, by Theorem 2 in ref. [11], the matrix $\boldsymbol{B}$ is totally positive, namely, all of its minors are greater than or equal to 0. Specifically, all of its ordered principal minors are greater than or equal to 0. Since the matrices corresponding to the ordered principal minors of the matrix $\boldsymbol{B}$ are all invertible, the ordered principal minors of the matrix $\boldsymbol{B}$ are nothing but greater than 0. By Lemma 3, $\lambda_i(\boldsymbol{B})>0$, $i=1, 2, \cdots, n-2$. Together with $\|\boldsymbol{B}\|_\infty=1$, we have

$$0<\lambda_i(\boldsymbol{B})\leq 1,\ i=1, 2, \cdots, n-2.$$

Hence $0\leq\lambda_i(\boldsymbol{C})=1-\lambda_i(\boldsymbol{B})<1$, $i=1, 2, \cdots, n-2$ for $\boldsymbol{C}=\boldsymbol{I}-\boldsymbol{B}$. It means $\rho(\boldsymbol{C})<1$. Then the iterative format (7) is convergent.

The convergence of the iterative non-uniform B-spline curves $\{\boldsymbol{r}^k(t)|k=1, 2, \cdots\}$ means that all the non-uniform B-spline curves obtained within finite times iteration fit the given point set $\{\boldsymbol{P}_i\}_{i=1}^{n}$, and the fitting precision is finer and finer while increasing the iterative time, then, at last, the limit curve of the iterative non-uniform B-spline curve interpolates the given point set $\{\boldsymbol{P}_i\}_{i=1}^{n}$. But, the iterative non-uniform planar B-spline curve obtained with iterative format (7) does not hold convexity-preserving property, so we must do little modification to the iterative format (7) for getting convexity-preserving iterative non-uniform planar B-spline curve.

1.3 Convexity-preserving iterative non-uniform planar B-spline curve

Convexity-preserving property is a basic requirement to planar fitting (interpolating) curve. By the properties of planar B-spline curve, the convexity of its control polygon ensures the convexity of itself. Suppose the polyline L obtained by orderly connecting the planar point set $\{\boldsymbol{P}_i\}_{i=1}^{n}$ is convex. Since the adjustments to the control points of the iterative non-uniform planar B-spline curve are little at each iteration, commonly, the planar B-spline curves within the initial some iterative times are convexity preserving. But, with the increasing of the iterative times, the obtained planar B-spline curve probably cannot hold convexity-preserving property. We design an algorithm to examine whether the iterative non-uniform planar B-spline curve after each iteration is convexity preserving. When the iterative non-uniform planar B-spline curve $\boldsymbol{r}^k(t)$ is convexity preserving, but the next curve $\boldsymbol{r}^{k+1}(t)$ is not, we can adjust the control points of the curve $\boldsymbol{r}^{k+1}(t)$, so that the convexity of its control polygon is the same as that of the control polygon of $\boldsymbol{r}^k(t)$, and then $\boldsymbol{r}^{k+1}(t)$ is convexity-preserving.

As for the planar polyline $L=\overline{\boldsymbol{P}_1\boldsymbol{P}_2\cdots\boldsymbol{P}_n}$ in fig. 2, the angle from the vector $\overrightarrow{\boldsymbol{P}_i\boldsymbol{P}_{i+1}}$ to $\overrightarrow{\boldsymbol{P}_{i+1}\boldsymbol{P}_{i+2}}$ is called the directional angle from $\overrightarrow{\boldsymbol{P}_i\boldsymbol{P}_{i+1}}$ to $\overrightarrow{\boldsymbol{P}_{i+1}\boldsymbol{P}_{i+2}}$, written as $\alpha_i=(\overline{\boldsymbol{P}_i\boldsymbol{P}_{i+1}},\overline{\boldsymbol{P}_{i+1}\boldsymbol{P}_{i+2}})$, with prescription that the counter-clockwise directional angle is positive and the clockwise directional angle is negative. Similarly, as to planar polyline $L'=\overline{\boldsymbol{Q}_1\boldsymbol{Q}_2\cdots\boldsymbol{Q}_n}$, there are directional angles $\beta_i=(\overline{\boldsymbol{Q}_i\boldsymbol{Q}_{i+1}},\ \overline{\boldsymbol{Q}_{i+1}\boldsymbol{Q}_{i+2}})$, i=1, 2, $\cdots$, n−2. When sgn(α_i) = sgn(β_i), i = 1, 2, $\cdots$, n−2, we say that both the polylines L 'and L hold the same direction. Then, if the polyline L is convex, we can deduce that the polyline L' is also convex. Hence, after obtaining the non-uniform planar B-spline curve $\boldsymbol{r}^{k+1}(t)$ by iterating from $\boldsymbol{r}^k(t)$, we only need to examine whether their corresponding control polygons hold the same direction. If not, we adjust the control points $\{\boldsymbol{P}_i^{k+1}\}_{i=0}^{n+1}$ of the curve $\boldsymbol{r}^{k+1}(t)$ according to the following algorithm so that the control polygon L^{k+1} of the curve $\boldsymbol{r}^{k+1}(t)$ holds the same direction as the control polygon L^k of the curve $\boldsymbol{r}^k(t)$. The iterative non-uniform planar B-spline curve so generated is convexity preserving.

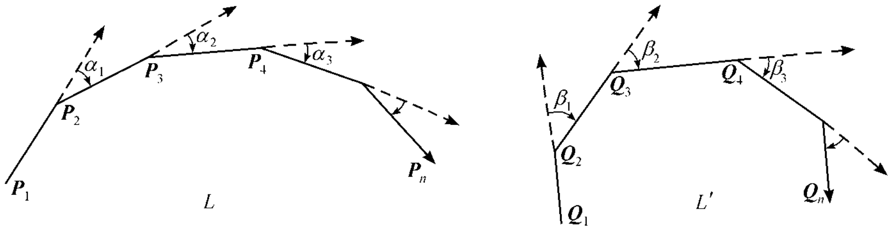

Fig. 2. Polyline L is convexity, and if sgn(α_i)=sgn(β_i), i=1, 2, $\cdots$, n−2, polyline L' is also convexity.

Suppose that the vertices of the polyline $\bar{L}$ are $\{\bar{\boldsymbol{P}}_i\}_{i=0}^{n+1}$, and those of the polyline $\hat{L}$ are $\{\hat{\boldsymbol{P}}_i\}_{i=0}^{n+1}$. The polyline $L'=\dfrac{\bar{L}+\hat{L}}{2}$ means that its vertices are $\left\{\dfrac{\bar{\boldsymbol{P}}_i+\hat{\boldsymbol{P}}_i}{2}\right\}_{i=0}^{n+1}$.

Moreover, suppose $\operatorname{dist}(\hat{L},\bar{L})=\max\limits_{1\le i\le n}\|\bar{\boldsymbol{P}}_i-\hat{\boldsymbol{P}}_i\|$, L^j is the control polygon of the curve $\boldsymbol{r}^j(t)$(j=k, k+1), and D is a prescribed threshold. Thereby, we have the following algorithm.

Algorithm 1. (Adjusting the control points of the curve $\boldsymbol{r}^{k+1}(t)$)

1. //Searching for a piece of polyline within L^k and L^{k+1}, which holds the same di-

```
rection as $L^k$;
    2. $\hat{L}=L^k$;  $\bar{L}=L^{k+1}$;  $L'=\frac{\bar{L}+\hat{L}}{2}$;
    3. While ($L'$ and $L^k$ hold different directions)
    4. $\left\{\bar{L}=L';\ L'=\frac{\bar{L}+\hat{L}}{2};\right\}$
    5. //Searching for a polyline as close as possible to $L^{k+1}$ and with the same direction as $L^k$;
    6. $\hat{L}=L'$;  $\bar{L}=L^{k+1}$;
    7. While (dist$(\hat{L},\ \bar{L})$>$D$)
    8. $\{$ $L'=\frac{\bar{L}+\hat{L}}{2}$;
    9. if (Both $L'$ and $L^k$ hold the same direction)  $\hat{L}=L'$;
    10. else  $\bar{L}=L'$; $\}$
    11. Output  $\hat{L}$;
```

We will give some interpretation for Algorithm 1 as follows. Firstly, and obviously, step 7 in the algorithm implies that the circulation does not end until the distance between the two polylines is sufficiently small (less than D). Secondly, and essentially, the algorithm searches for a suitable value $\varepsilon_k \in (0,\ 1)$, k=0, 1, $\cdots$, so that the control polygon L^{k+1} obtained by orderly connecting the point sequence $\boldsymbol{P}_i^{k+1}=\boldsymbol{P}_i^k+\varepsilon_k\boldsymbol{\Delta}_i^k$, i=1, 2, $\cdots$, n holds the same direction as the control polygon L^k.

After adjusting the control points of the curve $\boldsymbol{r}^{k+1}(t)$ by Algorithm 1, the (k+2)-th adjusting vector of its j-th control point is

$$
\begin{aligned}
&\boldsymbol{\Delta}_j^{k+1}\\
&=\boldsymbol{P}_j-\boldsymbol{r}^{k+1}(t_{j+2})=\boldsymbol{P}_j-\sum_{i=0}^{n+1}(\boldsymbol{P}_i^k+\varepsilon_k\boldsymbol{\Delta}_i^k)B_i^4(t_{j+2})\\
&=\boldsymbol{P}_j-(\boldsymbol{P}_{j-1}^k+\varepsilon_k\boldsymbol{\Delta}_{j-1}^k)B_{j-1}^4(t_{j+2})-(\boldsymbol{P}_j^k+\varepsilon_k\boldsymbol{\Delta}_j^k)B_j^4(t_{j+2})-(\boldsymbol{P}_{j+1}^k+\varepsilon_k\boldsymbol{\Delta}_{j+1}^k)B_{j+1}^4(t_{j+2})\\
&=(\boldsymbol{P}_j-\boldsymbol{r}^k(t_{j+2}))-\varepsilon_k\boldsymbol{\Delta}_{j-1}^kB_{j-1}^4(t_{j+2})-\varepsilon_k\boldsymbol{\Delta}_j^kB_j^4(t_{j+2})-\varepsilon_k\boldsymbol{\Delta}_{j+1}^kB_{j+1}^4(t_{j+2})
\end{aligned}
$$

$$= -\varepsilon_k B_{j-1}^4(t_{j+2})\boldsymbol{\Delta}_{j-1}^k + (1-\varepsilon_k B_j^4(t_{j+2}))\boldsymbol{\Delta}_j^k - \varepsilon_k B_{j+1}^4(t_{j+2})\boldsymbol{\Delta}_{j+1}^k,$$

$$\varepsilon_k \in (0,1),\ \ k=0,\ 1,\ \cdots;j=2,\ 3,\ \cdots n-1.$$

Similar to (7), we have the convexity-preserving iterative format of the adjusting vectors of the control point set in matrix

$$[\boldsymbol{\Delta}_2^{k+1}, \boldsymbol{\Delta}_3^{k+1}, \cdots, \boldsymbol{\Delta}_{n-1}^{k+1}]^{\mathrm{T}} = \boldsymbol{C}_k[\boldsymbol{\Delta}_2^k, \boldsymbol{\Delta}_3^k, \cdots, \boldsymbol{\Delta}_{n-1}^k]^{\mathrm{T}},\ \ \boldsymbol{C}_k = \boldsymbol{I} - \varepsilon_k \boldsymbol{B},\ 0 < \varepsilon_k < 1;\ k = 0,1,\cdots; \tag{11}$$

here, $\boldsymbol{I}$ is $n-2$ rank identity matrix, and $\boldsymbol{B}$ is the same as (8).

In the following, we will show that the convexity-preserving iterative format (11) is convergent.

Theorem 2. The convexity-preserving iterative format (11) of the non-uniform planar B-spline curve is convergent.

Proof. As the proof of Theorem 1, we have $0<\lambda_i(\boldsymbol{B})\leq 1$, $i=1, 2, \cdots, n-2$. Suppose there is $n-2$ rank invertible matrix $\boldsymbol{X}$, so that

$$\boldsymbol{X}^{-1}\boldsymbol{BX} = \mathrm{diag}(\lambda_1(\boldsymbol{B}), \lambda_2(\boldsymbol{B}), \cdots, \lambda_{n-2}(\boldsymbol{B}));$$

then

$$\boldsymbol{X}^{-1}\boldsymbol{C}_k\boldsymbol{X} = \mathrm{diag}(1-\varepsilon_k\lambda_1(\boldsymbol{B}), 1-\varepsilon_k\lambda_2(\boldsymbol{B}), \cdots, 1-\varepsilon_k\lambda_{n-2}(\boldsymbol{B})).$$

Together with $0<\varepsilon_k<1$, we have $0<1-\varepsilon_k\lambda_i(\boldsymbol{B})<1$, $k=0, 1, \cdots$; $i=1, 2, \cdots, n-2$. Then

$$\prod_{k=0}^{\infty}\boldsymbol{C}_k = \boldsymbol{X}\mathrm{diag}\left(\prod_{k=0}^{\infty}(1-\varepsilon_k\lambda_1(\boldsymbol{B})),\ \prod_{k=0}^{\infty}(1-\varepsilon_k\lambda_2(\boldsymbol{B})),\cdots,\prod_{k=0}^{\infty}(1-\varepsilon_k\lambda_n(\boldsymbol{B}))\right)\boldsymbol{X}^{-1} = 0.$$

It means the convergence of the convexity-preserving iterative format (11).

2 Iterative non-uniform B-spline surface

2.1 Deducing of the iterative format

Given an ordered point set $\{\boldsymbol{P}_{ij}\}_{i=1}^{m}{}_{j=1}^{n} \in \mathbb{R}^3$, we can fit it with iterative non-uniform B-spline surface. In the following iterative procedure, suppose that the non-uniform B-spline bases $B_i^4(u)$ and $B_i^4(v)$ are defined on the knot vectors $\{u_i\}_{i=0}^{m+5}$ and $\{v_j\}_{j=0}^{n+5}$, here

$$\begin{cases} u_i = u_{i-1} + \dfrac{1}{n}\displaystyle\sum_{j=1}^{n} \| \boldsymbol{P}_{i-2,j} - \boldsymbol{P}_{i-3,j} \|,\ \ i = 4,\ 5,\ \cdots, m+2\ ; \\ u_0 = u_1 = u_2 = u_3 = 0,\ \ u_{m+2} = u_{m+3} = u_{m+4} = u_{m+5}; \end{cases} \tag{12}$$

$$\begin{cases} v_j = v_{j-1} + \dfrac{1}{m}\displaystyle\sum_{i=1}^{m} \| \boldsymbol{P}_{i,j-2} - \boldsymbol{P}_{i,j-3} \|, \quad j = 4, 5, \cdots, n+2; \\ v_0 = v_1 = v_2 = v_3 = 0, v_{n+2} = v_{n+3} = v_{n+4} = v_{n+5}. \end{cases} \tag{13}$$

At the beginning of the iteration, let

$$\begin{cases} \{\boldsymbol{P}_{ij}^0 = \boldsymbol{P}_{ij}\}_{i=1}^{m}{}_{j=1}^{n}, \ \{\boldsymbol{P}_{0j}^0 = \boldsymbol{P}_{1j}^0, \ \boldsymbol{P}_{m+1,j}^0 = \boldsymbol{P}_{m,j}^0\}_{j=1}^{n}, \ \{\boldsymbol{P}_{i0}^0 = \boldsymbol{P}_{i1}^0, \ \boldsymbol{P}_{i,n+1}^0 = \boldsymbol{P}_{i,n}^0\}_{i=1}^{m}; \\ \boldsymbol{P}_{00}^0 = \boldsymbol{P}_{11}^0, \ \boldsymbol{P}_{0,n+1}^0 = \boldsymbol{P}_{1,n}^0, \ \boldsymbol{P}_{m+1,n+1}^0 = \boldsymbol{P}_{m,n}^0, \ \boldsymbol{P}_{m+1,0}^0 = \boldsymbol{P}_{m,1}^0. \end{cases} \tag{14}$$

First, taking $\{\boldsymbol{P}_{ij}^0\}_{i=0}^{m+1}{}_{j=0}^{n+1}$ as the control point set, we obtain a piece of bi-cubic non-uniform B-spline surface

$$\boldsymbol{r}^0(u,v) = \sum_{i=0}^{m+1}\sum_{j=0}^{n+1} \boldsymbol{P}_{ij}^0 B_i^4(u) B_j^4(v), \qquad (u,v) \in [u_3, u_{m+2}] \times [v_3, v_{n+2}]. \tag{15}$$

Further, let

$$\begin{cases} \boldsymbol{\varDelta}_{ij}^0 = \boldsymbol{P}_{ij} - \boldsymbol{r}^0(u_{i+2}, v_{j+2}), \ \ i = 1,2,\cdots,m, j = 1,2,\cdots,n; \\ \{\boldsymbol{\varDelta}_{h,0}^0 = \boldsymbol{\varDelta}_{h,1}^0, \ \ \boldsymbol{\varDelta}_{h,n+1}^0 = \boldsymbol{\varDelta}_{h,n}^0\}_{h=1}^{m}, \{\boldsymbol{\varDelta}_{0,l}^0 = \boldsymbol{\varDelta}_{1,l}^0, \ \ \boldsymbol{\varDelta}_{m+1,l}^0 = \boldsymbol{\varDelta}_{m,l}^0\}_{l=1}^{n}; \\ \boldsymbol{\varDelta}_{00}^0 = \boldsymbol{\varDelta}_{m+1,0}^0 = \boldsymbol{\varDelta}_{0,n+1}^0 = \boldsymbol{\varDelta}_{m+1,n+1}^0 = \mathbf{0}; \end{cases}$$

and let

$$\{\boldsymbol{P}_{ij}^1 = \boldsymbol{P}_{ij}^0 + \boldsymbol{\varDelta}_{ij}^0\}_{i=0}^{m+1}{}_{j=0}^{n+1};$$

then, with $\{\boldsymbol{P}_{ij}^1\}_{i=0}^{m+1}{}_{j=0}^{n+1}$ as the control point set, after the first iteration we obtain the bi-cubic non-uniform B-spline surface

$$\boldsymbol{r}^1(u,v) = \sum_{i=0}^{m+1}\sum_{j=0}^{n+1} \boldsymbol{P}_{ij}^1 B_i^4(u) B_j^4(v), \qquad (u,v) \in [u_3, u_{m+2}] \times [v_3, v_{n+2}].$$

Similarly, if we have gotten the non-uniform B-spline surface $\boldsymbol{r}^k(u, v)$ after the k-th iteration, let

$$\begin{cases} \boldsymbol{\varDelta}_{ij}^k = \boldsymbol{P}_{ij} - \boldsymbol{r}^k(u_{i+2}, v_{j+2}) \ \ i = 1,2,\cdots,m, j = 1,2,\cdots,n; \\ \{\boldsymbol{\varDelta}_{h,0}^k = \boldsymbol{\varDelta}_{h,1}^k, \ \ \boldsymbol{\varDelta}_{h,n+1}^k = \boldsymbol{\varDelta}_{h,n}^k\}_{h=1}^{m}, \{\boldsymbol{\varDelta}_{0,l}^k = \boldsymbol{\varDelta}_{1,l}^k, \ \ \boldsymbol{\varDelta}_{m+1,l}^k = \boldsymbol{\varDelta}_{m,l}^k\}_{l=1}^{n}; \\ \boldsymbol{\varDelta}_{00}^k = \boldsymbol{\varDelta}_{m+1,0}^k = \boldsymbol{\varDelta}_{0,n+1}^k = \boldsymbol{\varDelta}_{m+1,n+1}^k = \mathbf{0}; \end{cases} \tag{16}$$

and let

$$\{\boldsymbol{P}_{ij}^{k+1} = \boldsymbol{P}_{ij}^k + \boldsymbol{\varDelta}_{ij}^k\}_{i=0}^{m+1}{}_{j=0}^{n+1}; \tag{17}$$

then, we can get

$$\boldsymbol{r}^{k+1}(u,v)=\sum_{i=0}^{m+1}\sum_{j=0}^{n+1}\boldsymbol{P}_{ij}^{k+1}B_i^4(u)\,B_j^4(v),\qquad (u,v)\in[u_3,u_{m+2}]\times[v_3,v_{n+2}], \tag{18}$$

the non-uniform B-spline surface after the (k+1)-th iteration. The sequence of the non-uniform B-spline surfaces so generated, $\{\boldsymbol{r}^{\,k}(u, v)|k=0, 1, \cdots\}$, is called the iterative non-uniform B-spline surface.

Since

$$\begin{aligned}\boldsymbol{\varDelta}_{hl}^{k+1}&=\boldsymbol{P}_{hl}-\sum_{i=0}^{m+1}\sum_{j=0}^{n+1}(\boldsymbol{P}_{ij}^{k}+\boldsymbol{\varDelta}_{ij}^{k})B_i^4(u_{h+2})B_j^4(v_{l+2})\\&=\boldsymbol{P}_{hl}-\sum_{i=0}^{m+1}\sum_{j=0}^{n+1}\boldsymbol{P}_{ij}^{k}B_i^4(u_{h+2})B_j^4(v_{l+2})-\sum_{i=0}^{m+1}\sum_{j=0}^{n+1}\boldsymbol{\varDelta}_{ij}^{k}B_i^4(u_{h+2})B_j^4(v_{l+2})\\&=\boldsymbol{\varDelta}_{hl}^{k}-\sum_{i=h-1}^{h+1}\sum_{j=l-1}^{l+1}\boldsymbol{\varDelta}_{ij}^{k}B_i^4(u_{h+2})B_j^4(v_{l+2})\,;\ h=1, 2, \cdots, m,\ l=1, 2, \cdots, n,\ k=0, 1, \cdots;\end{aligned}$$

and noting that

$$\{\boldsymbol{\varDelta}_{h,0}^{k+1}=\boldsymbol{\varDelta}_{h,1}^{k+1},\ \boldsymbol{\varDelta}_{h,n+1}^{k+1}=\boldsymbol{\varDelta}_{h,n}^{k+1}\}_{h=1}^{m},\quad \{\boldsymbol{\varDelta}_{0,l}^{k+1}=\boldsymbol{\varDelta}_{1,l}^{k+1},\ \boldsymbol{\varDelta}_{m+1,l}^{k+1}=\boldsymbol{\varDelta}_{m,l}^{k+1}\}_{l=1}^{n},$$

$$\boldsymbol{\varDelta}_{00}^{k+1}=\boldsymbol{\varDelta}_{m+1,0}^{k+1}=\boldsymbol{\varDelta}_{0,n+1}^{k+1}=\boldsymbol{\varDelta}_{m+1,n+1}^{k+1}=\mathbf{0};\ \ k=0, 1, \cdots;$$

together with

$$\boldsymbol{\varDelta}_{00}^{k}=\boldsymbol{\varDelta}_{01}^{k}=\boldsymbol{\varDelta}_{10}^{k}=\boldsymbol{\varDelta}_{11}^{k}=\mathbf{0}\,,\ \ \boldsymbol{\varDelta}_{0,n+1}^{k}=\boldsymbol{\varDelta}_{0,n}^{k}=\boldsymbol{\varDelta}_{1,n+1}^{k}=\boldsymbol{\varDelta}_{1,n}^{k}=\mathbf{0},$$

$$\boldsymbol{\varDelta}_{m+1,n+1}^{k}=\boldsymbol{\varDelta}_{m,n+1}^{k}=\boldsymbol{\varDelta}_{m+1,n}^{k}=\boldsymbol{\varDelta}_{m,n}^{k}=\mathbf{0},\ \ \boldsymbol{\varDelta}_{m+1,0}^{k}=\boldsymbol{\varDelta}_{m,0}^{k}=\boldsymbol{\varDelta}_{m+1,1}^{k}=\boldsymbol{\varDelta}_{m,1}^{k}=\mathbf{0};$$

we get the iterative format of the adjusting vectors of the control point set in matrix

$$\boldsymbol{\varDelta}^{k+1}=\boldsymbol{C\varDelta}^{k},\ \ \boldsymbol{C}=\boldsymbol{I}-\boldsymbol{B},\ \ k=0,\ 1,\ \cdots; \tag{19}$$

here

$$\boldsymbol{\varDelta}^{j}=[\boldsymbol{\varDelta}_{11}^{j},\boldsymbol{\varDelta}_{12}^{j},\cdots,\boldsymbol{\varDelta}_{1n}^{j},\boldsymbol{\varDelta}_{21}^{j},\cdots,\boldsymbol{\varDelta}_{2n}^{j},\cdots,\boldsymbol{\varDelta}_{m1}^{j},\cdots,\boldsymbol{\varDelta}_{mn}^{j}]^{\mathrm{T}},\ \ j=k,k+1; \tag{20}$$

$\boldsymbol{I}$ is an $mn\times mn$ identity matrix, and matrix $\boldsymbol{B}$ is the Kronecker product of the matrices $\boldsymbol{B}_1$ and $\boldsymbol{B}_2$, namely

$$\boldsymbol{B}=\boldsymbol{B}_1\otimes\boldsymbol{B}_2, \tag{21}$$

$$\boldsymbol{B}_1=\begin{bmatrix} B_0^4(u_3) & 0 & & & & & & \\ B_1^4(u_4) & B_2^4(u_4) & B_3^4(u_4) & & & & & \\ & B_2^4(u_5) & B_3^4(u_5) & B_4^4(u_5) & & & & \\ & \ddots & \ddots & \ddots & & & & \\ & & & B_{m-3}^4(u_m) & B_{m-2}^4(u_m) & B_{m-1}^4(u_m) & \\ & & & & B_{m-2}^4(u_{m+1}) & B_{m-1}^4(u_{m+1}) & B_m^4(u_{m+1}) \\ & & & & & 0 & B_{m+1}^4(u_{m+2}) \end{bmatrix}, \tag{22}$$

$$\boldsymbol{B}_2=\begin{bmatrix} B_0^4(v_3) & 0 & & & & & & \\ B_1^4(v_4) & B_2^4(v_4) & B_3^4(v_4) & & & & & \\ & B_2^4(v_5) & B_3^4(v_5) & B_4^4(v_5) & & & & \\ & \ddots & \ddots & \ddots & & & & \\ & & & B_{n-3}^4(v_n) & B_{n-2}^4(v_n) & B_{n-1}^4(v_n) & \\ & & & & B_{n-2}^4(v_{n+1}) & B_{n-1}^4(v_{n+1}) & B_n^4(v_{n+1}) \\ & & & & & 0 & B_{n+1}^4(v_{n+2}) \end{bmatrix}. \tag{23}$$

2.2 Convergence of the iterative non-uniform B-spline surface

Lemma 4[10]**.** Let the matrix $\boldsymbol{A}\in\mathbb{R}_{m\times m}$, and the matrix $\boldsymbol{B}\in\mathbb{R}_{m\times m}$, each characteristic value of the matrix $\boldsymbol{A}\otimes\boldsymbol{B}$ can be expressed as the product between the characteristic value of $\boldsymbol{A}$ and that of $\boldsymbol{B}$. That is, if $\lambda(\boldsymbol{A})=\{\lambda_1, \cdots, \lambda_m\}$ and $\lambda(\boldsymbol{B})=\{\mu_1, \cdots, \mu_n\}$, we have

$$\lambda(\boldsymbol{A}\otimes\boldsymbol{B})=\{\lambda_i\mu_j: i=1, \cdots, m; j=1, \cdots, n\}.$$

Here, the characteristic values are counted with their algebraic multiplicity.

In order to analyze the convergence of the iterative non-uniform B-spline surface by Lemma 4, we first write the matrices (22) and (23) as the matrices divided into blocks, namely

$$\boldsymbol{B}_1=\begin{bmatrix} B_0^4(u_3) & \boldsymbol{0} & 0 \\ \boldsymbol{F}_1 & \boldsymbol{D}_1 & \boldsymbol{G}_1 \\ 0 & \boldsymbol{0} & B_{m+1}^4(u_{m+2}) \end{bmatrix}, \boldsymbol{B}_2=\begin{bmatrix} B_0^4(v_3) & \boldsymbol{0} & 0 \\ \boldsymbol{F}_2 & \boldsymbol{D}_2 & \boldsymbol{G}_2 \\ 0 & \boldsymbol{0} & B_{n+1}^4(v_{n+2}) \end{bmatrix}, \tag{24}$$

where matrices $\boldsymbol{D}_1$ and $\boldsymbol{D}_2$ are defined similar to (8), only substituting the variable u or v for the variable t, and $m-2$ or $n-2$ rank for the rank of the matrix in (8), additionally, we have

$$\begin{cases} \boldsymbol{F}_1 = \left[B_1^4(u_4),0,0,\cdots,0\right]_{(m-2)\times 1}^{\mathrm{T}}, \ \boldsymbol{G}_1 = \left[0,0,\cdots,0,B_m^4(u_{m+1})\right]_{(m-2)\times 1}^{\mathrm{T}} \\ \boldsymbol{F}_2 = \left[B_1^4(v_4),0,0,\cdots,0\right]_{(n-2)\times 1}^{\mathrm{T}}, \ \boldsymbol{G}_2 = \left[0,0,\cdots,0,B_n^4(v_{n+1})\right]_{(n-2)\times 1}^{\mathrm{T}} \end{cases}. \tag{25}$$

Theorem 3. The iterative format (19) of the non-uniform B-spline surface is convergent.

Proof. Similar to the proof of Theorem 1, we have

$$0<\lambda_i(\boldsymbol{D}_1)\leq 1,\ i=1,2,\ \cdots,m-2;\ 0<\lambda_i(\boldsymbol{D}_2)\leq 1,\ i=1,2,\ \cdots,n-2.$$

Moreover, since the characteristic polynomials of the matrices $\boldsymbol{B}_1$ and $\boldsymbol{B}_2$ are

$$\begin{cases} (\lambda - B_0^4(u_3))(\lambda - B_{m+1}^4(u_{m+2}))\times \det(\lambda\boldsymbol{I}-\boldsymbol{D}_1) = (\lambda-1)^2\det(\lambda\boldsymbol{I}-\boldsymbol{D}_1) \\ (\lambda - B_0^4(v_3))(\lambda - B_{n+1}^4(v_{n+2}))\times \det(\lambda\boldsymbol{I}-\boldsymbol{D}_2) = (\lambda-1)^2\det(\lambda\boldsymbol{I}-\boldsymbol{D}_2) \end{cases}, \tag{26}$$

we have

$$0<\lambda_i(\boldsymbol{B}_1)\leq 1,\ i=1,2,\ \cdots,m;\ 0<\lambda_i(\boldsymbol{B}_2)\leq 1,\ i=1,2,\ \cdots,n.$$

Together with Lemma 4,

$$0<\lambda_i(\boldsymbol{B})=\lambda_i(\boldsymbol{B}_1\otimes\boldsymbol{B}_2)\leq 1,\ i=1,2,\ \cdots,mn.$$

Since $\boldsymbol{C}=\boldsymbol{I}-\boldsymbol{B}$, we have $0\leq\lambda_i(\boldsymbol{C})=1-\lambda_i(\boldsymbol{B})<1$, $i=1,2,\ \cdots,mn$. It means $\rho(\boldsymbol{C})<1$. Hence the iterative format (19) is convergent.

3 Analysis of the convergence rate and the computational complexity

Definition 1[10]**.** Let $\boldsymbol{G}$ be the iterative matrix of the iterative non-uniform B-spline curve (surface), and let $\rho(\boldsymbol{G})$ be its spectral radius; then, $R(\boldsymbol{G})=-\ln\rho(\boldsymbol{G})$ is called the asymptotic convergence rate of the iteration.

First we analyze the convergence rate of the iteration. Since the iterative matrix of the iterative format (7) is the matrix $\boldsymbol{C}$, and

$$\rho(C)\leq\|C\|_\infty=\max_{2\leq i\leq n-1}\{1-B_i^4(t_{i+2})+B_{i-1}^4(t_{i+2})+B_{i+1}^4(t_{i+2})\}=\max_{2\leq i\leq n-1}\{2-2B_i^4(t_{i+2})\},$$

the smaller $\max\limits_{2\leq i\leq n-1}\{2-2B_i^4(t_{i+2})\}$, the faster the convergence rate of the iterative format (7). Similarly, for the iterative format (11), the smaller $\max\limits_{2\leq i\leq n-1}\{(1+\varepsilon_k)-2\varepsilon_k B_i^4(t_{i+2})\}$, $k=0,1,\ \cdots$, the faster the convergence rate of (11); and for (19), the smaller $\max\limits_{\substack{1\leq i\leq m\\1\leq j\leq n}}\{2-2B_i^4(u_{i+2})B_j^4(v_{j+2})\}$, the faster convergence rate of (19).

Second, we analyze the computational complexity of the iteration. As to the iterative non-uniform B-spline curve (surface), suppose the number of the fitted points is *N*. Obviously, the computational complexity for each iteration is *O*(*N*). For each iteration, it is required to calculate the *N* points on the B-spline curve (surface) at the knots with *N* times de Boor algorithm and to calculate the distances between these points to their corresponding fitted points with *N* times square operation.

4 Numerical examples

In the following examples, the circle points denote the fitted points, the polyline (mesh) denotes the control polygon (control mesh) obtained by orderly connecting the fitted points, the curve (surface) denotes the iterative curve (surface), and the points marked '+' denote the points that are at the parameter knots. The fitting error of the iterative non-uniform B-spline curve is taken as error= $\max\limits_{2\le i\le n-2} \| \boldsymbol{r}^k(t_{i+2}) - \boldsymbol{P}_i \|$, and the error of the iterative non-uniform B-spline surface is taken as error= $\max\limits_{\substack{1\le i\le m\\1\le j\le n}} \| \boldsymbol{r}^k(u_{i+2}, v_{j+2}) - \boldsymbol{P}_{ij} \|$. The readers can refer to sections 2 and 3 for the symbols in the above two formulae. In fig. 3, a space helix is fitted by the iterative non-uniform B-spline curve. In fig. 4, a planar free curve is fitted by the iterative non-uniform B-spline curve. fig. 5 illustrates the convexity-preserving property of the planar iterative

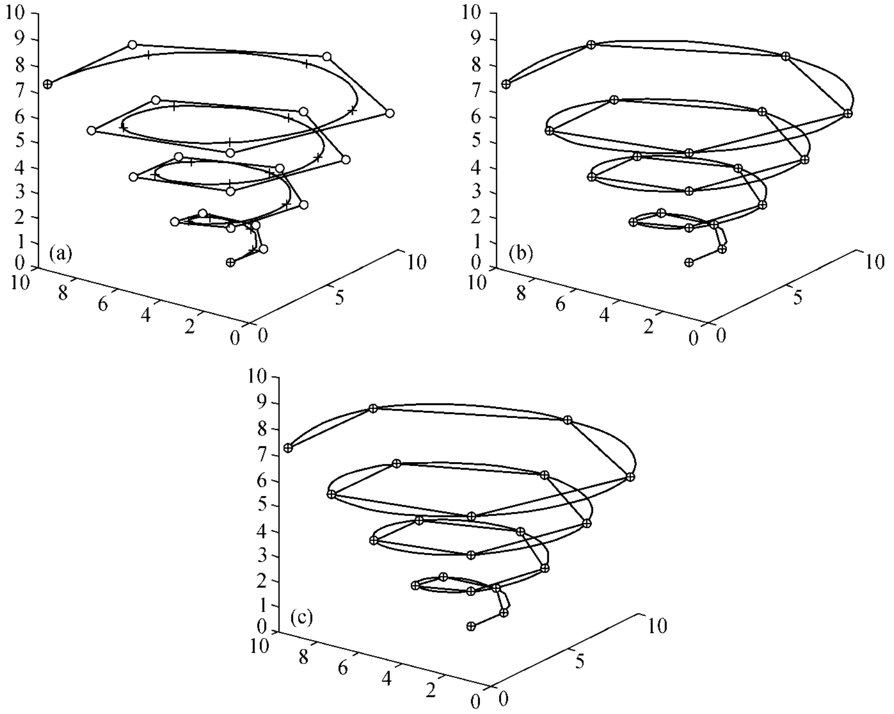

Fig. 3. Space iterative non-uniform B-spline curve fitting helix. Top left: The iterative time is 0, the error is 1.960338e000; top right: the iterative time is 4, the error is 6.626295e-003; bottom: the iterative time is 15, the error is 3.765069e-005.

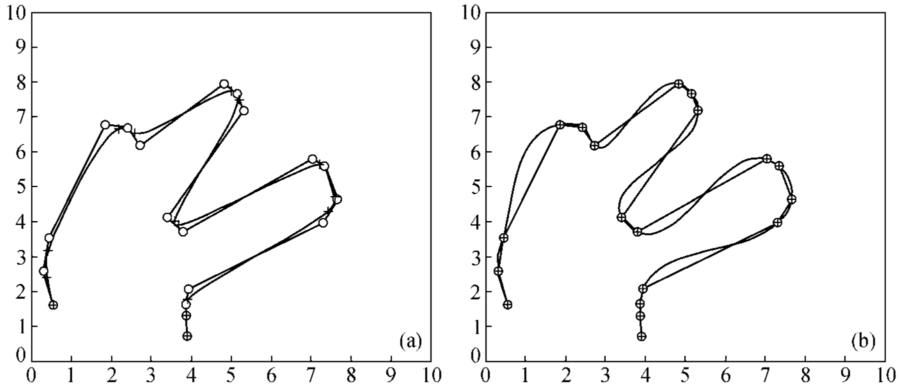

Fig. 4. Planar iterative non-uniform B-spline curve fitting a free curve. Left: The iterative time is 0, the error is 4.222137e-001; Right: the iterative time is 30, the error is 7.073353e-005.

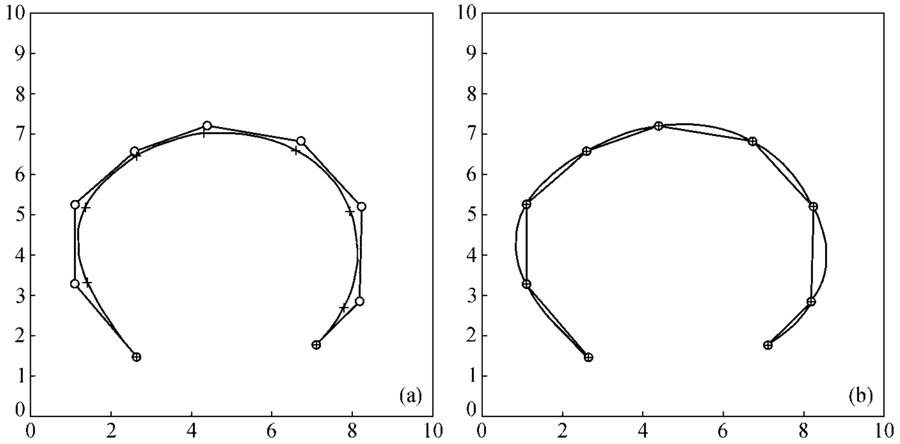

Fig. 5. The convexity-preserving property of the planar iterative non-uniform B-spline curve. Left: The iterative time is 0, the error is 3.700325e–001; Right: the iterative curve is convexity-preserving after 30 times iteration, the error is 8.389196e-008.

non-uniform B-spline curve. In fig. 6, the function peaks in Matlab is fitted by the iterative non-uniform B-spline surface. We list the fitting precisions for iterative non-uniform B-spline curves (surfaces) above in table 1.

Table 1 The fitting precisions of the iterative non-uniform B-spline curve (surface) in fig. 3—fig. 6

	0-th time	2-th time	4-th time	15-th time	30-th time
Fig. 3	1.9603e000	2.8645e–001	6.6263e–003	3.7651e–005	2.1624e–007
Fig. 4	4.2221e–001	1.9064e–001	1.1371e–001	7.7307e–003	7.0734e–005
Fig. 5	3.7003e–001	6.7963e–002	4.9086e–002	2.0160e–004	8.3892e–008
Fig. 6	3.3628e+001	3.5526e–001	1.7233e–001	1.5282e–002	1.2695e–003

5 Conclusions

In this paper, based on the idea of profit and loss modification, we construct an iterative format to generate the sequence of non-uniform B-spline curves and surfaces for

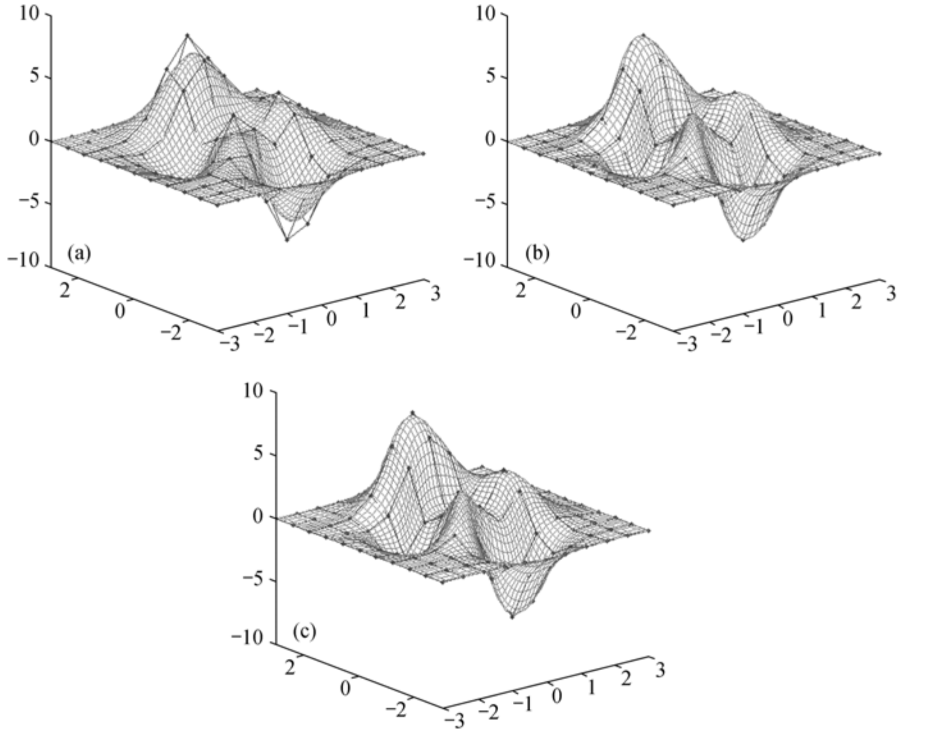

Fig. 6. Iterative non-uniform B-spline surface fitting the function peaks in Matlab. Top left: The iterative time is 0, the error is 3.362778e+001; Top right: The iterative time is 3, the error is 2.422128e−001; Bottom: The iterative time is 15, the error is1.528187e−002.

fitting given ordered point set, and show its convergence. Moreover, we present an improved iterative format to generate the convexity-preserving iterative non-uniform B-spline curve. We also analyze their convergence rate and computational complexity, and give some numerical examples. The theoretic deducing and illustrations show that the iterative non-uniform B-spline curve (surface) avoids the disadvantages of the conventional B-spline interpolating methods, such as solving a global linear system for the control points, etc. By the method of iterative non-uniform B-spline curve (surface), for a given ordered point set, we can get a group of fitting curves (surfaces) with different precisions and holding the advantages such as locality, explicit expression, and satisfying NURBS standard, etc. In addition, their limit curve (surface) interpolates the given point set. As the future work, the research for corresponding problems of the iterative NURBS curve (surface) is required.

Acknowledgements This work is supported jointly by the National Natural Science Foundation of China (Grant Nos. 60373033, 60333010), National Natural Science Foundation for Innovative Research Groups (No. 60021201) and the Foundation of State Key Basic Research 973 Item (Grant No. 2002CB312101).

References

1. Barsky, B. A., Greenberg, D. P., Determining a set of B-spline control vertices to generate an interpolating surface, Comput. Graph and Image Process, 1980, 14(3): 203—226.

2. Ma, W., Kruth, J.-P., NURBS curve and surface fitting and interpolation, in: Dæhlen, M., Lyche, T., Schumaker, L. eds., Mathematical Methods for Curves and Surfaces, Nashville & London: Vanderbilt University Press, 1995.
3. Loe, K. F., α-B-spline: a linear singular blending B-spline, The Visual Computer, 1996, 12: 18—25.[DOI]
4. de Boor, C., A Practical Guide to Splines, New York: Springer-Verlag, 1978.
5. Kvasov, B. I., GB-splines and their properties, Ann. Numer. Math., 1996, 3: 139—149.
6. Gregory, J. A., Shape preserving spline interpolation, Computer-Aided Design, 1986, 18: 53—57.
7. Qi, D. X., Tian, Z. X., Zhang, Y. X. et al., The method of numeric polish in curve fitting, Acta Mathematica Sinica (in Chinese), 1975, 18(3): 173—184.
8. de Boor, C., How does Agee's smoothing method work? in Proceedings of the 1979 Army Numerical Analysis and Computers Conference, ARO Report 79-3, Army Research Office, 1979, 299—302.
9. Qi, D. X., Notes on the mathematic method of computer aided geometric modeling, Journal North China University of Technology (in Chinese), 1991, 3(1): 1—8.
10. Chen, J. L., Chen, X. H., Special Matrices (in Chinese), Beijing: Tsinghua University Press, 2001, 370—371, 429—430.
11. de Boor C., Total positivity of the spline collocation matrix, Indiana University Mathematics Journal, 1976, 25(6): 541—551.