Explicit Matrix Expressions of Progressive Iterative Approximation

Jie Chen¹,² and Guo-Jin Wang*¹,²
¹Department of Mathematics, Zhejiang University, Hangzhou 310027, China
²State Key Laboratory of CAD&CG, Zhejiang University, Hangzhou 310027, China

Abstract – Just by adjusting the control points iteratively, progressive iterative approximation (PIA) presents an intuitive and straightforward scheme such that the resulting limit curve (surface) can interpolate the original data points. In order to obtain more flexibility, adjusting only a subset of the control points, a new method called local progressive iterative approximation (LPIA) has also been proposed. But to this day, there are two problems about PIA and LPIA: (1) Only an approximation process is discussed, but the accurate convergence curves (surfaces) are not given. (2) In order to obtain an interpolating curve (surface) with high accuracy, recursion computations are needed time after time, which result in a large workload. To overcome these limitations, this paper gives an explicit matrix expression of the control points of the limit curve (surface) by the PIA or LPIA method, and proves that the column vector consisting of the control points of the PIA's limit curve (or surface) can be obtained by multiplying the column vector consisting of the original data points on the left by the inverse matrix of the collocation matrix (or the Kronecker product of the collocation matrices in two direction) of the blending basis at the parametric values chosen by the original data points. Analogously, the control points of the LPIA's limit curve (or surface) can also be calculated by one-step. Furthermore, the G¹ joining conditions between two adjacent limit curves obtained from two neighboring data points sets are derived. Finally, a simple LPIA method is given to make the given tangential conditions at the endpoints can be satisfied by the limit curve.

Keywords: Explicit expression, Progressive iterative approximation, Local progressive iterative approximation, Limit control polygon, G¹ joining

1. Introduction

In Computer Aided Geometric Design (CAGD), to fit or approximate the data points by using polynomial parametric curve (surface) is an important and basic project. Given a sequence of data points, in order to get a proper parametric curve (surface) which interpolates the original data points, a classical and effective method is called least-square method. In this method, the approximation curve (surface) can be derived by minimizing the least-square error. Recently, a new method called progressive iterative approximation (PIA) which can address the above project has been researched by many scholars (Lin et al., 2004, 2005; Delgado et al., 2007; Cheng et al., 2009; Lin, 2010; Lu, 2010; Chen et al., 2011). By adjusting the control points of a blending curve or patch iteratively, a sequence of curves or patches is generated progressively. With the increase of the iterative time, the distance between the original data points and the blending parametric curves or patches becomes more and more close, finally the limit curve (surface) can interpolate the original data points.

In (Ando, 1987), a type of totally positive matrices and their properties were discussed, this provided partial theoretic base of the research of the PIA property. In (Qi et al., 1975) and (de Boor, 1979), the PIA property for the uniform cubic B-spline basis was opened out, respectively. Later, in (Lin et al., 2004), this result was extended to the non-uniform cubic B-spline basis and the non-uniform bicubic tensor product B-spline surfaces; in (Lin et al., 2005), it was proved that both the curve and tensor product surface generated by the normalized totally positive (NTP) basis (Karlin, 1968) satisfied the PIA property. In (Delgado and Peña, 2007), the convergence rates of the different NTP basis were compared, and it was shown that the normalized B-basis satisfied the PIA property with the fastest convergence rates. In order to improve the iteration convergence rates, Lu (2010) introduced a new method of weighted PIA and gave a way to choose the optimal value of the weight. Cheng et al. (2009) constructed the loop subdivision surface based progressive interpolation. Furthermore, different from the previous method, Lin (2010) proposed an idea of local PIA which approximated only a chosen subset of the initial control points, not all of them, so it was more flexible. Chen and Wang (2011) extended the PIA property of the univariate NTP bases to the bivariate Bernstein basis over a triangle domain for constructing triangular Bézier surfaces. However, the limit location of the PIA model cannot be calculated accurately, and if the higher
accuracy is requested by users, the time-consuming iterative calculation must be needed. This severely affects the efficiency in reverse engineering applications by the PIA method. As for the method of fitting data points based on the subdivision technique, Suzuki et al. (1999) have pointed out that, “The basic idea is to use the subdivision limit position (SLP) to adapt the control mesh of the subdivision surface to the data points.... It does fail to capture local characteristics of data points. Consequently, the proposed method is not suitable for generating a surface that precisely interpolates the data points.” On the other hand, Generating subdivision surface always needs some subdivision steps for a net which is a considerable time consuming process. To overcome these shortcomings, this paper gives an explicit expression of the limit curve of the limit surface (by the PIA or LPIA method). Its strongpoint consists in that, firstly the interpolation for data points by our limit curve (surface) is precise absolutely, and secondly the evaluation process is direct and rapid because it needs only a matrix multiplication by using the explicit expression. In a word, both the computational accuracy and efficiency could be reached optimal simultaneously. Moreover, the \( G^i \) continuous condition between two limit curves generated by two adjacent data sets respectively using the PIA method is derived. An application of this method is that the interpolation problem of the scattered data points can be derived. An application of this method is that the strongpoint consists in that the initial curve \( \tilde{C}(i) \) has the PIA property. In the iterative process, for the adjusting vectors set

\[
\Delta_i = [\Delta_0, \Delta_1, \cdots, \Delta_n] \quad (k = 0, 1, \cdots)
\]

we obtain the iterative format \( \Delta^{k+1} = (I - B)\Delta^k \), where \( I \) is the identity matrix, and

\[
B(B_0, \cdots, B_n ; t_0, \cdots, t_n) = (B_j(t_i))_{i=0}^n ; \quad j = 0, \cdots, n
\]

is the collocation matrix of the blending basis \( B_0(t), B_1(t), \cdots, B_n(t) \) at the parametric values \( \{t_i\}_{i=0}^n \). It is obvious that if the spectral radius of the matrix \( I - B \) is less than 1, i.e., \( \rho(I - B) < 1 \), then

\[
\lim_{k \to \infty} C^k(i) = P^0_i.
\]

2. Explicit Expressions of the Control Points of the Limit Curve (Surface)

2.1 In the case of PIA method

Given a sequence of points \( \{P_i\}_{i=0}^n \), each point \( P_i \) is assigned to a parameter value \( t_i \), \( i = 0, 1, \cdots, n \). Let \((B_0(t), B_1(t), \cdots, B_n(t))\) be a blending basis, that is, each basis function is nonnegative and satisfies the expression \( \sum_{i=0}^n B_i(t) = 1 \). Then we obtain an initial curve

\[
C^0(t) = \sum_{i=0}^n P_i B_i(t)
\]  

For every control point \( P_i \) letting an adjusting vector \( \Delta_i = P_i - C^0(t_i) \), \( i = 0, 1, \cdots, n \)

and new data point \( P_i = P_i^0 + \Delta_i \), \( i = 0, 1, \cdots, n \)

we can get the second curve

\[
C^1(t) = \sum_{i=0}^n P_i^1 B_i(t)
\]  

Similarly, we can get the curve \( C^{k+1}(i) \) after the \((k+1)\)-th iteration

\[
C^{k+1}(t) = \sum_{i=0}^n P_i^{k+1} B_i(t)
\]

where

\[
\Delta_i^0 = P_i - C^0(t_i) \quad i = 0, 1, \cdots, n
\]

\[
P_i^{k+1} = P_i^k + \Delta_i \quad i = 0, 1, \cdots, n
\]

If \( \lim_{k \to \infty} C^k(t_i) = P^0_i \), we call that the initial curve \( \tilde{C}(i) \) has the PIA property. In the iterative process, for the adjusting vectors set

\[
\Delta^k = [\Delta_0, \Delta_1, \cdots, \Delta_n] \quad (k = 0, 1, \cdots)
\]

we obtain the iterative format \( \Delta^{k+1} = (I - B)\Delta^k \), where \( I \) is the identity matrix, and

\[
B^T(B_0, \cdots, B_n ; t_0, \cdots, t_n) = (B_j(t_i))_{i=0}^n ; \quad j = 0, \cdots, n
\]

is the collocation matrix of the blending basis \( B_0(t), B_1(t), \cdots, B_n(t) \) at the parametric values \( \{t_i\}_{i=0}^n \). It is obvious that if the spectral radius of the matrix \( I - B \) is less than 1, i.e., \( \rho(I - B) < 1 \), then

\[
\lim_{k \to \infty} C^k(i) = P^0_i.
\]

The control points after \( k + 1 \) iterations can be expressed as

\[
P_i^{k+1} = P_i^k + \Delta_i^k = P_i^k + P_i - C^k(t) = P_i^k + \sum_{j=0}^n P_j^k B_j(t)
\]

\[i = 0, 1, \cdots, n; \quad k = 0, 1, \cdots\]

Letting

\[
P^{k+1} = [P_0^{k+1}, P_1^{k+1}, \cdots, P_n^{k+1}]^T,
\]

\[
P^k = [P_0^k, P_1^k, \cdots, P_n^k]^T, \quad P = [P_0, P_1, \cdots, P_n]^T
\]

we rewrite the above formulas in matrix form:

\[
P^{k+1} - P = (I - B)P^k
\]

Since \( B \) is a nonsingular matrix,

\[
P^{k+1} - B^{-1}P = (I - B)(P^{k+1} - B^{-1}P) = (I - B)^2(P^{k-1} - B^{-1}P)
\]

\[= \cdots = (I - B)^k(P^{k-1} - B^{-1}P)
\]

As \( \rho(I - B) < 1 \), then

\[
\lim_{k \to \infty} P^{k+1} = B^{-1}P
\]
This means that the control points of the limit curve which interpolates the original data points \( \{P_{ij}\}_{i=0}^{n} \) are \( B^{-1}P \). Thus, the column vector consisting of the control points of the PIA’s limit curve can be obtained by multiplying the column vector \( P \) consisting of \( \{P_{ij}\}_{i=0}^{n} \) on the left by the inverse matrix of the collocation matrix \( B=(B(t_j))^m_{i=0} \), generating from the blending basis \( B_i \) with the parametric values \( \{t_j\}_{j=0}^{n} \), which are chosen by \( \{P_{ij}\}_{i=0}^{n} \). It is evident that the computational process is simple and direct and the result is accurate.

Similarly, we can discuss the case of surfaces. Given a sequence of points \( \{P_{ij}\}_{i=0}^{m} \), each point \( P_{ij} \) is assigned to a pair of parameter values \( (u_i, v_j) \), \( i=0, 1, \cdots, m; j=0, 1, \cdots, n \), where

\[ u_0<v_1<\cdots<u_m, v_0<v_1<\cdots<v_n. \]

Similar with the case of curves, we can obtain an initial surface

\[ S_0(u,v)=\sum_{i=0}^{m} \sum_{j=0}^{n} P_{ij} B_i(u) B_j(v) \]  

and a sequence of surfaces

\[ S_k(u,v)=\sum_{i=0}^{m} \sum_{j=0}^{n} P^{k+1}_{ij} B_i(u) B_j(v) \]  

where

\[ \{P^{0}_{ij}\}=\{P_{ij}\}_{i=0}^{m} \, \{P^{k+1}_{ij}\}=\{P_{ij}\}_{i=0}^{m} + \Delta_{ij} B_i(u_i) B_j(v_j) \]  

\[ i=0, 1, \cdots, m; j=0, 1, \cdots, n; k=0, 1, \cdots \]  

\[ \lim_{k \to \infty} S_k(u,v)=P^\infty_{ij}, \quad i=0, 1, \cdots, m; j=0, 1, \cdots, n \]  

where \( \Delta_{ij}=[\Delta_{ij,0,0}, \Delta_{ij,0,1}, \cdots, \Delta_{ij,m,0}, \cdots, \Delta_{ij,m,m}]^T \)  

\[ k=0, 1, \cdots \]  

the iterative format for \( \Delta_{ij} \) is \( \Delta_{ij}=(I-B)\Delta_{ij} \), where \( I \) is the identity matrix, and \( B \) is the Kronecker product of two collocation matrices \( B_1 \) and \( B_2 \), that is \( B=B_1 \otimes B_2 \), here

\[ B_1=(B_1(u_i))_{i=0}^{m} \, \cdots, B_2=(B_2(v_j))_{j=0}^{n} \]  

It is obvious that if the spectral radius of the matrix \( I-B \) is less than 1, i.e., \( \rho(I-B)<1 \), then \( \lim_{k \to \infty} S_k(u,v)=P^\infty_{ij} \).

The control points after \( k+1 \) iterations can be expressed as

\[ P^{k+1}_{ij}=P^{k}_{ij}+\Sigma^{k}_{l=0} \Sigma^{n}_{j=0} P^{k}_{lj} B_i(u_l) B_j(v_j) \]

\[ =P^{k}_{ij}+\Sigma^{m}_{i=0} \Sigma^{n}_{j=0} P^{k}_{ij} B_i(u_i) B_j(v_j) \]  

\[ i=0, 1, \cdots, m; j=0, 1, \cdots, n; k=0, 1, \cdots. \]

Letting

\[ P^{k+1}_{ij}=[P^{k+1}_{00}, \cdots, P^{k+1}_{0n}, P^{k+1}_{10}, \cdots, P^{k+1}_{1n}, \cdots, P^{k+1}_{mn}]^T \]  

\[ P^{k}=[P^{0}_{00}, \cdots, P^{0}_{0n}, P^{0}_{10}, \cdots, P^{0}_{1n}, \cdots, P^{0}_{mn}]^T \]  

\[ P=[P_{00}, P_{01}, \cdots, P_{0n}, P_{10}, \cdots, P_{1n}, \cdots, P_{mn}]^T \]  

We rewrite the above formulas in matrix form:

\[ P^{k+1}=(I-B)P^{k} \]  

Noting that \( B \) is a nonsingular matrix, so

\[ P^{k+1}=(I-B)(P^{k-1}-B^{-1})P \]  

\[ =(I-B)^2(P^{k-1}-B^{-1})P \]

As \( \rho(I-B)<1 \), then

\[ \lim_{k \to \infty} P^{k+1}=B^{-1}P \]

This indicates that the control points of the limit surface which interpolates the original data points \( \{P_{ij}\}_{i=0}^{m} \) are \( B^{-1}P \). Thus, the column vector consisting of the control points of the PIA’s limit surface can be computed straight by multiplying the column vector \( P \) consisting of \( \{P_{ij}\}_{i=0}^{m} \) on the left by the inverse matrix of the Kronecker product \( B=B_1 \otimes B_2 \), where \( B_1, B_2 \) are two collocation matrices of the blending basis \( B_i(u_i)_{i=0}^{m} \) and \( B_j(v_j)_{j=0}^{n} \) at the parametric values \( \{u_i\}_{i=0}^{m}, \{v_j\}_{j=0}^{n} \), respectively, corresponding to \( \{P_{ij}\}_{i=0}^{m} \).

### 2.2 In the case of LPIA method

In order to obtain more flexibility, Lin (2010) presented a new method called local progressive iterative approximation. This method only approximated a subset of the initial points, not all of them. The main idea and skill is only adjusted a subset of the control points in the iterative processes. And the shape of the parametric surfaces (curves) can be adjusted locally.

Given a set of the points \( \{P_{ij}\}_{i=0}^{m} \), we suppose only the \( i_{th}, i_{th}, \cdots, i_{th} \) points are adjusted, and the other points, the \( j_{th}, j_{th}, \cdots, j_{th} \) remain unchanged. Different from global PIA method, the new control points at each iterative step are given by

\[ P^{k+1}_{ij}=P^{k}_{ij}+\Delta_{ij} \]

\[ l \in \{i_{0}, i_{1}, \cdots, i_{l-1}\}, k=0, 1, \cdots, \]

\[ P^{k+1}_{ij}=P^{k}_{ij}+\Delta_{ij} = P^{k}_{ij}+\sum_{j=0}^{n} P^{k}_{lj} B_i(u_l) B_j(v_j) \]

\[ h \in \{i_{0}, i_{1}, \cdots, i_{l-1}\}, k=0, 1, \cdots, \]
Letting
\[ P^{k+1} = [P_{j_0}^{k+1}, P_{j_1}^{k+1}, \ldots, P_{j_{m-1}}^{k+1}, P_{j_m}^{k+1}]^T \]  
(24)
\[ P^k = [P_{j_0}^k, P_{j_1}^k, \ldots, P_{j_{m-1}}^k, P_{j_m}^k]^T \]  
(25)
\[ P = [P_{j_0}, P_{j_1}, \ldots, P_{j_{m-1}}, P_{j_m}]^T \]  
(26)

We rewrite the above formulas in matrix form as follows:
\[ P^{k+1} = P + (I_{n+1} - B)P^k \]  
(27)
where
\[ B = \begin{pmatrix} I_{j+1} & 0 \\ D_1 & B_2 \end{pmatrix} \]  
(28)
\[ B_2 = \begin{bmatrix} B_{i_0}(t_{i_0})B_{i_1}(t_{i_1}) \cdots B_{i_l}(t_{i_l}) \\ \vdots & \vdots & \vdots \\ B_{i_{m-1}}(t_{i_{m-1}})B_{i_m}(t_{i_m}) \end{bmatrix} \]  
(29)

here \( I_{j+1} \) is the \((J+1) \times (J+1)\) identity matrix, \( 0 \) is the \((J+1) \times (J+1)\) zero matrix. Note that \( B \) is a nonsingular matrix, so \( B \) is also a nonsingular matrix, then
\[ P^{k+1} - B^{-1}P = (I - B)(P^k - B^{-1}P) = (I - B)^{k+1} \]  
(30)
Since \( \rho(I_{j+1} - B_2) < 1 \),
\[ \lim_{k \to \infty} P^{k+1} - B^{-1}P = \begin{pmatrix} I_{j+1} & 0 \\ -B_2D_1 & B_2 \end{pmatrix} \]  
(31)

So we find that the control points of the LPIA's limit curve which interpolates the partial original data points \( \{P_{i,j}\}_{i,j=0}^{m,n} \) are \( B^k \).

Next we discuss the case of surfaces briefly. Given a set of the points \( \{P_{i,j}\}_{i=0}^{m} \), we suppose only the points with the subscripts \(\{i_0, j_0, (i_1, j_1), \ldots, (i_{m-1}, j_{m-1})\}\) are adjusted, and the other points with the subscripts \(\{k_0, l_0, (k_1, l_1), \ldots, (k_{j_m}, l_{j_m})\}\) remain unchanged. The new control points at each iterative step are given by
\[ P^{k+1}_{i,h} = P_{i,h} + \Delta_{i,h}^k \]  
(32)
\[ P^{k+1}_{i,h} = P_{i,h} + \Delta_{i,h}^k = P_{i,h} + \sum_{j=0}^{m-1} P_{i,j}B_{i}(u_j)B_{j}(v_h) \]  
(33)
\[ P = [P_{i,j}]_{i=0}^{m} \]  
(34)

We rewrite the above formulas in matrix form as
\[ P^{k+1} = P + (I_{n+1} - B)P^k \]  
(35)
where
\[ B = \begin{pmatrix} I_{j+1} & 0 \\ D_1 & B_2 \end{pmatrix} \]  
(36)
\[ D_1 = \begin{bmatrix} B_{i_0}(u_{i_0})B_{i_1}(u_{i_1})B_{i_2}(u_{i_2}) \cdots B_{i_{m-1}}(u_{i_{m-1}})B_{i_m}(v_h) \\ \vdots & \vdots & \vdots \\ B_{i_{m-1}}(u_{i_{m-1}})B_{i_m}(v_h) \end{bmatrix} \]  
(37)
\[ B_2 = \begin{bmatrix} B_{i_0}(u_{i_0})B_{i_1}(v_h)B_{i_2}(u_{i_2}) \cdots B_{i_{m-1}}(u_{i_{m-1}})B_{i_m}(v_h) \\ \vdots & \vdots & \vdots \\ B_{i_{m-1}}(u_{i_{m-1}})B_{i_m}(v_h) \end{bmatrix} \]  
(38)

here \( I_{j+1} \) is the \((J+1) \times (J+1)\) identity matrix, \( 0 \) is the \((J+1) \times (J+1)\) zero matrix. Noting that \( B \) is a nonsingular matrix, so \( B \) is also a nonsingular matrix, then
\[ P^{k+1} - B^{-1}P = (I - B)(P^k - B^{-1}P) = (I - B)^{k+1} \]  
(39)
Since \( \rho(I_{j+1} - B_2) < 1 \),
\[ \lim_{k \to \infty} P^{k+1} - B^{-1}P = \begin{pmatrix} I_{j+1} & 0 \\ -B_2D_1 & B_2 \end{pmatrix} \]  
(40)

It follows that the LPIA’s control points of the limit surface which interpolates the partial original data points \( \{P_{i,j}\}_{i,j=0}^{m,n} \) are \( B^k \).

2.3 Examples
Example 2.1 Consider the curve given by
\[ (x(t), y(t)) = (-\cos(t), \sin(t)), \quad t \in [0, 2\pi] \]
A sequence of 5 points \( \{P_j\}_{j=0}^4 \) is sampled from the parameter curve
\[
P_j = (x(s_j), y(s_j)), \quad s_j = \frac{i - 2}{4}, \quad i = 0, 1, \ldots , 4
\]

Letting \( \{P_j\}_{j=0}^4 \) be the control points, we can obtain an initial Bézier curve of degree 4 as follows:
\[
B_0^0(t) = \sum_{j=0}^{4} P_j B_j(t), \quad t \in [0, 1]
\]

Where \( B_j(t) = \binom{4}{j} (1-t)^{4-j} t^j \) are the Bernstein basis functions of degree 4. For convenience, we choose the uniform parameters \( \{t_i = i/4\}_{i=0}^4 \), the inverse matrix of the collocation matrix \( B \) of the blending basis \( \{B_j(t)\}_{j=0}^4 \) is
\[
B^{-1} = \begin{pmatrix}
1 & 0 & 0 & 0 & 0 \\
0.722 & -3.556 & 6.667 & -3.556 & 0.722 \\
-0.25 & 1.333 & -3 & 4 & -1.083 \\
0 & 0 & 0 & 0 & 1
\end{pmatrix}
\]

Then the column vector consisting of the control points of the limit curve can be instantly calculated by multiplying the column vector \( P \) consisting of \( \{P_j\}_{j=0}^4 \) on the left by Eq. (40):
\[
B^{-1} P = ((-1,0), (-1.052,0.771), (0,1.639), (1.052,0.771), (1,0))^T
\]

It is obvious that the result is absolutely accurate and the computational process is very fast because we need only a matrix multiplication and also Eq. (40) can be computed beforehand.

Fig. 2.1–2.3 illustrate the initial Bézier curve and the approximation curve after different iterations respectively by the PIA method. Fig. 2.4–2.6 illustrate the initial Bézier curve and the approximation curve.

**Example 2.2** A sequence of points \( \{P_{ij}\}_{i,j=0}^2 \) are given by \( \{(-1, 0.8, 1), (0.1, 1, 1.2), (0.8, 0.6, 0.9), (-1.2, 0, 1.3), (0, 0, 1.5), (1, -0.2, 1), (-0.9, -0.8, 0.9), (0.2, -0.7, 1.2), (0.9, -0.6, 0.9)\} \), the corresponding initial bi-quadric tensor product Bézier surface is
\[
S^0(u,v) = \sum_{i=0}^{2} \sum_{j=0}^{2} P_{ij} B_i(u) B_j(v)
\]

For convenience, we choose the uniform parameters \( \{u_i = i/2\}_{i=0}^2 \), \( \{v_j = j/2\}_{j=0}^2 \).

The Kronecker product \( B \) of two collocation matrices \( B_1 \) and \( B_2 \) is

![Fig. 2.2. Global approximation-after the 1st iteration.](image)

![Fig. 2.3. Global approximation-after the 5th iteration.](image)

![Fig. 2.4. Local approximation-initial Bézier curve.](image)
Then the column vector consisting of the control points of the limit surface can be directly obtained by multiplying the column vector $P$ consisting of $\{P_{i,j}\}^2_{i,j=0}$ on the left by Eq. (41):

$$B^{-1}P = \{(\begin{array}{c}-1, 0.8, 1 \\ 0.3, 1.3, 1.45 \\ 0.8, 0.6, 0.9 \\ -1.45, 0, 1.6, 5 \\ -0.15, -0.1, 2.225 \\ 1.15, -0.4, 1.1 \\ 0.9, -0.8, 0.9 \\ 0.4, -0.7, 1.5 \\ 0.9, -0.6, 0.9 \end{array})\}$$

It is evident that the result is absolutely accurate and the computational process is very rapid because we need only a matrix multiplication and also Eq. (41) can be computed beforehand.

Fig. 2.7–2.9 illustrate the initial Bézier surface and the approximation surfaces after different iterations respectively by the PIA method. Fig. 2.10–2.12 illustrate the initial Bézier surface and the approximation surfaces after different iterations respectively by the LPIA method (approximating $P_{1,0}$ and $P_{1,1}$ only). In each figure, the original control points and the control points of the limit surfaces are marked by solid points and hollow points respectively. Moreover, the limit surfaces and the approximation surfaces after different iterations are illustrated by grid surfaces and solid surfaces respectively.

**Example 2.3** In order to compare practical effects in reverse engineering between our method and subdivision technique, using the same set of data points locating on a face, shown as in Fig. 2.13, the algorithm of this paper and Loop subdivision algorithm (Loop, 1987) have been implemented, and the final results can be shown as in Fig. 2.15 and Fig. 2.16, respectively. Also Fig. 2.14 can be given if we implement conventional PIA after the first iteration. It is obvious to see that the subdivision algorithm and conventional PIA failed to capture local characteristics of data points, but our method can interpolate the all original data points accurately.

### 3. $G^4$ joining between two adjacent limit curve

Given two adjacent sequences of points $\{P_i\}^n_{i=0}$ and $\{Q_i\}^m_{i=0}$, where $P_0 = Q_0$. Also denoting the Bernstein basis functions of degree $n$ and $m$ by $(B^n_0(t), B^n_1(t), \ldots)$, $(B^m_0(t), B^m_1(t), \ldots)$,
interpolate these two original data points sets by the PIA method respectively. Suppose that the limit curve which interpolates $P_i$ is $C_1(t)$, satisfying $C_1(t_i) = P_i$, $i = 0, 1, \ldots, n$; the limit curve which interpolates $Q_j$ is $C_2(t)$, satisfying $C_2(t_j) = Q_j$, $j = 0, 1, \ldots, m$. We will discuss the conditions of $G^2$ joining between two adjacent limit curve $C_1(t)$ and $C_2(t)$.

Lemma 3.1 (Wang et al., 2001) Denote the Bernstein basis functions of degree $n$ by $(B^n_0(t), B^n_1(t), \ldots, B^n_n(t))$, $t \in [0, 1]$; given the control points $\{P_i\}_{i=0}^n$, we can obtain the initial Bézier curve as follows:
The necessary and sufficient conditions of $G^1$ joining $C_i(t)$ and $C_j(t)$ at the endpoint $P_0 = Q_0$ are

$$\begin{align*}
C_i(t_0) &= C_j(t_0) \\
C_i(t_0) &= l \cdot C_j(t_0)
\end{align*}$$

where $l$ is an arbitrary real number. As $C_i(t_0) = P_0 = Q_0 = C_j(t_0)$, the first condition of the above formula is satisfied naturally.

For convenience, we take $t_0 = \tau_0 = 0$. From Lemma 3.1, we know that the condition $C_i(t_0) = l \cdot C_j(t_0)$ is equivalent to $n(G_i - G_j) = l m(H_1 - H_0)$.

That is

$$n \sum_{k=0}^{\infty} ((B_n^{-1})_{1k} - (B_n^{-1})_{0k}) P_k = l m \sum_{k=0}^{\infty} ((B_m^{-1})_{1k} - (B_m^{-1})_{0k}) Q_k$$

(40)

where $(B_n^{-1})_{0k}$ is the element of the matrix $B_n^{-1}$ in the $k^{th}$ row and $j^{th}$ column.

Especially, if $n = m$, then the necessary and sufficient conditions of $G^1$ joining $C_i(t)$ and $C_j(t)$ at the endpoint $P_0 = Q_0$ are

$$\sum_{k=1}^{n} ((B_n^{-1})_{1k} - (B_n^{-1})_{0k}) (P_k - l Q_k) = 0$$

(41)

**Example 3.1** Consider the curve given by

$$x(t), y(t) = (1 - \cos(t), \sin(t)), \quad t \in [0, 2\pi]$$

A sequence of 5 points $\{P_i\}_{i=0}^4$ is sampled from the parameter curve

$$P_i = (x(s_i), y(s_i)) = i \cdot \frac{\pi}{4}, \quad i = 0, 1, \cdots, 4$$

The other curve is given by

$$x(t), y(t) = (\cos(t) - 1, -\sin(t)), \quad t \in [0, 2\pi]$$

A sequence of 5 points $\{Q_i\}_{i=0}^4$ is sampled from the parameter curve

$$Q_i = (x(s_i), y(s_i)) = i \cdot \frac{\pi}{4}, \quad i = 0, 1, \cdots, 4$$

Letting $\{P_i\}_{i=0}^4$ and $\{Q_i\}_{i=0}^4$ be the control points of two Bézier curves, respectively, we can obtain two initial Bézier curves of degree 4 respectively as follows:

$$C_i(t) = \frac{4}{i} \sum_{j=0}^{4} P_j B_i(t), \quad C_j(t) = \frac{4}{i} \sum_{j=0}^{4} Q_j B_i(t), \quad t \in [0, 1]$$

where $B_i(t) = \binom{4}{i} (1-t)^{4-i} t^i$ are the Bernstein basis functions of degree 4. For convenience, we choose the uniform parameters $\{t=\frac{i}{4}\}_{i=0}^4$.

It is obvious that the $G^1$ joining conditions of the two Bézier curves can be satisfied by the original control points $\{P_i\}_{i=0}^4$ and $\{Q_i\}_{i=0}^4$. Fig. 3.1 shows the joining effect of two limit curves. Fig. 3.2–3.3 are for the
demonstration of the joining effect by fixing \( \{P_i\}_{i=0}^4 \) and adjusting a part of \( \{Q_i\}_{i=0}^4 \). In each figure, the original control points \( \{P_i\}_{i=0}^4 \), \( \{Q_i\}_{i=0}^4 \) and the control points of the limit curves are marked by solid points and hollow points respectively. Moreover, the limit curves \( C_1(t) \) and \( C_2(t) \) are illustrated by solid lines and dashed lines respectively.

4. The LPIA method with tangential conditions at endpoints

In this section, we give a simple method, so that not only the original data points can be interpolated, but also the given tangential conditions can be satisfied by the limit curve using the LPIA method.

Given a sequence of points \( \{P_i\}_{i=0}^n \), each point \( P_i \) is assigned to a parameter value \( t_i \), \( i = 0, 1, \ldots, n \). Let \( (B_0(t), B_1(t), \ldots, B_n(t)) \) be a Bernstein basis of degree \( n \), and denote the limit Bézier curve interpolating the original points \( \{P_i\}_{i=0}^n \) by \( C(t) \), satisfying \( C_i(t) = P_i \), \( i = 0, 1, \ldots, n \). Suppose that \( t_0 = 0 \), that is \( C_i(t_0) = P_0 \).

From Lemma 3.1, it is obvious that the tangent direction at the endpoint of a Bézier curve is determined by the control points \( P_0 \) and \( P_1 \). In order to make the tangent direction at the endpoint \( P_0 \) of the limit curve \( C(t) \) is consistent with the vector \( \alpha \), that is \( C'(t_0) \)
each step of iteration process, the extend the original data points to . Then at derived. This method can be extended to the conditions between two adjacent limit curves are 

Jie Chen and Guo-Jin Wang Explicit Matrix Expressions of Progressive Iterative Approximation

Based on this simple explicit expression, the approximation errors can be ignored. Furthermore, the limit curves (surfaces) by the PIA and the points of the limit curves (surfaces) by the PIA method. In order to obtain the matrix, we get the explicit expressions of the control points of the limit curves (surfaces) rapidly, directly and accurately.

Moreover, the limit curves solid points and hollow-block point respectively. The control points are illustrated by solid lines and hollow points respectively.

**Example 4.1** Consider the curve given by 

\[(x(t), y(t)) = (\cos(t), \sin(t)), \quad t \in [0, 2\pi]\]

A sequence of 5 points \(\{P_i\}_{i=0}^{4}\) is sampled from the parameter curve 

\[P_i = (x(s_i), y(s_i)), \quad s_i = i \frac{\pi}{4}, \quad i = 0, 1, \cdots, 4\]

Fig. 4.1–4.3 illustrate the approximation effects by the LPIA method with different tangential conditions at the endpoint \(P_0\). In each figure, the original control points \(\{P_i\}_{i=0}^{4}\) and the adding point \(\hat{P}_1\) are marked by solid points and hollow-block point respectively. Moreover, the limit curves \(C(t)\) and the corresponding control points are illustrated by solid lines and hollow points respectively.

**5. Conclusion**

It is intuitive, stable and flexible to interpolate the data points by the PIA method. In order to obtain the limit curves (surfaces) rapidly, directly and accurately by recursive iterations, using the tool of collocation matrix, we get the explicit expressions of the control points of the limit curves (surfaces) by the PIA and the LPIA method. This result is effectual and simple, and the approximation errors can be ignored. Furthermore, based on this simple explicit expression, the \(G^1\) joining conditions between two adjacent limit curves are derived. This method can be extended to the \(G^1\) joining of multiple limit curves similarly. The convex-preserving conditions of the limit curves with constrained tangential conditions can be researched in future work.

**Acknowledgements**

This work was supported by the National Natural Science Foundations of China under Grant No. 6107-0065 and No. 60933007.

**References**


CHEN Jie Ph. D. candidate in the Department of Mathematics, Zhejiang University. His main research interest is computer aided geometric design. E-mail: jiechenalpha@yahoo.com.cn

WANG Guo-Jin Professor in the Department of Mathematics, Zhejiang University. His research interest covers computer aided geometric design and applied approximation theory. E-mail: wanggj@zju.edu.cn