Computer Aided Geometric Design 27 (2010) 322–339

Contents lists available at ScienceDirect

Computer Aided Geometric Design

www.elsevier.com/locate/cagd

Local progressive-iterative approximation format for blending curves and patches

Hongwei Lin

State Key Lab. of CAD&CG, Zhejiang University, Hangzhou 310027, China

ARTICLE INFO

Article history:
Received 13 July 2009
Received in revised form 12 November 2009
Accepted 17 January 2010
Available online 20 January 2010

Keywords:
Progressive-iterative approximation
Local format
Data fitting
Geometric design

ABSTRACT

Just by adjusting the control points iteratively, progressive-iterative approximation presents an intuitive and straightforward way to fit data points. It generates a curve or patch sequence with finer and finer precision, and the limit of the sequence interpolates the data points. The progressive-iterative approximation brings more flexibility for shape controlling in data fitting. In this paper, we design a local progressive-iterative approximation format, and show that the local format is convergent for the blending curve with normalized totally positive basis, and the bi-cubic B-spline patch, which is the most commonly used patch in geometric design. Moreover, a special adjustment manner is designed to make the local progressive-iterative approximation format is convergent for a generic blending patch with normalized totally positive basis. The local progressive-iterative approximation format adjusts only a part of the control points of a blending curve or patch, and the limit curve or patch interpolates the corresponding data points. Based on the local format, data points can be fit adaptively.

1. Introduction

By adjusting the control points of a blending curve or patch iteratively, a sequence of curves or patches can be generated. If the limit of the sequence interpolates the initial control points, we say that the blending curve or patch has the *progressive-iterative approximation* (abbr. *PIA*) property. The PIA property presents an intuitive and straightforward way to generate a sequence of curves or patches with finer and finer precision for data point fitting.

The progressive-iterative approximation (PIA) property of the uniform cubic B-spline curve is discovered by Qi et al. (1975) and de Boor (1979), respectively. In Lin et al. (2004), the authors show that the non-uniform cubic B-spline curve and patch also hold the property. Furthermore, the result is extended to the blending curve and patch with normalized totally positive basis (Lin et al., 2005). That is, any blending curve or patch with normalized totally positive basis has the progressive-iterative approximation (PIA) property. In Delgado and Peña (2007), the convergence rates of different bases are compared, and the basis with the fastest convergence rate is found. Moreover, it is proved that the rational B-spline curve and surface (NURBS) have the property, too (Shi and Wang, 2006). Recently, Martin et al. (2009) devise an iterative format for fitting, which is actually the progressive-iterative approximation (PIA) format for the uniform periodic cubic B-spline.

Furthermore, the PIA format has been extended to subdivision surface fitting. Cheng et al. design the PIA format of subdivision fitting for loop subdivision surface (Cheng et al., 2008, 2009), and prove its convergence. Fan et al. develop the PIA format of Doo–Sabin subdivision surface fitting (Fan et al., 2008). The PIA format for Catmull–Clark subdivision surface fitting is proposed in Chen et al. (2008).

E-mail address: hwlin@zjucadcg.cn.

0167-8396/$ – see front matter
doi:10.1016/j.cagd.2010.01.003

Recently, Maekawa et al. invent an iterative fitting format, called *interpolation by geometric algorithm* (Maekawa et al., 2007; Gofuku et al., 2009), which is similar to PIA format. The main difference between PIA and *interpolation by geometric algorithm* is that the former depends on parametric (or algebraic) distance, while the latter relies on geometric distance. The convergence of the interpolation by geometric algorithm is validated by experiments, without theoretical assurance.

Noticeably, the PIA format mentioned above is *global*, which needs to adjust all of the control points of a curve or patch. In this paper, we first design a *local* progressive-iterative approximation (PIA) format for a blending curve with normalized totally positive basis, and prove its convergence. By the local format, we can adjust only a subset of the control points progressively, and the corresponding points on the limit curve still interpolate the corresponding subset of the initial data points. The local progressive-iterative approximation format brings more flexibility to data fitting: (1) A data point sequence can be fit by one by one point adjustment. (2) The fitting precision for each point can be controlled separately. In other words, some points can be fit with high precision, while other points are fit with low precision. Moreover, though it is difficult to show the convergence of the local PIA format for a generic blending patch, we show that, for a most commonly used case, that is, bi-cubic B-spline patch, the local PIA format is convergent. In addition, a special adjusting manner is developed, which ensures the local PIA format for a generic blending patch with normalized totally positive basis is convergent. However, in the case of fitting a bi-cubic B-spline patch to regular gridded data, the usual method to directly compute the interpolating patch is sometimes faster than the PIA method, as the usual method only requires to solve a tri-diagonal system.

This paper is organized as follows. In Section 2, we develop the local progressive-iterative approximation (PIA) format for a blending curve with normalized totally positive basis, and show its convergence. In Section 3, we design the local PIA format for a blending patch, and present some results on its convergence. Section 4 shows the convergence of the local PIA for a bi-cubic B-spline patch. Moreover, a special adjustment manner is developed in Section 5, which ensures the convergence of the local PIA for a generic blending patch with normalized totally positive basis. Some experimental results are illustrated in Section 6. Finally, the last section concludes the paper.

2. Local PIA format for a blending curve

Given an ordered point sequence $\{\boldsymbol{P}_i \in \mathbb{R}^3 \mid i = 0, 1, \ldots, n\}$, each point $\boldsymbol{P}_i$ is assigned a parameter value t_i, $i = 0, 1, \ldots, n$, satisfying

$$t_0 < t_1 < \cdots < t_n. \tag{1}$$

With the initial control points,

$$\left\{\boldsymbol{P}_i^0 = \boldsymbol{P}_i \;\middle|\; i = 0, 1, \ldots, n\right\}, \tag{2}$$

and a normalized totally positive basis,

$$\left\{B_i(t) \;\middle|\; i = 0, 1, \ldots, n\right\}, \quad \text{with } B_i(t) \geqslant 0, \text{ and } \sum_{i=0}^{n} B_i(t) = 1, \tag{3}$$

an initial blending curve can be constructed, that is,

$$\boldsymbol{P}^0(t) = \sum_{i=0}^{n} \boldsymbol{P}_i^0 B_i(t). \tag{4}$$

By adjusting the control points $\boldsymbol{P}_i^k$, $i = 0, 1, \ldots, n$, $k = 1, 2, \ldots$, along the vectors

$$\boldsymbol{\Delta}_i^k = \boldsymbol{P}_i - \boldsymbol{P}^k(t_i), \tag{5}$$

namely,

$$\boldsymbol{P}_i^{k+1} = \boldsymbol{P}_i^k + \boldsymbol{\Delta}_i^k, \tag{6}$$

we get a curve sequence

$$\boldsymbol{P}^k(t) = \sum_{i=0}^{n} \boldsymbol{P}_i^k B_i(t), \quad k = 0, 1, \ldots. \tag{7}$$

It has been shown in Lin et al. (2004, 2005) that

$$\lim_{k\to\infty} \boldsymbol{\Delta}_i^k = 0 \quad \text{and} \quad \lim_{k\to\infty} \boldsymbol{P}^k(t_i) = \boldsymbol{P}_i, \quad i = 0, 1, \ldots, n. \tag{8}$$

That is, the limit curve of $\{\boldsymbol{P}^k(t)\}$ interpolates the point sequence $\{\boldsymbol{P}_i \mid i = 0, 1, \ldots, n\}$. It is the so-called progressive-iterative approximation of a blending curve with normalized totally positive basis. Notably, the above progressive-iterative approximation format is *global*, since all of the control points need to be adjusted.

In this section, we will present a more flexible format, the *local* progressive-iterative approximation format, which adjusts only a subset of the control points, not all of them.

Suppose only the i_0th, i_1th, $\ldots$, i_Ith control points are adjusted, and the other control points, the j_0th, j_1th, $\ldots$, j_Jth, remain unchanged. That is, $\boldsymbol{P}_h^k = \boldsymbol{P}_h$, $h \in \{j_0, j_1, \ldots, j_J\}$, $k = 0, 1, \ldots$. Therefore, the vectors $\{\boldsymbol{\Delta}_i^k \mid i = 0, 1, \ldots, n,\ k = 0, 1, \ldots\}$ (Eq. (5)) fall into two classes, one for the adjusted control points, called *adjusting vectors*, and the other for the unchanged control points, called *difference vectors*.

On one hand, for the adjusting vector $\boldsymbol{\Delta}_l^{k+1}$, $l \in \{i_0, i_1, \ldots, i_I\}$, we have

$$\begin{aligned}
\boldsymbol{\Delta}_l^{k+1} &= \boldsymbol{P}_l - \sum_{i=0}^{n} \boldsymbol{P}_i^{k+1} B_i(t_l) \\
&= \boldsymbol{P}_l - \sum_{i \in \{j_0, j_1, \ldots, j_J\}} \boldsymbol{P}_i B_i(t_l) - \sum_{i \in \{i_0, i_1, \ldots, i_I\}} \left(\boldsymbol{P}_i^k + \boldsymbol{\Delta}_i^k\right) B_i(t_l) \\
&= \boldsymbol{P}_l - \sum_{i=0}^{n} \boldsymbol{P}_i^k B_i(t_l) - \boldsymbol{\Delta}_{i_0}^k B_{i_0}(t_l) - \boldsymbol{\Delta}_{i_1}^k B_{i_1}(t_l) - \cdots - \boldsymbol{\Delta}_{i_I}^k B_{i_I}(t_l) \\
&= \boldsymbol{\Delta}_l^k - \boldsymbol{\Delta}_{i_0}^k B_{i_0}(t_l) - \boldsymbol{\Delta}_{i_1}^k B_{i_1}(t_l) - \cdots - \boldsymbol{\Delta}_{i_I}^k B_{i_I}(t_l) \\
&= -\boldsymbol{\Delta}_{i_0}^k B_{i_0}(t_l) - \boldsymbol{\Delta}_{i_1}^k B_{i_1}(t_l) - \cdots + \left(1 - B_l(t_l)\right)\boldsymbol{\Delta}_l^k - \cdots - \boldsymbol{\Delta}_{i_I}^k B_{i_I}(t_l).
\end{aligned} \tag{9}$$

On the other hand, the difference vector $\boldsymbol{\Delta}_h^{k+1}$, $h \in \{j_0, j_1, \ldots, j_J\}$, corresponding to the unchanged control point $\boldsymbol{P}_h$ is

$$\begin{aligned}
\boldsymbol{\Delta}_h^{k+1} &= \boldsymbol{P}_h - \sum_{i=0}^{n} \boldsymbol{P}_i^{k+1} B_i(t_h) \\
&= \boldsymbol{P}_h - \sum_{i \in \{j_0, j_1, \ldots, j_J\}} \boldsymbol{P}_i B_i(t_h) - \sum_{i \in \{i_0, i_1, \ldots, i_I\}} \left(\boldsymbol{P}_i^k + \boldsymbol{\Delta}_i^k\right) B_i(t_h) \\
&= \boldsymbol{P}_h - \sum_{i=0}^{n} \boldsymbol{P}_i^k B_i(t_h) - \boldsymbol{\Delta}_{i_0}^k B_{i_0}(t_h) - \boldsymbol{\Delta}_{i_1}^k B_{i_1}(t_h) - \cdots - \boldsymbol{\Delta}_{i_I}^k B_{i_I}(t_h) \\
&= \boldsymbol{\Delta}_h^k - \boldsymbol{\Delta}_{i_0}^k B_{i_0}(t_h) - \boldsymbol{\Delta}_{i_1}^k B_{i_1}(t_h) - \cdots - \boldsymbol{\Delta}_{i_I}^k B_{i_I}(t_h).
\end{aligned} \tag{10}$$

Specifically, denote

$$\Delta^k = \left[\boldsymbol{\Delta}_{j_0}^k, \boldsymbol{\Delta}_{j_1}^k, \ldots, \boldsymbol{\Delta}_{j_J}^k, \boldsymbol{\Delta}_{i_0}^k, \boldsymbol{\Delta}_{i_1}^k, \ldots, \boldsymbol{\Delta}_{i_I}^k\right]^T, \tag{11}$$

the iterative format for Δ^k is

$$\Delta^{k+1} = C\Delta^k, \quad k = 0, 1, \ldots, \tag{12}$$

where C is the iterative matrix,

$$C = \begin{bmatrix} I & C_1 \\ 0 & C_2 \end{bmatrix}. \tag{13}$$

Here, I is the identity matrix of rank $(J+1) \times (J+1)$, and

$$C_1 = \begin{bmatrix} -B_{i_0}(t_{j_0}) & -B_{i_1}(t_{j_0}) & \cdots & -B_{i_I}(t_{j_0}) \\ -B_{i_0}(t_{j_1}) & -B_{i_1}(t_{j_1}) & \cdots & -B_{i_I}(t_{j_1}) \\ \cdots & \cdots & \cdots & \cdots \\ -B_{i_0}(t_{j_J}) & -B_{i_1}(t_{j_J}) & \cdots & -B_{i_I}(t_{j_J}) \end{bmatrix}, \tag{14}$$

$$C_2 = I - B_2, \tag{15}$$

where I is the identity matrix of rank $(I+1) \times (I+1)$, and

$$B_2 = \begin{bmatrix} B_{i_0}(t_{i_0}) & B_{i_1}(t_{i_0}) & \cdots & B_{i_I}(t_{i_0}) \\ B_{i_0}(t_{i_1}) & B_{i_1}(t_{i_1}) & \cdots & B_{i_I}(t_{i_1}) \\ \cdots & \cdots & \cdots & \cdots \\ B_{i_0}(t_{i_I}) & B_{i_1}(t_{i_I}) & \cdots & B_{i_I}(t_{i_I}) \end{bmatrix}. \tag{16}$$

Note that, the iterative matrix C (Eq. (13)) is a reducible matrix, so the adjusting vectors and difference vectors can be handled individually.

Theorem 1. *If the matrix B_2 (16) is nonsingular, the adjusted control points $\boldsymbol{P}_l^k$ tend to $\boldsymbol{P}_l$, $l \in \{i_0, i_1, \ldots, i_I\}$, when $k \Rightarrow \infty$. That is, $\boldsymbol{P}_l^k \Rightarrow \boldsymbol{P}_l$, $k \Rightarrow \infty$, $l \in \{i_0, i_1, \ldots, i_I\}$.*

Proof. For any $l \in \{i_0, i_1, \ldots, i_I\}$, since $\boldsymbol{\Delta}_l^k = \boldsymbol{P}_l - \boldsymbol{P}^k(t_l)$, we only need to show $\lim_{k\to\infty} \boldsymbol{\Delta}_l^k = 0$.

As stated above, since the iterative matrix C (Eq. (13)) is reducible, the adjusting vectors and difference vectors can be dealt with individually. If we denote

$$\Delta_I^k = \left[\boldsymbol{\Delta}_{i_0}^k, \boldsymbol{\Delta}_{i_1}^k, \ldots, \boldsymbol{\Delta}_{i_I}^k\right]^T,$$

we have

$$\Delta_I^{k+1} = C_2 \Delta_I^k. \tag{17}$$

That is, C_2 is the iterative matrix for the adjusting vectors.

From Eq. (15), we have $C_2 = I - B_2$. The nonsingular matrix B_2 (16) is a principal sub-matrix of the totally positive collocation matrix of the normalized totally positive basis $\{B_i(t) \mid i = 0, 1, \ldots, n\}$, so B_2 is a nonsingular totally positive matrix, and all of its eigenvalues satisfy $0 < \lambda(B_2) \leqslant 1$. Therefore, the spectral radius of the iterative matrix C_2, $\rho(C_2) = \rho(1 - B_2) < 1$, and then $\Delta_I^k \Rightarrow 0$, $k \Rightarrow \infty$. $\square$

Furthermore, in the following Theorem 2, we will deduce the limit vector of each difference vector $\boldsymbol{\Delta}_h^k$, $h \in \{j_0, j_1, \ldots, j_J\}$, when $k \Rightarrow \infty$. To do so, denote the eigenvalues of the matrix C_2 as $\lambda_0, \lambda_1, \ldots, \lambda_I$. Since B_2 (16) is a totally positive matrix, there exists the invertible matrix X, such that

$$C_2 = X^{-1} \operatorname{diag}(\lambda_0, \lambda_1, \ldots, \lambda_I) X,$$

where $\operatorname{diag}(\cdot)$ is the diagonal matrix. Additionally, denote

$$\Delta_J^k = \left[\boldsymbol{\Delta}_{j_0}^k, \boldsymbol{\Delta}_{j_1}^k, \ldots, \boldsymbol{\Delta}_{j_J}^k\right]^T. \tag{18}$$

Theorem 2. *The difference vector Δ_J^k (18) converges when $k \Rightarrow \infty$, and the limit vector is*

$$\Delta_J = \Delta_J^0 + D\Delta_I^0. \tag{19}$$

Here,

$$D = C_1 X^{-1} \operatorname{diag}\left(\frac{1}{1-\lambda_0}, \frac{1}{1-\lambda_1}, \ldots, \frac{1}{1-\lambda_I}\right) X,$$

where the matrix C_1 is presented in Eq. (14), and $0 \leqslant \lambda_i < 1$, $i = 0, 1, \ldots, I$.

Proof. It is obtained from Eq. (10) that, for any difference vector $\boldsymbol{\Delta}_h^k$, $h \in \{j_0, j_1, \ldots, j_J\}$, $k = 1, 2, \ldots,$

$$\begin{aligned}
\boldsymbol{\Delta}_h^k &= \boldsymbol{\Delta}_h^{k-1} - \boldsymbol{\Delta}_{i_0}^{k-1} B_{i_0}(t_h) - \boldsymbol{\Delta}_{i_1}^{k-1} B_{i_1}(t_h) - \cdots - \boldsymbol{\Delta}_{i_I}^{k-1} B_{i_I}(t_h) \\
&= \boldsymbol{\Delta}_h^0 - B_{i_0}(t_h) \sum_{j=0}^{k-1} \boldsymbol{\Delta}_{i_0}^j - B_{i_1}(t_h) \sum_{j=0}^{k-1} \boldsymbol{\Delta}_{i_1}^j - \cdots - B_{i_I}(t_h) \sum_{j=0}^{k-1} \boldsymbol{\Delta}_{i_I}^j.
\end{aligned}$$

Therefore,

$$\Delta_J^k = \Delta_J^0 + C_1 \sum_{j=0}^{k-1} \Delta_I^j. \tag{20}$$

On the other hand, from the proof of Theorem 1, we know that $C_2 = I - B_2$, and B_2 is a nonsingular totally positive matrix. So, there exists an invertible matrix X, such that

$$C_2 = X^{-1} \operatorname{diag}(\lambda_0, \lambda_1, \ldots, \lambda_I) X,$$

where $0 \leqslant \lambda_i < 1$, $i = 0, 1, \ldots, I$, are the eigenvalues of the matrix C_2. According to the iterative format (17) of Δ_I^k, it follows,

$$\Delta_I^j = C_2 \Delta_I^{j-1} = X^{-1} \operatorname{diag}\left(\lambda_0^j, \lambda_1^j, \ldots, \lambda_I^j\right) X \Delta_I^0, \quad j = 1, 2, \ldots.$$

Hence,

$$\sum_{j=0}^{k-1}\Delta_I^j = X^{-1}\sum_{j=0}^{k-1}\operatorname{diag}\big(\lambda_0^j,\lambda_1^j,\ldots,\lambda_I^j\big)X\Delta_I^0 = X^{-1}\operatorname{diag}\left(\frac{1-\lambda_0^k}{1-\lambda_0},\ldots,\frac{1-\lambda_I^k}{1-\lambda_I}\right)X\Delta_I^0. \tag{21}$$

Substituting the above equation (21) to Eq. (20), leads to

$$\Delta_J^k = \Delta_J^0 + C_1X^{-1}\operatorname{diag}\left(\frac{1-\lambda_0^k}{1-\lambda_0},\ldots,\frac{1-\lambda_I^k}{1-\lambda_I}\right)X\Delta_I^0.$$

Because $\lambda_i \in [0,1)$, $i=0,1,\ldots,I$, when $k \Rightarrow \infty$, Δ_J^k converges to

$$\Delta_J = \Delta_J^0 + C_1X^{-1}\operatorname{diag}\left(\frac{1}{1-\lambda_0},\ldots,\frac{1}{1-\lambda_I}\right)X\Delta_I^0.$$

This concludes the proof. □

Specifically, if only one control point $\boldsymbol{P}_l$ is adjusted, and all of the others remain unchanged, we have the corollary below.

Corollary 1. *If only one control point $\boldsymbol{P}_l$ is adjusted, the difference vector*

$$\lim_{k\to\infty}\boldsymbol{\Delta}_h^k = \boldsymbol{\Delta}_h^0 - \frac{B_l(t_h)}{B_l(t_l)}\boldsymbol{\Delta}_l^0, \quad h=0,1,\ldots,l-1,l+1,\ldots,n.$$

2.1. Rational curves

In this section, we will show that the local progressive-iterative approximation format for a rational curve,

$$\boldsymbol{T}(t) = \frac{\sum_{i=0}^n \omega_i\boldsymbol{P}_iB_i(t)}{\sum_{i=0}^n \omega_iB_i(t)}, \tag{22}$$

is also convergent. Here, $\boldsymbol{P}_i$ are the given data points (Eq. (2)), $B_i(t)$ are the normalized totally positive basis (Eq. (3)), and $\omega_i > 0$ are the weights endowed to the data points where $i=0,1,\ldots,n$.

By assigning each data point a parameter t_i, $i=0,1,\ldots,n$, as shown in Eq. (1), and introducing the homogeneous coordinates $\boldsymbol{Q}_i^0 = \boldsymbol{Q}_i = (\omega_i\boldsymbol{P}_i,\omega_i)$, $i=0,1,\ldots,n$, the following global progressive-iterative approximation format for a rational curve in homogeneous form can be constructed,

$$\begin{cases}\boldsymbol{R}^k(t) = \sum\limits_{j=0}^n \boldsymbol{Q}_j^kB_j(t),\\ \boldsymbol{\Delta}_i^k = \boldsymbol{Q}_i - \boldsymbol{R}^k(t_i),\\ \boldsymbol{Q}_i^{k+1} = \boldsymbol{Q}_i^k + \boldsymbol{\Delta}_i^k,\\ i=0,1,\ldots,n;\ k=0,1,\ldots.\end{cases} \tag{23}$$

The curve sequence $\{\boldsymbol{R}^k(t) \mid k=0,1,\ldots\}$ is called the *iterative rational curve sequence in homogeneous form*. Moreover, suppose $\boldsymbol{R}^k(t) = \{\omega^k(t)\boldsymbol{T}^k(t),\omega^k(t)\}$, where $\omega^k(t) = \sum_{i=0}^n\omega_i^kB_i(t)$. By projecting, it can be transformed into Euclidean coordinate form,

$$\boldsymbol{T}^k(t) = \frac{\sum_{i=0}^n \omega_i^k\boldsymbol{P}_i^kB_i(t)}{\sum_{i=0}^n \omega_i^kB_i(t)}, \quad k=0,1,\ldots, \tag{24}$$

called the *iterative rational curve sequence in Euclidean form*.

For simplicity, denote $\boldsymbol{R}_i^k = \boldsymbol{R}^k(t_i)$, $\boldsymbol{T}_i^k = \boldsymbol{T}^k(t_i)$, and $\Omega_i^k = \omega^k(t_i)$. We have,

Theorem 3. *The iterative rational curve sequence in homogeneous form* (*Eq.* (23)) *is convergent, namely,* $\lim_{k\to\infty}\boldsymbol{R}_i^k = \boldsymbol{Q}_i$, $i=0,1,\ldots,n$.

Because the iterative matrix of $\{\boldsymbol{R}^k(t) \mid k=0,1,\ldots\}$ is the same as that of the corresponding iterative format for a polynomial curve, the proof of Theorem 3 is very similar to that of the convergence of the global progressive-iterative format for a polynomial curve. For the details of the proof, please refer to Lin et al. (2004, 2005) and Shi and Wang (2006).

Furthermore, we have,

Theorem 4. $\lim_{k\to\infty} \boldsymbol{T}_i^k = \boldsymbol{P}_i$, *and* $\lim_{k\to\infty} \Omega_i^k = \omega_i$, $i = 0, 1, \ldots, n$.

The above theorem means that, in Euclidean space, the rational curve sequence $\{\boldsymbol{T}^k(t)\}$ will interpolate the data point $\boldsymbol{P}_i$ at the parameter t_i, and $\omega^k(t_i)$ tends to ω_i when $k \Rightarrow \infty$, where $i = 0, 1, \ldots, n$. The result of the above theorem is proved in Shi and Wang (2006), but to cubic NURBS curve and surface. However, the proof can be employed to prove the general result in the above theorem without any modification. For the sake of integrality, we append the proof in Appendix A.

Similarly, the local progressive-iterative approximation format for the rational curve can be defined, by allowing only a subset of the control points $\boldsymbol{Q}_l^k$, $l \in \{i_0, i_1, \ldots, i_I\}$, $k = 0, 1, \ldots$, to be adjusted, and the other control points unchanged. The result below is the direct corollary of Theorems 1, 3, 4.

Corollary 2. *For any adjusted control point* $\boldsymbol{Q}_l^k$, $l \in \{i_0, i_1, \ldots, i_I\}$, $k = 0, 1, \ldots$, $\lim_{k\to\infty} \boldsymbol{R}_l^k = \boldsymbol{Q}_l$, $\lim_{k\to\infty} \boldsymbol{T}_l^k = \boldsymbol{P}_l$, *and* $\lim_{k\to\infty} \Omega_l^k = \omega_l$.

3. Local PIA format for a blending patch

In this section, we develop the local PIA format for a blending patch, and study its convergence. At last, we show that if the local PIA format for a polynomial patch is convergent, so is the format for the corresponding rational patch. For clarity in developing the local PIA format, we need to first introduce the global PIA format (Lin et al., 2004, 2005).

3.1. Global PIA format

Given an ordered point set $\{\boldsymbol{P}_{ij} \in \mathbb{R}^3\}_{i=0}^{m}{}_{j=0}^{n}$, each point $\boldsymbol{P}_{ij}$ is assigned a parameter value (u_i, v_j), $i = 0, 1, \ldots, m$, $j = 0, 1, \ldots, n$, satisfying

$$u_0 < u_1 < \cdots < u_m, \qquad v_0 < v_1 < \cdots < v_n. \tag{25}$$

With the control points

$$\left\{\boldsymbol{P}_{ij}^0 = \boldsymbol{P}_{ij}\right\}_{i=0}^{m}{}_{j=0}^{n}, \tag{26}$$

an initial surface can be constructed, that is,

$$\boldsymbol{S}^0(u, v) = \sum_{i=0}^{m}\sum_{j=0}^{n} \boldsymbol{P}_{ij}^0 B_i(u) B_j(v), \tag{27}$$

where $B_i(u), B_j(v)$ are basis functions.

By adjusting the control points $\boldsymbol{P}_{ij}^k, k = 1, 2, \ldots$, along the vectors,

$$\boldsymbol{\Delta}_{ij}^k = \boldsymbol{P}_{ij} - \boldsymbol{S}^k(u_i, v_j), \tag{28}$$

that is,

$$\boldsymbol{P}_{ij}^{k+1} = \boldsymbol{P}_{ij}^k + \boldsymbol{\Delta}_{ij}^k, \tag{29}$$

we get a surface sequence

$$\boldsymbol{S}^k(u, v) = \sum_{i=0}^{m}\sum_{j=0}^{n} \boldsymbol{P}_{ij}^k B_i(u) B_j(v), \quad k = 0, 1, \ldots. \tag{30}$$

The global progressive-iterative approximation (PIA) property (Lin et al., 2004, 2005) of the blending patch (27) is that

$$\begin{aligned}&\lim_{k\to\infty} \boldsymbol{\Delta}_{ij}^k = 0, \quad \text{or equivalently,}\\ &\lim_{k\to\infty} \boldsymbol{S}^k(u_i, v_j) = \boldsymbol{P}_{ij}, \quad i = 0, 1, \ldots, m,\ j = 0, 1, \ldots, n.\end{aligned} \tag{31}$$

That is, the limit surface of $\{\boldsymbol{S}^k(u, v)\}$ interpolates the point set $\{\boldsymbol{P}_{ij} \mid i = 0, 1, \ldots, m,\ \ j = 0, 1, \ldots, n\}$.

Specifically, denote

$$\Delta^l = \left[\boldsymbol{\Delta}_{00}^l, \boldsymbol{\Delta}_{01}^l, \ldots, \boldsymbol{\Delta}_{0n}^l, \boldsymbol{\Delta}_{11}^l, \ldots, \boldsymbol{\Delta}_{1n}^l, \ldots, \boldsymbol{\Delta}_{m1}^l, \ldots, \boldsymbol{\Delta}_{mn}^l\right]^T, \quad l = k, k+1, \tag{32}$$

the iterative format for Δ^l is

$$\Delta^{k+1} = G\Delta^k, \quad G = I - D,\ k = 0, 1, \ldots, \tag{33}$$

where I is the identity matrix; the collocation matrix D is the Kronecker product of D_1 and D_2, namely,

$$D = D_1 \otimes D_2, \tag{34}$$

with

$$D_1 = \begin{bmatrix} B_0(u_0) & B_1(u_0) & \cdots & B_m(u_0) \\ B_0(u_1) & B_1(u_1) & \cdots & B_m(u_1) \\ \cdots & \cdots & \cdots & \cdots \\ B_0(u_m) & B_1(u_m) & \cdots & B_m(u_m) \end{bmatrix},$$
$$D_2 = \begin{bmatrix} B_0(v_0) & B_1(v_0) & \cdots & B_n(v_0) \\ B_0(v_1) & B_1(v_1) & \cdots & B_n(v_1) \\ \cdots & \cdots & \cdots & \cdots \\ B_0(v_n) & B_1(v_n) & \cdots & B_n(v_n) \end{bmatrix}. \tag{35}$$

If the basis is normalized totally positive, then D_1 and D_2 are stochastic and totally positive matrices. Thus, if D_1 and D_2 are nonsingular, the spectral radius of the matrix D is in the interval $(0, 1]$, so the iterative format (33) is convergent to zero, meaning that the limit surface interpolates the given point set $\{\boldsymbol{P}_{ij} \in \mathbb{R}^3\}_{i=0}^{m}{}_{j=0}^{n}$ (Lin et al., 2004, 2005).

3.2. Local PIA format

Now, we can present the local progressive-iterative approximation (PIA) format. The local PIA format is more flexible than the global one, which permits only a part of the control points to be adjusted, not the whole. Thus, the fitting can be performed adaptively. That is, if the fitting precision at some data points does not achieve the pre-defined threshold, the control points corresponding to these data points can be adjusted by the local PIA format to improve the fitting precision, while other control points remain unchanged. Therefore, the local PIA format can save the computational resources significantly, especially when the number of the data points is huge.

Suppose only the control points with indices $(k_0, l_0), (k_1, l_1), \ldots, (k_J, l_J)$ are adjusted, and other control points remain unchanged. To present the local progressive-iterative approximation (PIA) format, we re-arrange Δ^l (32) as

$$\Delta^l = \left[\boldsymbol{\Delta}^l_{i_0,j_0}, \boldsymbol{\Delta}^l_{i_1,j_1}, \ldots, \boldsymbol{\Delta}^l_{i_I,j_I}, \boldsymbol{\Delta}^l_{k_0,l_0}, \boldsymbol{\Delta}^l_{k_1,l_1}, \ldots, \boldsymbol{\Delta}^l_{k_J,l_J}\right]^T, \quad l = k, k+1. \tag{36}$$

Here, $\boldsymbol{\Delta}^l_{i_0,j_0}, \ldots, \boldsymbol{\Delta}^l_{i_I,j_I}$ correspond to the unchanged control points, called *difference vectors*, while $\boldsymbol{\Delta}^l_{k_0,l_0}, \ldots, \boldsymbol{\Delta}^l_{k_J,l_J}$ correspond to the adjusted control points, called *adjusting vectors*. They are arranged in row-major order, respectively. In the following, adjusting vectors and difference vectors will be studied separately.

First, the iterative format for the adjusting vector $\boldsymbol{\Delta}^l_{k_r,l_r}$, $r = 0, 1, \ldots, J$, is

$$\begin{aligned}
\boldsymbol{\Delta}^{k+1}_{k_r,l_r} &= \boldsymbol{P}_{k_r,l_r} - \boldsymbol{S}^{k+1}(u_{k_r}, v_{l_r}) \\
&= \boldsymbol{P}_{k_r,l_r} - \sum_{i=0}^{m}\sum_{j=0}^{n} \boldsymbol{P}^k_{ij} B_i(u_{k_r}) B_j(v_{l_r}) - \boldsymbol{\Delta}^k_{k_0,l_0} B_{k_0}(u_{k_r}) B_{l_0}(v_{l_r}) \\
&\quad - \cdots - \boldsymbol{\Delta}^k_{k_J,l_J} B_{k_J}(u_{k_r}) B_{l_J}(v_{l_r}) \\
&= -\boldsymbol{\Delta}^k_{k_0,l_0} B_{k_0}(u_{k_r}) B_{l_0}(v_{l_r}) - \cdots - \big(B_{k_r}(u_{k_r}) B_{l_r}(v_{l_r}) - 1\big)\boldsymbol{\Delta}^k_{k_r,l_r} \\
&\quad - \cdots - \boldsymbol{\Delta}^k_{k_J,l_J} B_{k_J}(u_{k_r}) B_{l_J}(v_{l_r}).
\end{aligned} \tag{37}$$

Second, the iterative format for the difference vector $\boldsymbol{\Delta}^l_{i_r,j_r}$, $r = 0, 1, \ldots, I$, is

$$\begin{aligned}
\boldsymbol{\Delta}^{k+1}_{i_r,j_r} &= \boldsymbol{P}_{i_r,j_r} - \boldsymbol{S}^{k+1}(u_{i_r}, v_{j_r}) \\
&= \boldsymbol{P}_{i_r,j_r} - \sum_{i=0}^{m}\sum_{j=0}^{n} \boldsymbol{P}^k_{ij} B_i(u_{i_r}) B_j(v_{j_r}) - \boldsymbol{\Delta}^k_{k_0,l_0} B_{k_0}(u_{i_r}) B_{l_0}(v_{j_r}) \\
&\quad - \cdots - \boldsymbol{\Delta}^k_{k_J,l_J} B_{k_J}(u_{i_r}) B_{l_J}(v_{j_r}) \\
&= \boldsymbol{\Delta}^k_{i_r,j_r} - \boldsymbol{\Delta}^k_{k_0,l_0} B_{k_0}(u_{i_r}) B_{l_0}(v_{j_r}) - \cdots - \boldsymbol{\Delta}^k_{k_J,l_J} B_{k_J}(u_{i_r}) B_{l_J}(v_{j_r}).
\end{aligned} \tag{38}$$

Combining Eqs. (37) and (38), we can construct the iterative format for Δ^l, $l = k, k+1$ (36), that is,

$$\Delta^{k+1} = C\Delta^k, \quad k = 0, 1, 2, \ldots, \tag{39}$$

where C is the iterative matrix,

$$C = \begin{bmatrix} I & C_1 \\ 0 & C_2 \end{bmatrix}. \tag{40}$$

Here, I is the identity matrix of rank $(I+1) \times (I+1)$, and

$$C_1 = \begin{bmatrix} -B_{k_0}(u_{i_0})B_{l_0}(v_{j_0}) & -B_{k_1}(u_{i_0})B_{l_1}(v_{j_0}) & \cdots & -B_{k_J}(u_{i_0})B_{l_J}(v_{j_0}) \\ -B_{k_0}(u_{i_1})B_{l_0}(v_{j_1}) & -B_{k_1}(u_{i_1})B_{l_1}(v_{j_1}) & \cdots & -B_{k_J}(u_{i_1})B_{l_J}(v_{j_1}) \\ \cdots & \cdots & \cdots & \cdots \\ -B_{k_0}(u_{i_I})B_{l_0}(v_{j_I}) & -B_{k_1}(u_{i_I})B_{l_1}(v_{j_I}) & \cdots & -B_{k_J}(u_{i_I})B_{l_J}(v_{j_I}) \end{bmatrix},$$

$$C_2 = \begin{bmatrix} 1-B_{k_0}(u_{k_0})B_{l_0}(v_{l_0}) & -B_{k_1}(u_{k_0})B_{l_1}(v_{l_0}) & \cdots & -B_{k_J}(u_{k_0})B_{l_J}(v_{l_0}) \\ -B_{k_0}(u_{k_1})B_{l_0}(v_{k_1}) & 1-B_{k_1}(u_{k_1})B_{l_1}(v_{l_1}) & \cdots & -B_{k_J}(u_{k_1})B_{l_J}(v_{l_1}) \\ \cdots & \cdots & \cdots & \cdots \\ -B_{k_0}(u_{k_J})B_{l_0}(v_{l_J}) & -B_{k_1}(u_{k_J})B_{l_1}(v_{l_J}) & \cdots & 1-B_{k_J}(u_{k_J})B_{l_J}(v_{l_J}) \end{bmatrix}. \tag{41}$$

Note that the iterative matrix (40) is reducible, so we can handle the adjusting vectors and difference vectors individually. Specifically, the iterative format for the adjusting vectors is

$$\Delta_J^{k+1} = C_2 \Delta_J^k = (I - E)\Delta_J^k, \tag{42}$$

where $\Delta_J^l = [\mathbf{\Delta}_{k_0,l_0}^l, \mathbf{\Delta}_{k_1,l_1}^l, \ldots, \mathbf{\Delta}_{k_J,l_J}^l]$, $l = k, k+1$, I is the identity matrix of rank $(J+1) \times (J+1)$, and

$$E = \begin{bmatrix} B_{k_0}(u_{k_0})B_{l_0}(v_{l_0}) & B_{k_1}(u_{k_0})B_{l_1}(v_{l_0}) & \cdots & B_{k_J}(u_{k_0})B_{l_J}(v_{l_0}) \\ B_{k_0}(u_{k_1})B_{l_0}(v_{k_1}) & B_{k_1}(u_{k_1})B_{l_1}(v_{l_1}) & \cdots & B_{k_J}(u_{k_1})B_{l_J}(v_{l_1}) \\ \cdots & \cdots & \cdots & \cdots \\ B_{k_0}(u_{k_J})B_{l_0}(v_{l_J}) & B_{k_1}(u_{k_J})B_{l_1}(v_{l_J}) & \cdots & B_{k_J}(u_{k_J})B_{l_J}(v_{l_J}) \end{bmatrix}. \tag{43}$$

In fact, the matrix E (43) is a principal sub-matrix of the matrix D (34). If the first row of E is in the ith row of D, the first column of E is in the ith column of D, and so on.

The next section studies the convergence of the local PIA format.

3.3. Convergence

Theorem 5. *The local progressive-iterative approximation* (*PIA*) *format* (42) *is convergent, if* $0 < \lambda(E) \leqslant 1$, *where* $\lambda(E)$ *denotes the eigenvalue of* E.

The proof to Theorem 5 is straightforward, because $0 < \lambda(E) \leqslant 1$ is equivalent to $0 \leqslant \lambda(C_2) < 1$ (42).

In Lin et al. (2004, 2005), it has been shown that, if the basis is normalized totally positive, then D_1 and D_2 (35) are stochastic and totally positive matrices. Thus, if both D_1 and D_2 are nonsingular, the spectral radius of the matrix D (34) is in the interval $(0, 1]$, so the global PIA format (33) is convergent to zero.

However, since the Kronecker product D (34) of two totally positive matrices D_1 and D_2 (35) is no longer totally positive, it is not clear whether the eigenvalue of its principal sub-matrix E (43) is still in the interval $(0, 1]$. Fortunately, in Section 4, we show that, to the most commonly used bi-cubic B-spline patch, $0 < \lambda(E) \leqslant 1$, and then the local iterative format (42) is convergent.

In the following, we will show what will happen if the local PIA format (42) is convergent.

In fact, if the matrix E (43) can be diagonalized, so is C_2 (41). That is, there exists a nonsingular matrix X, such that

$$C_2 = X^{-1} \operatorname{diag}(\lambda_0, \lambda_1, \ldots, \lambda_J) X,$$

where λ_i, $i = 0, 1, \ldots, J$, are the eigenvalues of C_2 (41).

Denote the difference vector set by $\Delta_I^l = [\mathbf{\Delta}_{i_0,j_0}^l, \mathbf{\Delta}_{i_1,j_1}, \ldots, \mathbf{\Delta}_{i_I,j_I}]$, we have,

Theorem 6. *If the local PIA format* (42) *is convergent, and the matrix* C_2 (41) *can be diagonalized, the difference vectors* Δ_I^l *converge when* $k \Rightarrow \infty$, *and the limit vectors are*

$$\Delta_I = \Delta_I^0 + F\Delta_J^0. \tag{44}$$

Here,

$$F = C_1 X^{-1} \operatorname{diag}\left(\frac{1}{1-\lambda_0}, \frac{1}{1-\lambda_1}, \ldots, \frac{1}{1-\lambda_J}\right) X,$$

where the matrix C_1 *is presented in Eq.* (41), *and* $0 \leqslant \lambda_i < 1$, $i = 0, 1, \ldots, J$, *are the eigenvalues of* C_2 (41).

The proof of this theorem is similar to that of Theorem 2.

Specifically, if only one control point $\boldsymbol{P}_{i_0,j_0}$ is adjusted, and all of the others remain unchanged, we have the corollary below.

Corollary 3. *If only one control point $\boldsymbol{P}_{i_0,j_0}$ is adjusted, the difference vector*

$$\lim_{k\to\infty} \boldsymbol{\Delta}_{ij}^k = \boldsymbol{\Delta}_{ij}^0 - \frac{B_{i_0}(u_i)B_{j_0}(v_j)}{B_{i_0}(u_{i_0})B_{j_0}(v_{j_0})}\boldsymbol{\Delta}_{i_0,j_0}^0.$$

3.4. Rational case

Furthermore, we will study the local PIA format for a rational patch, that is,

$$\boldsymbol{T}(u,v) = \frac{\sum_{i=0}^m \sum_{j=0}^n \omega_{ij}\boldsymbol{P}_{ij}B_i(u)B_j(v)}{\sum_{i=0}^m \sum_{j=0}^n \omega_{ij}B_i(u)B_j(v)}, \tag{45}$$

whose homogeneous form is

$$\boldsymbol{R}(u,v) = \sum_{i=0}^m \sum_{j=0}^n \boldsymbol{Q}_{ij}B_i(u)B_j(v). \tag{46}$$

Here, $\boldsymbol{Q}_{ij} = (\omega_{ij}\boldsymbol{P}_{ij}, \omega_{ij})$, $\boldsymbol{P}_{ij}$ are the given data points (Eq. (26)), and $\omega_{ij} > 0$ are the weights endowed to the data points, $i = 0, 1, \ldots, m$, $j = 0, 1, \ldots, n$.

As stated above, suppose only $J+1$ control points are adjusted, that is, $\boldsymbol{Q}_{k_r,l_r}$, $r = 0, 1, \ldots, J$, we have the local PIA format for the rational patch (46),

$$\begin{cases} \boldsymbol{R}^k(u,v) = \sum_{i=0}^m \sum_{j=0}^n \boldsymbol{Q}_{ij}^k B_i(u)B_j(v), \\ \boldsymbol{\Delta}_{ij}^k = \boldsymbol{Q}_{ij} - \boldsymbol{R}^k(u_i,v_j), \\ \boldsymbol{Q}_{k_r,l_r}^{k+1} = \boldsymbol{Q}_{k_r,l_r}^k + \boldsymbol{\Delta}_{k_r,l_r}^k, \\ i = 0,1,\ldots,m,\ j = 0,1,\ldots,n,\ r = 0,1,\ldots,J,\ k = 0,1,\ldots, \end{cases} \tag{47}$$

where $\boldsymbol{Q}_{ij}^0 = \boldsymbol{Q}_{ij}$, $i = 0, 1, \ldots, m$, $j = 0, 1, \ldots, n$.

It should be pointed out that the local PIA format and iterative matrix for the rational patch in homogeneous form (47) are the same as Eqs. (39) and (40), respectively. Thus, if denote

$$\begin{cases} \Omega_{k_r,l_r}^k = \sum_{i=0}^m \sum_{j=0}^n \omega_{ij}^k B_i(u_{k_r})B_j(v_{l_r}), \\ \boldsymbol{R}_{k_r,l_r}^k = \boldsymbol{R}^k(u_{k_r}, v_{l_r}), \\ \boldsymbol{T}_{k_r,l_r}^k = \boldsymbol{T}^k(u_{k_r}, v_{l_r}), \\ r = 0,1,\ldots,J, \end{cases}$$

we have,

Theorem 7. *If $0 < \lambda(E) \leqslant 1$ (43), where $\lambda(E)$ denotes the eigenvalue of E, the local PIA for the rational patch in homogeneous form (Eq. (47)) is convergent, namely, $\lim_{k\to\infty} \boldsymbol{R}_{k_r,l_r}^k = \boldsymbol{Q}_{k_r,l_r}$; furthermore, $\lim_{k\to\infty} \boldsymbol{T}_{k_r,l_r}^k = \boldsymbol{P}_{k_r,l_r}$, and $\lim_{k\to\infty} \Omega_{k_r,l_r}^k = \omega_{k_r,l_r}$, $r = 0, 1, \ldots, J$.*

For the proof of the above theorem, please refer to Shi and Wang (2006) or Appendix A.

4. Convergence of the local PIA format for the bi-cubic B-spline patch

In this section, we will show that the local progressive-iterative approximation (PIA) format (42) is convergent for bi-cubic B-spline patches.

Refer to Lin et al. (2004), suppose the collocation matrices D_1 and D_2 of the clamped non-uniform cubic B-spline basis are:

$$D_1 = \begin{bmatrix} a_1 & 0 & & & \\ c_1 & a_2 & b_2 & & \\ & \ddots & \ddots & \ddots & \\ & & c_{m-2} & a_{m-1} & b_{m-1} \\ & & & 0 & a_m \end{bmatrix} = \begin{bmatrix} a_1 & 0 & 0 \\ \boldsymbol{C}_1 & \widehat{D}_1 & \boldsymbol{B}_{m-1} \\ 0 & 0 & a_m \end{bmatrix},$$

$$D_2 = \begin{bmatrix} e_1 & 0 & & & \\ g_1 & e_2 & f_2 & & \\ & \ddots & \ddots & \ddots & \\ & & g_{n-2} & e_{n-1} & f_{n-1} \\ & & & 0 & e_n \end{bmatrix} = \begin{bmatrix} e_1 & 0 & 0 \\ \boldsymbol{G}_1 & \widehat{D}_2 & \boldsymbol{F}_{m-1} \\ 0 & 0 & e_n \end{bmatrix}, \tag{48}$$

respectively, which are nonsingular, stochastic, and totally positive matrices. Then, the global PIA iterative matrix for bi-cubic B-spline patch is $D = D_1 \otimes D_2$ (Lin et al., 2004).

Moreover, since the boundary control points of the clamped bi-cubic B-spline patch do not affect its inside, the boundary and inner control points can be considered independently. We have the following lemma.

Lemma 1. *The eigenvalues of the principal sub-matrix of the matrix $\widehat{D} = \widehat{D}_1 \otimes \widehat{D}_2$ are all real and in the interval* $(0, 1]$.

Proof. Since both matrices $\widehat{D}_1$ and $\widehat{D}_2$ (48) are nonsingular and totally positive matrices, and $\|\widehat{D}_1\|_2 = \|\widehat{D}_2\|_2 = 1$, the eigenvalues of the two matrices are all in the interval $(0, 1]$. Therefore, the eigenvalues of the matrix $\widehat{D}$, which is the Kronecker product of $\widehat{D}_1$ and $\widehat{D}_2$, are also in $(0, 1]$.

Note that both matrices $\widehat{D}_1$ and $\widehat{D}_2$ are tri-diagonal matrices, so they are similar to symmetric matrices, with invertible diagonal transformation matrices Ω and Φ as follows, respectively (Chen and Chen, 2000),

$$\Omega = \operatorname{diag}(\omega_2, \ldots, \omega_{m-1}), \qquad \Phi = \operatorname{diag}(\phi_2, \ldots, \phi_{n-1}),$$

where

$$\omega_i = \sqrt{\frac{b_{i-1}}{c_{i-1}}}\omega_{i-1}, \quad i = 3, \ldots, m-1,\ \omega_2 = 1,$$

$$\phi_j = \sqrt{\frac{f_{j-1}}{g_{j-1}}}\phi_{j-1}, \quad j = 3, \ldots, n-1,\ \phi_2 = 1.$$

Therefore, if we denote

$$K = \Omega \otimes \Phi = \begin{bmatrix} \omega_2 \Phi & & & \\ & \omega_3 \Phi & & \\ & & \ddots & \\ & & & \omega_{m-1}\Phi \end{bmatrix},$$

$\widetilde{D} = K\widehat{D}K^{-1}$ is a symmetric matrix, namely, $\widehat{D}$ is similar to a symmetric matrix. Furthermore, since K is a diagonal matrix, any principal sub-matrix of $\widehat{D}$, that is, $\widehat{D}([i_1, \ldots, i_k], [i_1, \ldots, i_k])$, is also similar to the principal sub-matrix of $\widetilde{D} = K\widehat{D}K^{-1}$, that is, $\widetilde{D}([i_1, \ldots, i_k], [i_1, \ldots, i_k])$, with the invertible diagonal transformation matrix $K([i_1, \ldots, i_k], [i_1, \ldots, i_k])$.

Moreover, as the eigenvalues of the matrix $\widehat{D}$ are all in $(0, 1]$, the symmetric matrix $\widetilde{D}$ is positive definite. Hence, all of the principal sub-matrices of $\widetilde{D}$ are positive definite, and then, all of its eigenvalues are positive. So are the eigenvalues of the principal sub-matrix of $\widehat{D}$. Together with the fact that $\|\widehat{D}\|_2 = 1$, we show the result. □

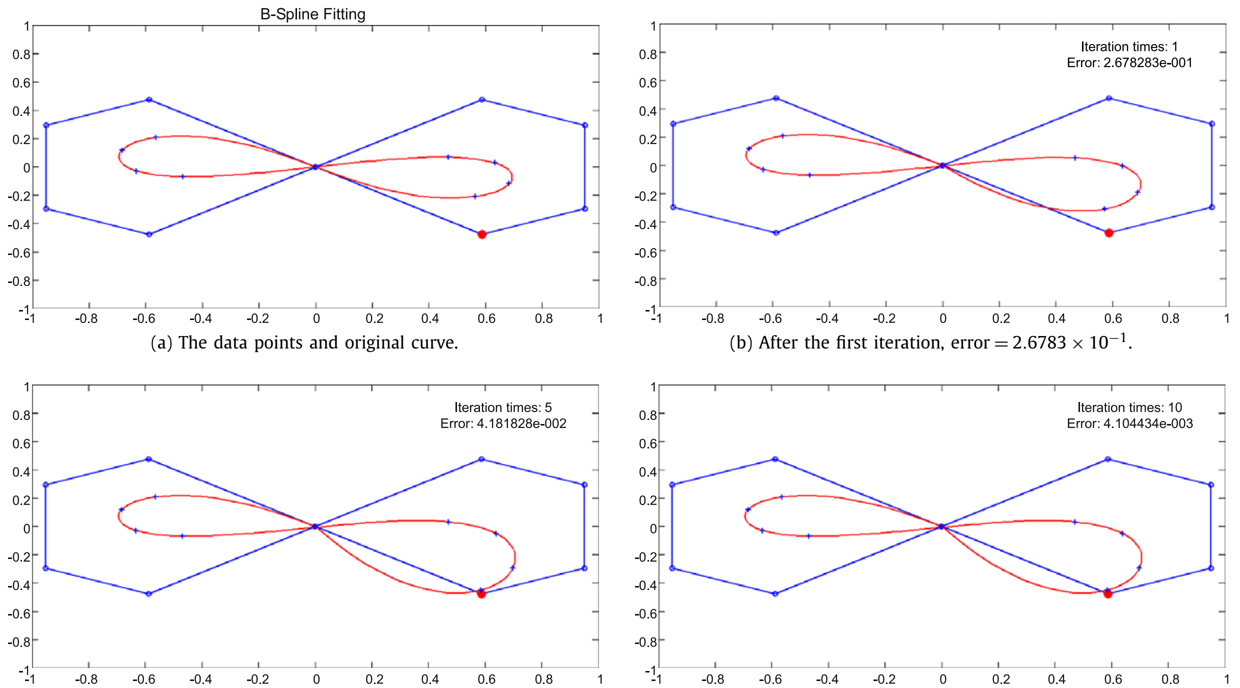
Now, we can present the main result of this section.

Theorem 8. *The local progressive-iterative approximation* (*PIA*) *format* (42) *for a bi-cubic B-spline patch is convergent.*

Proof. If the adjusted control points are all inner control points of a bi-cubic B-spline patch, the matrix E in the iterative format (42) is a principal sub-matrix of the matrix $\widehat{D}$ in Lemma 1. By Lemma 1, the eigenvalue of E satisfies $0 < \lambda(E) \leqslant 1$. Based on Theorem 5, the local iterative format is convergent.

Furthermore, if the adjusted control points include both boundary and inner control points of a bi-cubic B-spline patch, they can be considered independently, as stated above. First, the convergence of the adjusting vectors corresponding to the inner control points is ensured by the above analysis. Second, the adjustment of the boundary points is just a local PIA format for a cubic B-spline curve, so it is also convergent. □

As a corollary of Theorems 7 and 8, the local PIA format for a bi-cubic NURBS patch is convergent.



(a) The data points and original curve.

(b) After the first iteration, error $= 2.6783 \times 10^{-1}$.

(c) After the fifth iteration, error $= 4.1818 \times 10^{-2}$.

(d) After the tenth iteration, error $= 4.1044 \times 10^{-3}$.

Fig. 1. Approximate one point (red dot) with Bézier curve by local progressive-iterative approximation format. (For interpretation of colors in this figure, the reader is referred to the web version of this article.)

5. Convergence of the local PIA format for a generic blending patch

In this section, we will show that the local PIA format (42) is convergent for a generic blending patch with normalized totally positive basis, if the adjusted control points are altered row by row, or column by column.

Theorem 9. *If the adjusted control points lie in the same row (or column), and the corresponding collocation matrix is nonsingular, the local PIA format (42) will be convergent for any blending patch with normalized totally positive basis.*

Proof. Without loss of generality, suppose the adjusted control points lie in the ith row, that is, $\boldsymbol{P}_{i,j_0}, \boldsymbol{P}_{i,j_1}, \ldots, \boldsymbol{P}_{i,j_J}$, $j_0 < j_1 < \cdots < j_J$.

In this case, the iterative matrix E (43) can be re-written as

$$E = \left[B_i(u_i)\right] \otimes \begin{bmatrix} B_{j_0}(v_{j_0}) & \cdots & B_{j_J}(v_{j_0}) \\ \cdots & \cdots & \cdots \\ B_{j_0}(v_{j_J}) & \cdots & B_{j_J}(v_{j_J}) \end{bmatrix},$$

where $\otimes$ denotes the Kronecker product.

Since the basis of the blending patch is normalized totally positive, together with the property of the Kronecker product, the eigenvalues of the nonsingular matrix E are in $(0, 1]$. Based on Theorem 5, the local format is convergent.

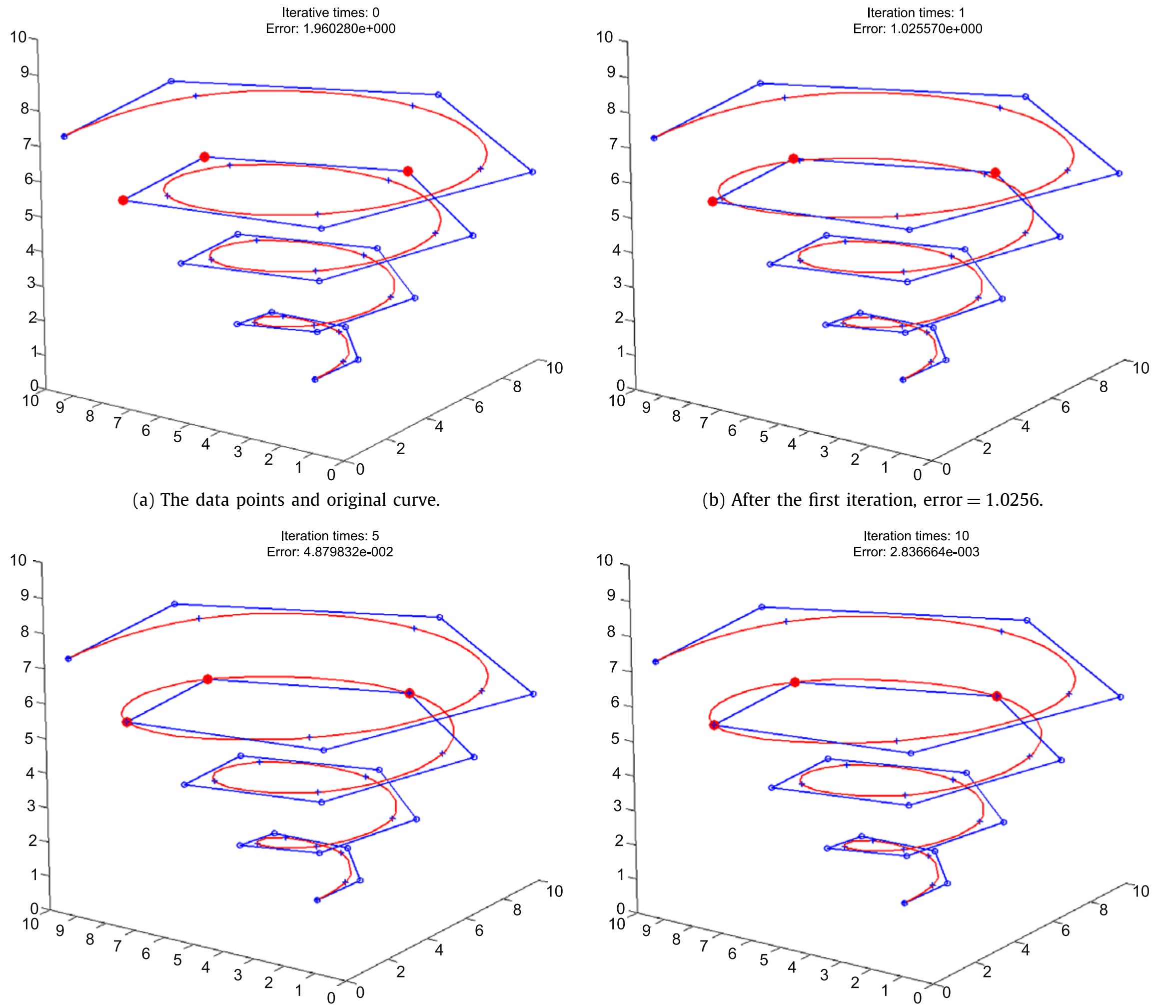
Similarly, the result is true if the adjusted control points lie in the same column. $\square$

Therefore, if a part of the control points of a generic blending patch with normalized totally positive basis is adjusted to fit a data point set, they can be adjusted row by row (or column by column). That is, we first alter one row of the adjusted control points by the local PIA format till the fitting precision satisfies the user's requirement; then, another row of adjusted control points are changed iteratively; and so on. Finally, all adjusted control points approximate the corresponding data points with the pre-defined precision.

6. Results and discussion

In this section, some examples are presented to illustrate the local approximation capability of the local PIA format. We take the fitting error for the case of curve as

$$\text{error} = \sum_{i \in \{i_0, i_1, \ldots, i_I\}} \left\| \boldsymbol{P}_i - \boldsymbol{P}^k(t_i) \right\|,$$



(a) The data points and original curve.

(b) After the first iteration, error = 1.0256.

(c) After the fifth iteration, error $= 4.8798 \times 10^{-2}$.

(d) After the tenth iteration, error $= 2.8366 \times 10^{-3}$.

Fig. 2. Approximate three continuous points (red dots) with cubic B-spline curve by local progressive-iterative approximation format. (For interpretation of colors in this figure, the reader is referred to the web version of this article.)

where $\{i_0, i_1, \ldots, i_I\}$ are the indices of the adjusted control points, and the fitting error for the patch as

$$\text{error} = \max\left\{\left\|\boldsymbol{P}_{ij} - \boldsymbol{P}^k(u_i, v_j)\right\| \;\middle|\; (i,j) \in \left\{(k_0,l_0),(k_1,l_1),\ldots,(k_J,l_J)\right\}\right\},$$

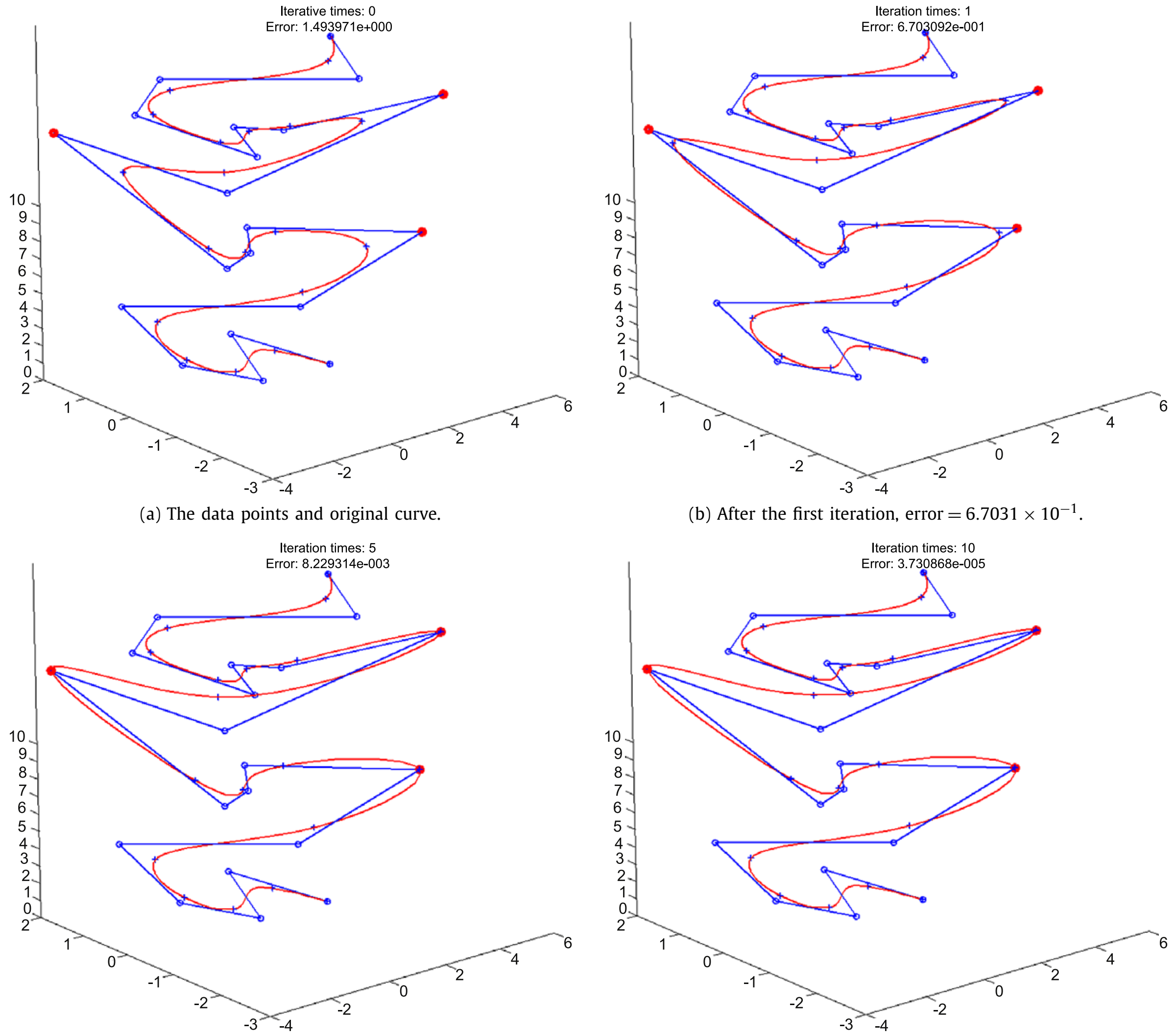
where $\{(k_0,l_0),(k_1,l_1),\ldots,(k_J,l_J)\}$ are the indices of the adjusted control points. Fig. 1 to Fig. 3 demonstrate the examples for curves, and Fig. 4 to Fig. 7 are the examples for patches. All examples are implemented with Matlab, and run on the PC with 2.83 GHz CPU and 3.25 GB memory.

A Bézier curve is a blending curve with Bernstein basis, which is totally positive. It has been shown in Lin et al. (2005) that, if the collocation matrix of the Bernstein basis is nonsingular, the global progressive-iterative format is convergent. Furthermore, based on Theorems 1 and 2 in Section 2, the local progressive-iterative approximation format of the Bézier curve is also convergent. Due to the global support property of the Bernstein basis function, adjusting even one control point except the head and tail one will influence the whole curve. Fig. 1 illustrates an example where only one control point is adjusted. The eleven data points are sampled from the curve Lemniscate of Gerono,

$$\begin{cases} x(t) = \cos(t), \\ y(t) = \sin(t)\cos(t), \end{cases}$$

at $t_i = -\frac{\pi}{2} + i\frac{2\pi}{10}$, $i = 0, 1, \ldots, 10$.

On the other hand, a B-spline curve is a blending curve with B-spline basis. Here, the B-spline curve is clamped. The collocation matrix of the B-spline basis at the knots is nonsingular and totally positive. Therefore, both the global and local progressive-iterative formats are convergent. Differing from the Bézier curve, adjusting a control point of a B-spline curve only influences several curve segments, not the whole. Fig. 2 demonstrates the local progressive-iterative approximation of



(a) The data points and original curve.

(b) After the first iteration, error $= 6.7031 \times 10^{-1}$.

(c) After the fifth iteration, error $= 8.2293 \times 10^{-3}$.

(d) After the tenth iteration, error $= 3.7309 \times 10^{-5}$.

Fig. 3. Approximate three separated points (red dots) with cubic NURB curve by local progressive-iterative approximation format. (For interpretation of colors in this figure, the reader is referred to the web version of this article.)

a clamped cubic B-spline curve, where three continuous control points are adjusted. In this example, twenty data points are sampled from the helix,

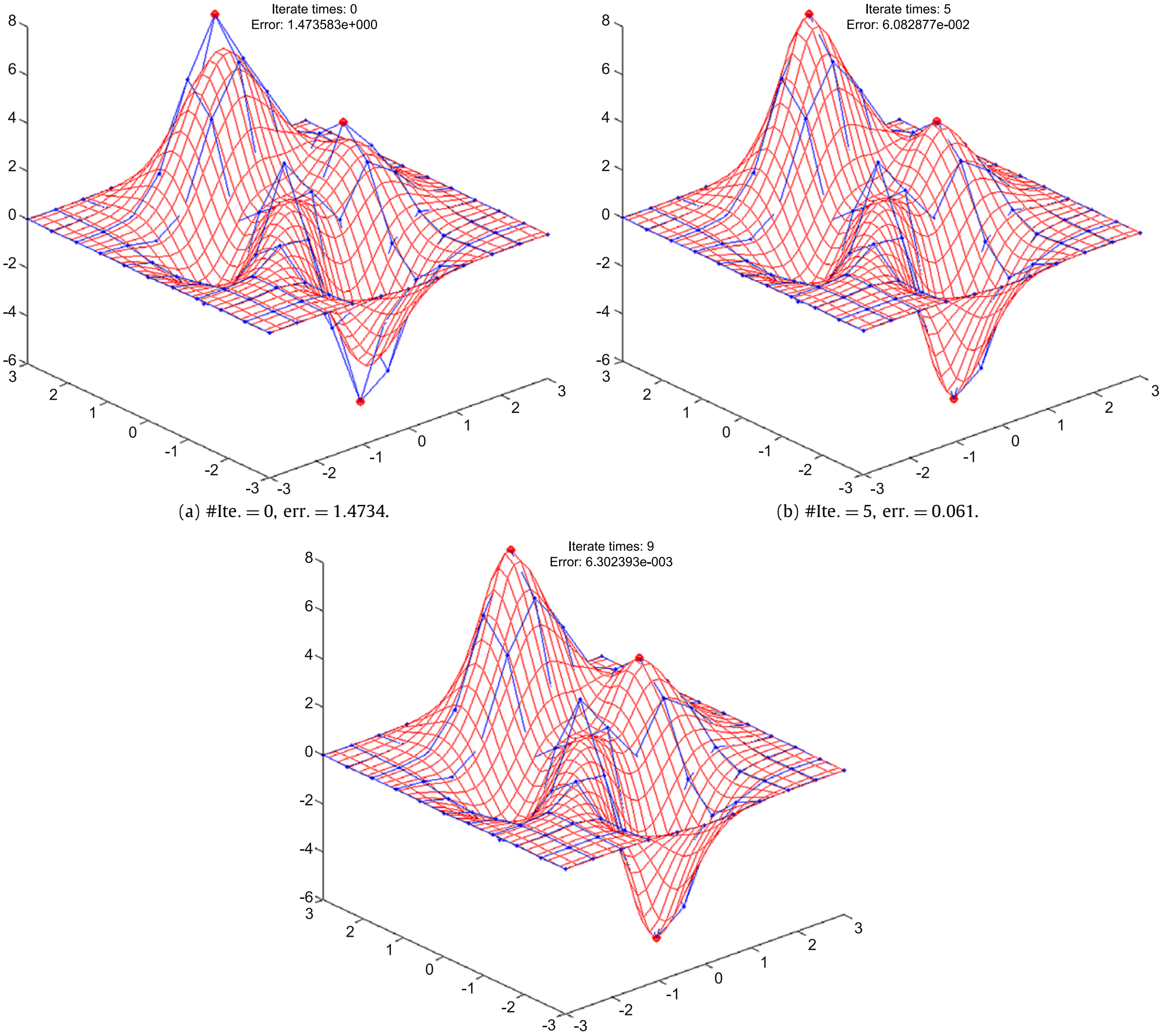
$$\begin{cases} x(t) = t\cos(4t) + 3, \\ y(t) = t\sin(4t) + 3, \\ z(t) = 1.2t, \end{cases}$$

at $t_i = \frac{3\pi}{10} + i\frac{\pi}{10}$, $i = 0, 1, \ldots, 19$.

Moreover, based on Theorems 3, 4, and Corollary 2, the local progressive-iterative approximation formats for rational Bézier curve and NURB (Non-Uniform Rational B-spline) curve are both convergent. Similarly, adjusting one control point of a rational Bézier curve influences the whole curve, while adjusting one control point of an NURB curve only affects several segments of the curve. Fig. 3 illustrates the local progressive-iterative approximation of a cubic NURB curve to three separated data points, namely, the seventh, eleventh, and thirteenth. The data points in homogeneous form in this example are (x_i, y_i, z_i, w_i), $i = 0, 1, \ldots, 19$, where (x_i, y_i, z_i) are sampled from

$$\begin{cases} x(t) = 3\cos t, \\ y(t) = 3\sin t\cos t, \\ z(t) = t, \end{cases}$$

at $t_i = i\frac{6\pi}{19}$, $i = 0, 1, \ldots, 19$, and $w_i = 1$, $i = 0, 1, \ldots, 19$, except $w_7 = w_{11} = w_{13} = 0.7$.



(a) #Ite. = 0, err. = 1.4734.

(b) #Ite. = 5, err. = 0.061.

(c) #Ite. = 9, err. = 0.006.

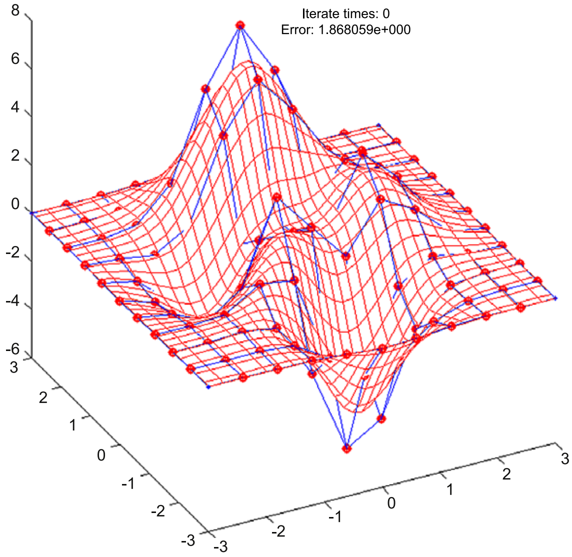
Fig. 4. Approximate three pre-specified points (red dots) with bi-cubic B-spline patch by local progressive-iterative approximation format. (For interpretation of colors in this figure, the reader is referred to the web version of this article.)

On the other hand, Fig. 4 to Fig. 7 are the examples for patches. In Fig. 4, the data points are sampled from the *peak* function of Matlab, and only three of them (displayed in red dots) are selected to be approximated by the local PIA of bi-cubic B-spline patch. As illustrated in Fig. 4, the initial fitting error is 1.4736. After nine iterations, the fitting error is reduced to 0.006.

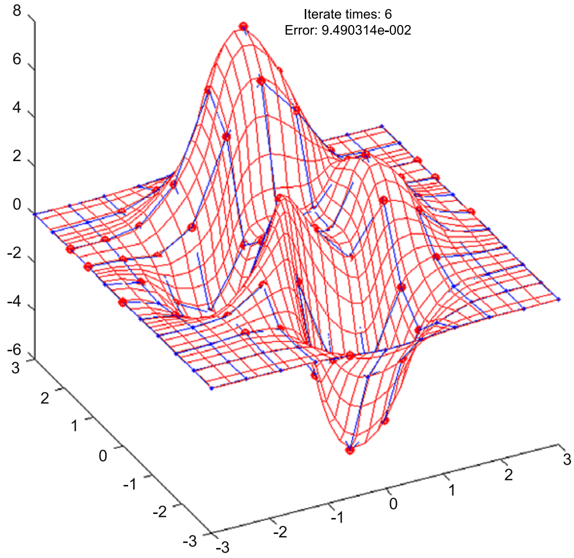
Furthermore, based on the local PIA format, we can design an adaptive fitting method to fit data points, by just adjusting the control points with fitting precision above a pre-defined threshold. Figs. 5–7 demonstrate the adaptive fitting method. In the three examples, the red dots represent the data points with fitting precision above a pre-defined threshold.

In Fig. 5, the data points sampled from the *peak* function of Matlab is fit using the adaptive manner. At the initial state, the fitting error is 1.8680, and the fitting precisions at nearly all data points are above the pre-defined threshold 10^{-2}, except at the four corners. By adjusting just the control points corresponding to these data points, whose fitting precision is above 10^{-2}, using local PIA of bi-cubic B-spline patch, the fitting precisions at all data points are below the pre-defined threshold 10^{-2} after eighteen iterations.

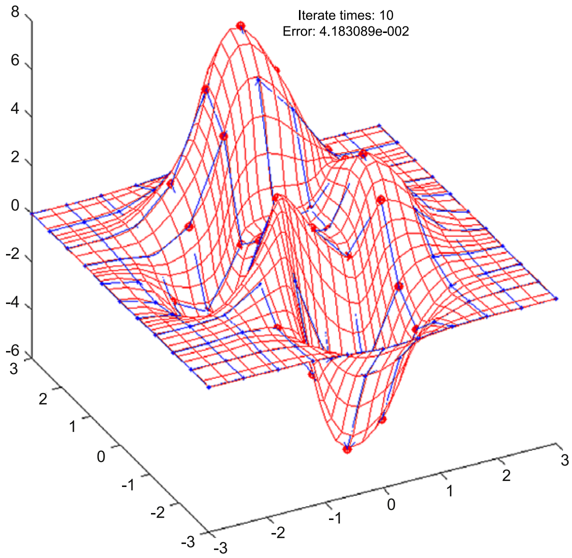
Fig. 6 is another example for adaptive fitting. The data points are sampled from a triangular mesh model *ear-cut*. The pre-defined fitting precision is 10^{-3}. With the increase of the iteration time, the number of the data points with fitting precision above 10^{-3}, displayed in red dots, is decreased continually (Fig. 6(a)–(c)). After seven iterations, the fitting precisions at all data points are below the pre-defined threshold 10^{-3} (Fig. 6(d)).



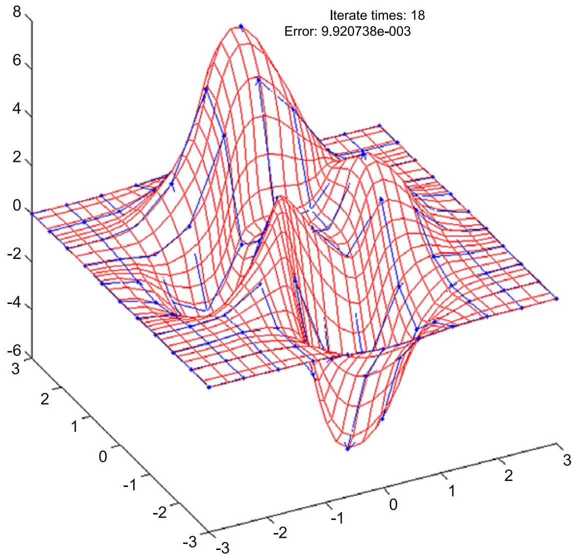
(a) #Ite. = 0, err. = 1.8680.



(b) #Ite. = 6, err. = 0.095.



(c) #Ite. = 10, err. = 0.042.



(d) #Ite. = 18, err. = 0.0099.

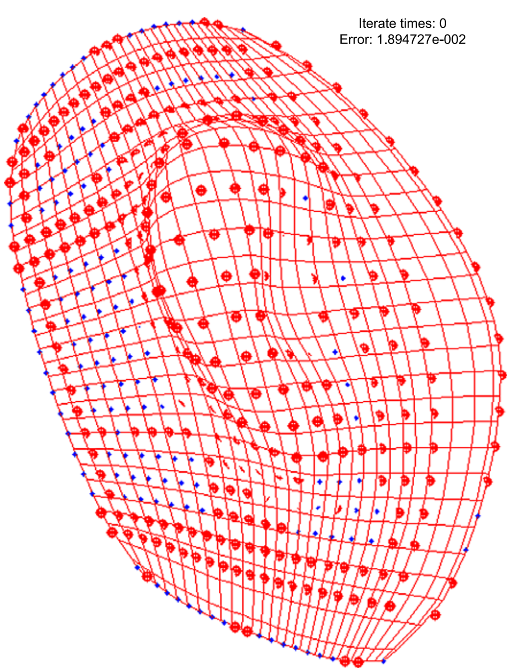
Fig. 5. Approximate the data points sampled from the *peak* function in Matlab by the adaptive fitting algorithm using local PIA for bi-cubic B-spline patch till the fitting precision for each point is below $e = 10^{-2}$, where the points with fitting precision above $e = 10^{-2}$ are plotted in red, and the points with precision below $e = 10^{-2}$ in blue. (For interpretation of colors in this figure, the reader is referred to the web version of this article.)

Table 1
Comparison on the run time between the local and global PIA.

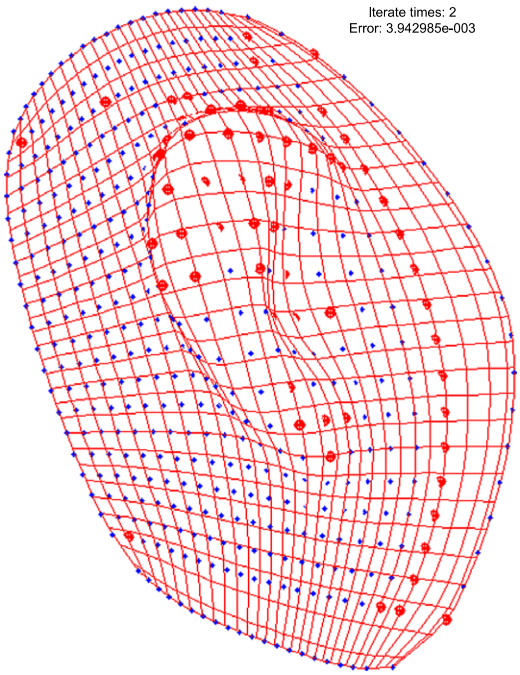
Model names	#Iterations	Time for local PIA	Time for global PIA
Peak (Fig. 5)	18	0.0237	0.3922
Ear-cut (Fig. 6)	7	0.0332	0.4217
Face (Fig. 7)	11	0.8832	6.0143

In Fig. 7, the data point set of 6561 points is sampled from the triangular model *face*. They are adaptively fit using the local PIA format of bi-cubic B-spline patch. The fitting precision threshold is set as 10^{-4}. After twelve iterations, the fitting precisions at all data points are below the threshold.

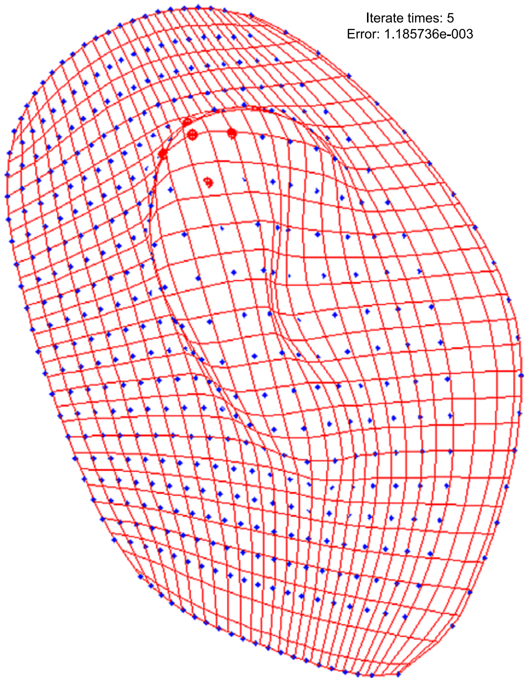
For comparison, we list in Table 1 the run time of the examples in Fig. 5 to Fig. 7, using the local PIA and global PIA, respectively. The second column of Table 1 is the iteration times. The third and fourth columns are the time cost by the local and global PIA, respectively, where the time is in seconds. It can be seen from Table 1 that, the local PIA is faster than the global PIA algorithm.



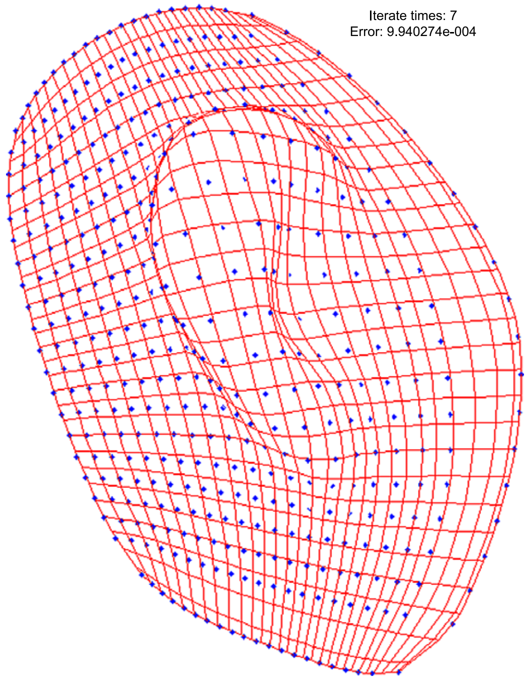
(a) #Ite. = 0, err. = 1.8947×10^{-2}.



(b) #Ite. = 2, err. = 3.9430×10^{-3}.



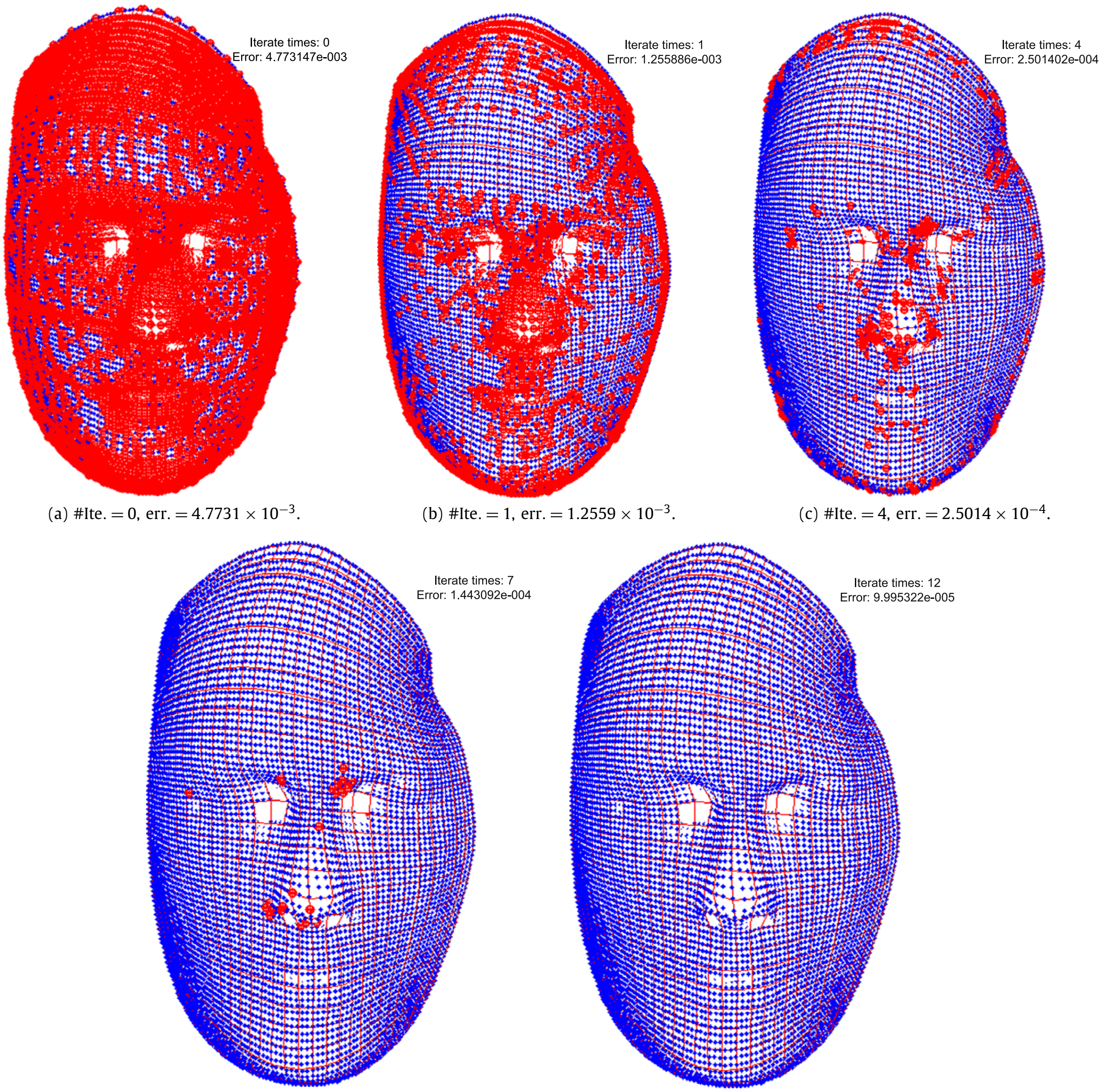
(c) #Ite. = 5, err. = 1.1857×10^{-3}.



(d) #Ite. = 7, err. = 9.9403×10^{-4}.

Fig. 6. Approximate $21 \times 21 = 441$ data points sampled from the *ear-cut* model by the adaptive fitting algorithm using local PIA for bi-cubic B-spline patch till the fitting precision for each point is below $e = 10^{-3}$, where the points with fitting precision above $e = 10^{-3}$ are plotted in red, and the points with precision below $e = 10^{-3}$ in blue. (For interpretation of colors in this figure, the reader is referred to the web version of this article.)

Noticeably, the convergence rate of the PIA format is determined by the smallest eigenvalue of the corresponding collocation matrix. In general, the convergence rate can be improved by multiplying a suitable coefficient before the adjusted vectors, to make the smallest eigenvalue larger.



(a) #Ite. $= 0$, err. $= 4.7731 \times 10^{-3}$.

(b) #Ite. $= 1$, err. $= 1.2559 \times 10^{-3}$.

(c) #Ite. $= 4$, err. $= 2.5014 \times 10^{-4}$.

(d) #Ite. $= 7$, err. $= 1.4431 \times 10^{-4}$.

(e) #Ite. $= 12$, err. $= 9.9953 \times 10^{-5}$.

Fig. 7. Approximate $81 \times 81 = 6561$ data points sampled from the *face* model by the adaptive fitting algorithm using local PIA for bi-cubic B-spline patch till the fitting precision at each point is below $e = 10^{-4}$, where the points with fitting precision above $e = 10^{-4}$ are plotted in red, and the points with precision below $e = 10^{-4}$ in blue. (For interpretation of colors in this figure, the reader is referred to the web version of this article.)

7. Conclusion

In this paper, we develop the local PIA format for blending curves and patches. Specifically, we show that the local PIA format for the blending curve with normalized totally positive basis, as well as the most commonly used bi-cubic B-spline patch, is convergent. Moreover, a special adjusting manner is developed to make the local PIA format for a generic blending patch with normalized totally positive basis convergent. The local PIA format adjusts only a part of the control points of a blending curve or patch, and the limit interpolates the part of the corresponding data points. Based on the local PIA format, an adaptive fitting algorithm can be designed, by just adjusting the control points with the fitting precision above a pre-defined threshold.

It should be pointed out that, though we have made lots of experiments on local PIA for a blending patch with normalized totally positive basis, and all of them are convergent, it is still not clear whether the local PIA for a generic blending patch with normalized totally positive basis is convergent in theory. It is our future work.

Acknowledgements

This paper is supported by Natural Science Foundation of China (Nos. 60970150, 60933008, 60736019), and Zhejiang Provincial Natural Science Foundation of China (Nos. Y1090416, R106449).

Appendix A. Proof to Theorem 4

Proof. (See Shi and Wang (2006).) $\forall\varepsilon > 0$, without loss of generality, let $\varepsilon < \frac{\omega_i}{2}$. Since $\boldsymbol{R}_i^k \to \boldsymbol{Q}_i$, $k \to \infty$, $\exists N > 0$, when $k > N$, we have $\|\boldsymbol{R}_i^k - \boldsymbol{Q}_i\| < \varepsilon$. Thus,

$$\left\|\boldsymbol{R}_i^k - \boldsymbol{Q}_i\right\|^2 = \left\|\Omega_i^k \boldsymbol{T}_i^k - \omega_i \boldsymbol{P}_i\right\|^2 + \left\|\Omega_i^k - \omega_i\right\|^2 < \varepsilon^2,$$

and then,

$$\left\|\Omega_i^k - \omega_i\right\| < \varepsilon, \quad \text{and} \quad \left\|\Omega_i^k \boldsymbol{T}_i^k - \omega_i \boldsymbol{P}_i\right\| < \varepsilon,$$

therefore,

$$\lim_{k\to\infty} \Omega_i^k = \omega_i.$$

On the other hand, let $\Omega_i^k = \omega_i + \eta$, with $|\eta| < \varepsilon$. Then,

$$\left\|\Omega_i^k \boldsymbol{T}_i^k - \omega_i \boldsymbol{P}_i\right\| = \left\|(\omega_i + \eta)\boldsymbol{T}_i^k - (\omega_i + \eta)\boldsymbol{P}_i + \eta\boldsymbol{P}_i\right\| = \left\|(\omega_i + \eta)\left(\boldsymbol{T}_i^k - \boldsymbol{P}_i\right) + \eta\boldsymbol{P}_i\right\| < \varepsilon,$$

so,

$$\left\|(\omega_i + \eta)\left(\boldsymbol{T}_i^k - \boldsymbol{P}_i\right)\right\| - \|\eta\boldsymbol{P}_i\| < \varepsilon,$$

namely,

$$\left\|\boldsymbol{T}_i^k - \boldsymbol{P}_i\right\| < \frac{(1 + \|\boldsymbol{P}_i\|)\varepsilon}{|\omega_i + \eta|} \leqslant \frac{(1 + \|\boldsymbol{P}_i\|)\varepsilon}{\omega_i - |\eta|}.$$

Since $|\eta| < \varepsilon < \frac{\omega_i}{2}$, we have

$$\left\|\boldsymbol{T}_i^k - \boldsymbol{P}_i\right\| < \frac{2(1 + \|\boldsymbol{P}_i\|)\varepsilon}{\omega_i}.$$

Hence,

$$\lim_{k\to\infty} \boldsymbol{T}_i^k = \boldsymbol{P}_i. \qquad \square$$

References

Chen, J.L., Chen, X.H., 2000. Special Matrices. Tsinghua University Press, Beijing (in Chinese).
Chen, Z., Luo, X., Tan, L., Ye, B., Chen, J., 2008. Progressive interpolation based on Catmull–Clark subdivision surfaces. In: Pacific Graphic 2008. Comput. Graphics Forum 27 (7), 1823–1827.
Cheng, F., Fan, F., Lai, S., Huang, C., Wang, J., Yong, J., 2008. Progressive interpolation using loop subdivision surfaces. In: Chen, F., Jüttler, B. (Eds.), Proceedings of Geometric Modeling and Processing (GMP), 2008. In: Lecture Notes in Comput. Sci., vol. 4975, pp. 526–533.
Cheng, F., Fan, F., Lai, S., Huang, C., Wang, J., Yong, J., 2009. Loop subdivision surface based progressive interpolation. J. Comput. Sci. Tech. 24 (1), 39–46.
de Boor C., 1979. 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, pp. 299–302.
Delgado, J., Peña, J.M., 2007. Progressive iterative approximation and bases with the fastest convergence rates. Computer-Aided Geom. Design 24 (1), 10–18.
Fan, F., Cheng, F., Lai, S., 2008. Subdivision based interpolation with shape control. Computer-Aided Design Appl. 5 (1–4), 539–547.
Gofuku, S., Tamura, S., Maekawa, T., 2009. Point-tangent/point-normal B-spline curve interpolation by geometric algorithms. Computer-Aided Design 41, 412–422.
Lin, H., Bao, H., Wang, G., 2005. Totally positive bases and progressive iteration approximation. Comput. Math. Appl. 50 (3–4), 575–586. 2005.4.
Lin, H., Wang, G., Dong, C., 2004. Constructing iterative non-uniform B-spline curve and surface to fit data points. Sci. China Ser. F 47 (3), 315–331.
Maekawa, T., Matsumoto, Y., Namiki, K., 2007. Interpolation by geometric algorithm. Computer-Aided Design 39, 313–323.
Martin, T., Cohen, E., Kirby, R.M., 2009. Volumetric parameterization and trivariate B-spline fitting using harmonic functions. Computer-Aided Geom. Design 26 (6), 648–664.
Qi, D., Tian, Z., Zhang, Y., Feng, J., 1975. The method of numeric polish in curve fitting. Acta Math. Sinica 18 (3), 173–184.
Shi, L., Wang, R., 2006. An iterative algorithm of NURBS interpolation and approximation. J. Math. Res. Exposition 26 (4), 735–743.