Progressive Iterative Approximation of SOR for Non-uniform Cubic B-spline Curve and Surface Interpolation

Liangchen Hu[1], Huahao Shou[1(✉)], and Shiaofen Fang[2]

[1] College of Science, Zhejiang University of Technology, Hangzhou, Zhejiang, China
shh@zjut.edu.cn

[2] Department of Computer and Information Science, Indiana University-Purdue University Indianapolis, Indianapolis, IN, USA

Abstract. Progressive iterative approximation (PIA) is an efficient data fitting technique which makes the initial curve or surface approximate the data points to be processed by successive iterations. However, since the spectral radius of iterative matrix in traditional PIA is relatively large, the iterative convergence rate is relatively slow, which results in poor efficiency of data fitting. In this paper, we develop a successive over-relaxation progressive iterative approximation (SOR-PIA) for non-uniform cubic B-splines to overcome the defect. Besides, we employ the equidistant search strategy to estimate the relaxation factor, which greatly accelerates the convergence speed of the iterative process. Experimental results show that SOR-PIA iterative interpolation can achieve a higher accuracy within the equivalent number of iterations compared with the standard PIA and weighted PIA (WPIA) iterative interpolation.

Keywords: B-spline interpolation · Progressive iterative approximation · Successive over-relaxation · Iterative acceleration

1 Introduction

Data fitting is a crucial basic task of modeling and forecasting, which is widely employed in various fields. In fact, there have been many innovations and improvements in data fitting technology, among which PIA plays an important role due to its many satisfactory features. As a geometrically intuitive approach, PIA takes given data points as initial control points and dynamically updates them in a certain direction. PIA method has a broad application prospect, and it is of great significance to find a new fast iterative method for its applications.

In 2004, Lin proved the property of profit-and-loss for non-uniform cubic B-spline curve and surface [1], and for blending curves and tensor product blending patches with normalized totally positive basis in 2005 [2]. The approach proposed in [2] is called progressive iterative approximation (PIA), which addresses

Supported by the National Science Foundation of China (No. 61572430).

H. Song and D. Jiang (Eds.): SIMUtools 2020, LNICST 369, pp. 339–349, 2021.
https://doi.org/10.1007/978-3-030-72792-5_29

both interpolation and approximation (including EPIA [3] and LSPIA [4,10,11]). On the problem of iterative acceleration of PIA, Lu [5] derived the optimal iteration weights for the best convergence speed, and named it as WPIA. For large scale data fitting problems, Lin proposed the LSPIA algorithm using T-splines [6]. PIA has also been applied for hexahedral mesh generation in finite element analysis [7]. Since the iteration speed is independent of the number of control points in LSPIA, it provides a robust and efficient solution for the equation system of a tetrahedral mesh for an ideal NURBS body[1] [8]. In [9], the method of how to determine fewer control points in the least square fitting is also considered, which improves the fitting efficiency. PIA has also been used in satellite image processing, pattern recognition, hand drawn curve approximation, rational curve approximation, and tree trunk shape modeling. Recently, Lin et al. [12] provided an overview of the interpolatory and approximate geometric iteration methods, including PIA, geometric iteration and so on, and summarized that geometric iterative methods have wide applications in academic studies and engineering practices. The above improvements have a common property that the spectral radii of their iterative matrices are essentially at the same level, limiting their performance. Inspired by this, we proposed an efficient progressive iterative approximation algorithm based on HSS iteration in [13], whose spectral radius is much smaller than PIA. The contributions of our paper can be summarized as follows. Innovatively, we develop a SOR-PIA technique for non-uniform cubic B-spline curve and surface interpolation. Besides, we employ the equidistant search strategy to estimate the relaxation factor.

The rest of the paper is organized as follows. Section 2 reviews the standard iterative non-uniform B-spline curve and surface and presents a simple analysis of the relationship between the standard iterative format and directly solving equation system. The successive over-relaxation iteration method is introduced in Sect. 3 for solving the iterative interpolation problem, including the equidistant search strategy to make approximate computation for the best relaxation factor and to accelerate the convergence speed. The results of our numerical experiments are presented in Sect. 4. Section 5 concludes the paper with some additional summary.

2 Standard Iterative Interpolation and Its Analysis

2.1 Iterative Format of Curve

Given a set of data points list $\{\boldsymbol{Q}_i\}_{i=1}^{m}$, the accumulative chord length parameterization method can be used to calculate the parameter values $\{u_i\}_{i=1}^{m}$ on the initial iteration curve corresponding to the data points. Cubic non-uniform B-spline basis function is defined by the knot vector $\big\{0\ 0\ 0\ 0\ u_2\ u_3\ \cdots\ u_{m-1}\ 1\ 1\ 1\ 1\big\}$, whose interior knots are the parameter values.

Let $\boldsymbol{P}_i^0 = \boldsymbol{Q}_i, i = 1, \cdots, m$, $\boldsymbol{P}_0^0 = \boldsymbol{P}_1^0$, $\boldsymbol{P}_{m+1}^0 = \boldsymbol{P}_m^0$, the initial iterative interpolation curve $\boldsymbol{C}^0(u) = \sum_{i=0}^{m+1} \boldsymbol{P}_i^0 N_{i,3}(u)$ can be constructed. Assuming

[1] The NURBS body, or NURBS solid, refers to a NURBS-based CAD model.

that the maximum number of iterations is N, the curve iterative formula is

$$\boldsymbol{P}_i^k = \boldsymbol{P}_i^{k-1} + \boldsymbol{\Delta}_i^{k-1}, i = 1, \cdots, m; k = 1, \cdots, N, \tag{1}$$

in which, the difference vector is calculated as

$$\boldsymbol{\Delta}_i^{k-1} = \boldsymbol{Q}_i - \boldsymbol{C}^{k-1}(u_i), i = 1, \cdots, m; k = 1, \cdots, N,$$

and $\boldsymbol{P}_0^k = \boldsymbol{P}_1^k$, $\boldsymbol{P}_{m+1}^k = \boldsymbol{P}_m^k$. The Eq. (1) can be rewritten in matrix form as follows

$$\boldsymbol{P}^k = (\boldsymbol{I} - \boldsymbol{N})\boldsymbol{P}^{k-1} + \boldsymbol{Q},$$

where $\boldsymbol{I}$ is unit matrix,

$$\boldsymbol{N} = \begin{pmatrix} \overset{1}{N_{1,3}(u_2)} & N_{2,3}(u_2) & N_{3,3}(u_2) & \\ & \ddots & \ddots & \\ & N_{m-2,3}(u_{m-1}) & N_{m-1,3}(u_{m-1}) & \underset{1}{N_{m,3}(u_{m-1})} \end{pmatrix},$$

and $\boldsymbol{P}^k = (\boldsymbol{P}_1^k, \boldsymbol{P}_2^k, \cdots, \boldsymbol{P}_m^k)^T$, $\boldsymbol{Q} = (\boldsymbol{Q}_1, \boldsymbol{Q}_2, \cdots, \boldsymbol{Q}_m)^T$.

2.2 Iterative Format of Surface

Given a set of data matrix $\{\boldsymbol{Q}_{i,j}\}_{i=1,j=1}^{m,n}$, the cumulative chord length parameterization method can be used to calculate the parameter values $\{u_i, v_j\}_{i=1,j=1}^{m,n}$ on the initial iteration surface corresponding to the data points. Bi-cubic non-uniform B-spline basis functions are defined respectively by the knot vectors $\{0\ 0\ 0\ 0\ u_2\ u_3\ \cdots\ u_{m-1}\ 1\ 1\ 1\ 1\}$ and $\{0\ 0\ 0\ 0\ v_2\ v_3\ \cdots\ v_{n-1}\ 1\ 1\ 1\ 1\}$, whose interior knots are the parameter values.

Let $\boldsymbol{P}_{i,j}^0 = \boldsymbol{Q}_{i,j}$, $\boldsymbol{P}_{i,0}^0 = \boldsymbol{P}_{i,1}^0$, $\boldsymbol{P}_{i,n+1}^0 = \boldsymbol{P}_{i,n}^0$, $\boldsymbol{P}_{0,j}^0 = \boldsymbol{P}_{1,j}^0$, $\boldsymbol{P}_{m+1,j}^0 = \boldsymbol{P}_{m,j}^0$, $i = 1, \cdots, m, j = 1, \cdots, n$; $\boldsymbol{P}_{0,0}^0 = \boldsymbol{P}_{1,1}^0$, $\boldsymbol{P}_{0,n+1}^0 = \boldsymbol{P}_{1,n}^0$, $\boldsymbol{P}_{m+1,0}^0 = \boldsymbol{P}_{m,1}^0$, $\boldsymbol{P}_{m+1,n+1}^0 = \boldsymbol{P}_{m,n}^0$, the bi-cubic initial iterative interpolation tensor product surface $\boldsymbol{C}^0(u, v) = \sum_{i=0}^{m+1} \sum_{j=0}^{n+1} \boldsymbol{P}_{i,j}^0 N_{i,3}(u) N_{j,3}(v)$ can be constructed. Suppose the maximum number of iterations is N, then the surface iterative formula is

$$\boldsymbol{P}_{i,j}^k = \boldsymbol{P}_{i,j}^{k-1} + \boldsymbol{\Delta}_{i,j}^{k-1}, i = 1, \cdots, m; j = 1, \cdots, n; k = 1, \cdots, N, \tag{2}$$

where the difference vector is calculated as

$$\boldsymbol{\Delta}_{i,j}^{k-1} = (\boldsymbol{Q}_{i,j} - \boldsymbol{C}^{k-1}(u_i, v_j)), i = 1, \cdots, m; j = 1, \cdots, n; k = 1, \cdots, N,$$

and $\boldsymbol{P}_{i,0}^k = \boldsymbol{P}_{i,1}^k$, $\boldsymbol{P}_{i,n+1}^k = \boldsymbol{P}_{i,n}^k$, $\boldsymbol{P}_{0,j}^k = \boldsymbol{P}_{1,j}^k$, $\boldsymbol{P}_{m+1,j}^k = \boldsymbol{P}_{m,j}^k$, $i = 1, \cdots, m, j = 1, \cdots, n$; $\boldsymbol{P}_{0,0}^k = \boldsymbol{P}_{1,1}^k$, $\boldsymbol{P}_{0,n+1}^k = \boldsymbol{P}_{1,n}^k$, $\boldsymbol{P}_{m+1,0}^k = \boldsymbol{P}_{m,1}^k$, $\boldsymbol{P}_{m+1,n+1}^k = \boldsymbol{P}_{m,n}^k$.

Similar to the case of curve iteration, the Eq. (2) can be rewritten in matrix form as follows

$$\boldsymbol{P}^k = (\boldsymbol{I} - \boldsymbol{N})\boldsymbol{P}^{k-1} + \boldsymbol{Q},$$

where $\boldsymbol{I}$ is unit matrix, $\boldsymbol{N} = \boldsymbol{N}_1 \otimes \boldsymbol{N}_2$ is the Kronecker product between $\boldsymbol{N}_1$ and $\boldsymbol{N}_2$,

$$\boldsymbol{N}_1 = \begin{pmatrix} 1 & & & & \\ N_{1,3}(u_2) & N_{2,3}(u_2) & N_{3,3}(u_2) & & \\ & \ddots & \ddots & \ddots & \\ & & N_{m-2,3}(u_{m-1}) & N_{m-1,3}(u_{m-1}) & N_{m,3}(u_{m-1}) \\ & & & & 1 \end{pmatrix},$$

$$\boldsymbol{N}_2 = \begin{pmatrix} 1 & & & & \\ N_{1,3}(v_2) & N_{2,3}(v_2) & N_{3,3}(v_2) & & \\ & \ddots & \ddots & \ddots & \\ & & N_{n-2,3}(v_{n-1}) & N_{n-1,3}(v_{n-1}) & N_{n,3}(v_{n-1}) \\ & & & & 1 \end{pmatrix},$$

and

$$\boldsymbol{P}^k = (\boldsymbol{P}^k_{1,1}, \boldsymbol{P}^k_{1,2}, \cdots, \boldsymbol{P}^k_{1,n}, \boldsymbol{P}^k_{2,1}, \boldsymbol{P}^k_{2,2}, \cdots, \boldsymbol{P}^k_{2,n}, \cdots, \boldsymbol{P}^k_{m,1}, \boldsymbol{P}^k_{m,2}, \cdots, \boldsymbol{P}^k_{m,n}),$$
$$\boldsymbol{Q}^k = (\boldsymbol{Q}^k_{1,1}, \boldsymbol{Q}^k_{1,2}, \cdots, \boldsymbol{Q}^k_{1,n}, \boldsymbol{Q}^k_{2,1}, \boldsymbol{Q}^k_{2,2}, \cdots, \boldsymbol{Q}^k_{2,n}, \cdots, \boldsymbol{Q}^k_{m,1}, \boldsymbol{Q}^k_{m,2}, \cdots, \boldsymbol{Q}^k_{m,n}).$$

2.3 Analysis of Standard Iterative Interpolation

Remark 1. The standard iteration format of non-uniform B-spline curve and surface can be seen as an iterative matrix decomposition $\boldsymbol{N} = \boldsymbol{I} - (\boldsymbol{I} - \boldsymbol{N})$ from $\boldsymbol{N}$, so that

$$\boldsymbol{NP} = (\boldsymbol{I} - (\boldsymbol{I} - \boldsymbol{N}))\boldsymbol{P} = \boldsymbol{Q}$$
$$\Downarrow$$
$$\boldsymbol{P} = (\boldsymbol{I} - \boldsymbol{N})\boldsymbol{P} + \boldsymbol{Q}.$$

If an initial vector $\boldsymbol{P}^0$ is substituted into the right-hand side of above formula, we can calculate a new vector $\boldsymbol{P}^1 = (\boldsymbol{I} - \boldsymbol{N})\boldsymbol{P}^0 + \boldsymbol{Q}$. When $\boldsymbol{P}^1$ is substituted into the right-hand side of the formula, we will get $\boldsymbol{P}^2$, and so on. As this process continues, we have $\boldsymbol{P}^k = (\boldsymbol{I} - \boldsymbol{N})\boldsymbol{P}^{k-1} + \boldsymbol{Q}, k = 1, 2, \cdots$. The coefficient matrix for this iterative process is: $(\boldsymbol{I} - \boldsymbol{N})$.

Lemma 1. *Standard iterative format of the non-uniform B-spline curve and surface is convergent.*

Proof. Details are referred to Ref. [1].

Theorem 1. *The standard iterative interpolation for non-uniform B-spline curve and surface is an iterative method for solving linear equations system.*

Proof. For the iterative interpolation of curve or surface, we know from Remark 1 as follows

$$\boldsymbol{P}^k = (\boldsymbol{I} - \boldsymbol{N})\boldsymbol{P}^{k-1} + \boldsymbol{Q}, k = 1, 2, \cdots.$$

We take limit to the above formula on both sides, then

$$\lim_{k\to\infty} \boldsymbol{P}^k = (\boldsymbol{I} - \boldsymbol{N}) \lim_{k\to\infty} \boldsymbol{P}^{k-1} + \boldsymbol{Q}. \quad (3)$$

By Lemma 1, we know the sequence of data points is convergent, suppose $\lim_{k\to\infty} \boldsymbol{P}^k = \boldsymbol{P}$, then Eq. (3) can be written as $\boldsymbol{NP} = \boldsymbol{Q}$, which is the equation system of non-uniform B-spline curve or surface interpolation. Thus, the standard iterative interpolation for non-uniform B-spline curve and surface is actually an iterative method for solving linear equations system.

3 The Derivation of SOR-PIA

In this section, we introduce the successive over-relaxation iteration for non-uniform B-spline curve and surface interpolation. Compared with the standard iteration interpolation method PIA [1,2], successive over-relaxation iteration has a significant reduction of iterative steps for the iteration process.

3.1 The Case of Curve

Maintain the same conditions as the standard iteration such as the size of the data points, the parameterization of the data points, the value of the knot vector, the initial values of the control vertices. Using $\boldsymbol{P}_1^k, \boldsymbol{P}_2^k, \cdots, \boldsymbol{P}_{i-1}^k$ that has been calculated before $\boldsymbol{P}_i^k$ to participate in the calculation of $\boldsymbol{P}_i^k$ in the standard iteration process for non-uniform B-spline curve interpolation, then the iteration curve $\boldsymbol{C}^{k-1}(u) = \sum_{p=0}^{m+1} \boldsymbol{P}_p^{k-1} N_{p,3}(u)$ can be replaced by $\boldsymbol{C}^{k,k-1}(u) = \sum_{p=0}^{i-1} \boldsymbol{P}_p^k N_{p,3}(u) + \sum_{p=i}^{m+1} \boldsymbol{P}_p^{k-1} N_{p,3}(u)$, and iterative format of curve can be written as follows

$$\boldsymbol{P}_i^k = \boldsymbol{P}_i^{k-1} + \boldsymbol{\Delta}_i^{k,k-1}, i = 1, \cdots, m; k = 1, 2, \cdots, \tag{4}$$

where the difference vector is calculated as

$$\boldsymbol{\Delta}_i^{k,k-1} = \boldsymbol{Q}_i - \boldsymbol{C}^{k,k-1}(u_i), i = 1, \cdots, m; k = 1, 2, \cdots.$$

Then we make an improvement on the Eq. (4) by adding a weight $\frac{\omega}{N_{i,3}(u_i)}$ to the iterative difference vector as follows

$$\boldsymbol{P}_i^k = \boldsymbol{P}_i^{k-1} + \frac{\omega}{N_{i,3}(u_i)} \boldsymbol{\Delta}_i^{k,k-1}, i = 1, \cdots, m; k = 1, 2, \cdots. \tag{5}$$

The iterative process expressed in Eq. (5) is progressive iterative approximation format of successive over-relaxation for non-uniform B-spline curve interpolation called SOR-PIA of curve. Its difference from the traditional PIA is that it uses $\boldsymbol{P}_1^k, \boldsymbol{P}_2^k, \cdots, \boldsymbol{P}_{i-1}^k$ that has been calculated before $\boldsymbol{P}_i^k$ to participate in the calculation of $\boldsymbol{P}_i^k$ and adds a weight $\frac{\omega}{N_{i,3}(u_i)}$ which includes a relaxation factor ω in iterative difference vector.

Rewrite the Eq. (5) as follows

$$\begin{aligned}\boldsymbol{P}_i^k &= \boldsymbol{P}_i^{k-1} + \frac{\omega}{N_{i,3}(u_i)} \boldsymbol{\Delta}_i^{k,k-1} = \boldsymbol{P}_i^{k-1} + \frac{\omega(\boldsymbol{Q}_i - \boldsymbol{C}^{k,k-1}(u_i))}{N_{i,3}(u_i)} \\ &= (1-\omega)\boldsymbol{P}_i^{k-1} + \omega \frac{\boldsymbol{Q}_i - \boldsymbol{P}_{i-1}^k N_{i-1,3}(u_i) - \boldsymbol{P}_{i+1}^k N_{i+1,3}(u_i)}{N_{i,3}(u_i)},\end{aligned}$$

therefore the Eq. (5) can be written in matrix form

$$\boldsymbol{P}^k = (1-\omega)\boldsymbol{P}^{k-1} + \omega \boldsymbol{D}^{-1}\left(\boldsymbol{Q} - \boldsymbol{L}\boldsymbol{P}^k - \boldsymbol{U}\boldsymbol{P}^{k-1}\right), k = 1, 2, \cdots,$$

in which, $\boldsymbol{D} + \boldsymbol{L} + \boldsymbol{U} = \boldsymbol{N}$, where $\boldsymbol{D}$ is a diagonal matrix, $\boldsymbol{L}$ is a strictly lower triangular matrix, $\boldsymbol{U}$ is a strictly upper triangular matrix.

3.2 The Case of Surface

Similar to the case of curve. Using $\boldsymbol{P}_{1,1}^k, \cdots, \boldsymbol{P}_{1,j}^k, \boldsymbol{P}_{2,1}^k, \cdots, \boldsymbol{P}_{2,j}^k, \cdots, \boldsymbol{P}_{i-1,1}^k, \cdots, \boldsymbol{P}_{i-1,j}^k, \boldsymbol{P}_{i,1}^k, \cdots, \boldsymbol{P}_{i,j-1}^k$ that has been calculated before $\boldsymbol{P}_{i,j}^k$ to participate in the calculation of $\boldsymbol{P}_{i,j}^k$ in the standard iteration process for non-uniform B-spline surface interpolation, then the iteration surface $\boldsymbol{C}^{k-1}(u,v) = \sum_{p=0}^{m+1}\sum_{q=0}^{n+1}\boldsymbol{P}_{p,q}^{k-1} N_{p,3}(u)N_{q,3}(v)$ can be replaced by $\boldsymbol{C}^{k,k-1}(u,v) = \sum_{p=0}^{i-1}\sum_{q=0}^{n+1}\boldsymbol{P}_{p,q}^k N_{p,3}(u)N_{q,3}(v) + \sum_{q=0}^{j-1}\boldsymbol{P}_{i,q}^k N_{i,3}(u)N_{q,3}(v) + \sum_{q=j}^{n+1}\boldsymbol{P}_{i,q}^{k-1} N_{i,3}(u)N_{q,3}(v) + \sum_{p=i+1}^{m+1}\sum_{q=0}^{n+1}\boldsymbol{P}_{p,q}^{k-1} N_{p,3}(u)N_{q,3}(v)$, and iterative format of surface can be written as follows

$$\boldsymbol{P}_{i,j}^k = \boldsymbol{P}_{i,j}^{k-1} + \boldsymbol{\Delta}_{i,j}^{k,k-1}, i = 1, \cdots, m; j = 1, \cdots, n; k = 1, 2, \cdots, \tag{6}$$

in which, the difference vector is calculated as

$$\boldsymbol{\Delta}_{i,j}^{k,k-1} = \boldsymbol{Q}_{i,j} - \boldsymbol{C}^{k,k-1}(u_i, v_j), i = 1, \cdots, m; j = 1, \cdots, n; k = 1, 2, \cdots.$$

Then we make an improvement on the Eq. (6) by adding a weight $\frac{\omega}{N_{i,3}(u_i)N_{j,3}(v_j)}$ to the iterative difference vector as follows

$$\begin{aligned} \boldsymbol{P}_{i,j}^k = \boldsymbol{P}_{i,j}^{k-1} + \frac{\omega}{N_{i,3}(u_i)N_{j,3}(v_j)}\boldsymbol{\Delta}_{i,j}^{k,k-1}, i = 1, \cdots, m; \\ j = 1, \cdots, n; k = 1, 2, \cdots, \end{aligned} \tag{7}$$

The iterative process expressed in Eq. (7) is progressive iterative approximation format of successive over-relaxation for non-uniform B-spline surface interpolation called SOR-PIA of surface. It possesses properties similar to that of curve iterative interpolation expressed in Eq. (5).

Similar to the case of curve, Eq. (7) can also be written as follows

$$\begin{aligned} \boldsymbol{P}_{i,j}^k &= \boldsymbol{P}_{i,j}^{k-1} + \frac{\omega}{N_{i,3}(u_i)N_{j,3}(v_j)}\boldsymbol{\Delta}_{i,j}^{k,k-1} = \boldsymbol{P}_{i,j}^{k-1} + \frac{\omega(\boldsymbol{Q}_i - \boldsymbol{C}^{k,k-1}(u_i, v_j))}{N_{i,3}(u_i)N_{j,3}(v_j)} \\ &= (1-\omega)\boldsymbol{P}_{i,j}^{k-1} + \omega\frac{\begin{pmatrix} \boldsymbol{Q}_{i,j} - \boldsymbol{P}_{i,j-1}^k N_{i,3}(u_i)N_{j-1,3}(v_j) - \boldsymbol{P}_{i,j+1}^{k-1} N_{i,3}(u_i)N_{j+1,3}(v_j) - \\ \sum_{q=j-1}^{j+1}((\boldsymbol{P}_{i-1,q}^k N_{i-1,3}(u_i) + \boldsymbol{P}_{i+1,q}^{k-1} N_{i+1,3}(u_i))N_{q,3}(v_j)) \end{pmatrix}}{N_{i,3}(u_i)N_{j,3}(v_j)}. \end{aligned}$$

When $i = 1, 2, \cdots, m, j = 1, 2, \cdots, n$, Eq. (7) can also be written in matrix form

$$\boldsymbol{P}^k = (1-\omega)\boldsymbol{P}^{k-1} + \omega \boldsymbol{D}^{-1}\left(\boldsymbol{Q} - \boldsymbol{L}\boldsymbol{P}^k - \boldsymbol{U}\boldsymbol{P}^{k-1}\right), k = 1, 2, \cdots,$$

in which, $\boldsymbol{D} + \boldsymbol{L} + \boldsymbol{U} = \boldsymbol{N}$, where $\boldsymbol{D}$ is a diagonal matrix, $\boldsymbol{L}$ is a strictly lower triangular matrix, $\boldsymbol{U}$ is a strictly upper triangular matrix, $\boldsymbol{N} = \boldsymbol{N}_1 \otimes \boldsymbol{N}_2$.

3.3 Computing the Relaxation Factor

As described in Sects. 3.1 and 3.2, the matrix form of SOR-PIA for non-uniform B-spline curve and surface interpolation can be written as

$$\begin{aligned}\boldsymbol{P}^k &= (1-\omega)\boldsymbol{P}^{k-1} + \omega\boldsymbol{D}^{-1}\left(\boldsymbol{Q} - \boldsymbol{L}P^k - \boldsymbol{U}P^{k-1}\right)\\ &= (\boldsymbol{D}+\omega\boldsymbol{L})^{-1}\left((1-\omega)\boldsymbol{D} - \omega\boldsymbol{U}\right)\boldsymbol{P}^{k-1} + \omega(\boldsymbol{D}+\omega\boldsymbol{L})^{-1}\boldsymbol{Q}, k = 1, 2, \cdots.\end{aligned} \tag{8}$$

Let $\boldsymbol{B}_\omega = (\boldsymbol{D}+\omega\boldsymbol{L})^{-1}\left((1-\omega)\boldsymbol{D} - \omega\boldsymbol{U}\right)$, and $\boldsymbol{l}_\omega = \omega(\boldsymbol{D}+\omega\boldsymbol{L})^{-1}\boldsymbol{Q}$, the Eq. (8) can be written as

$$\boldsymbol{P}^k = \boldsymbol{B}_\omega\boldsymbol{P}^{k-1} + \boldsymbol{l}_\omega, k = 1, 2, \cdots, \tag{9}$$

where $\boldsymbol{B}_\omega$ is the iteration matrix. It is obvious that $\rho(\boldsymbol{B}_\omega) \geq |\omega - 1|$. If Eq. (9) is convergent, then $|\omega - 1| \leq \rho(\boldsymbol{B}_\omega) < 1$, i.e. $0 < \omega < 2$.

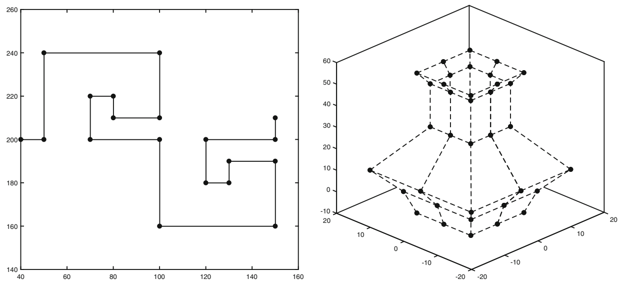
The speed of over-relaxation iteration is closely related to the relaxation factor. There is currently no automatic method for the selection of the relaxation factor. A common practice is to select multiple different values within [0,2] and compare them. Here, we adopt an equidistant search strategy with an interval of 0.05.

4 Experiments

In this section, we employ two examples to illustrate the effectiveness of the proposed SOR-PIA method and compare it with methods of PIA [1,2] and WPIA [6]. First of all, we give two test examples, namely, an example of iterative curve interpolation and an example of iterative surface interpolation. The fitting curve and surface we used are all non-uniform cubic B-spline curve and surface. The parameterization of data points is determined by the cumulative chord length parameterization method, and the knot vector is determined by the parameters of data points. The fitting error is measured by the Euclidean distance between the data points to be interpolated and the points corresponding to the parameter values on the curve or surface.

Example 1. 19 data points are taken in plane to constitute a 19×1 sequence, and shown in Fig. 1(a): (40, 200); (50, 200); (50, 240); (100, 240); (100, 210); (80, 210); (80, 220); (70, 220); (70, 200); (100, 200); (100, 160); (150, 160); (150, 190); (130, 190); (130, 180); (120, 180); (120, 200); (150, 200); (150, 210).

Example 2. 40 data points are taken in space to constitute a 7×9 matrix, shown in Fig. 1(b): (0, 0, 0);(0, –8, –10), (8, –8, –10), (8, 0, –10), (8, 8, –10), (0, 8, –10), (–8, 8, –10), (–8, 0, –10), (–8, –8, –10), (0, –8, –10); (0, –10, 0), (10, –10, 0), (10, 0, 0), (10, 10, 0), (0, 10, 0), (–10, 10, 0), (–10, 0, 0), (–10, –10, 0), (0, –10, 0); (0, –15, 10), (15, –15, 10), (15, 0, 10), (15, 15, 10), (0, 15, 10), (–15, 15, 10), (–15, 0, 10), (–15, –15, 10), (0, –15, 10); (0, –6, 30), (6, –6, 30), (6, 0, 30), (6, 6, 30), (0, 6, 30), (–6, 6, 30), (–6, 0, 30), (–6, –6, 30), (0, –6, 30); (0, –6, 50), (6, –6, 50), (6, 0, 50), (6, 6, 50), (0, 6, 50), (–6, 6, 50), (–6, 0, 50), (–6, –6, 50), (0, –6, 50); (0, –8, –55), (8, –8, –55), (8, 0, –55), (8, 8, –55), (0, 8, –55), (–8, 8, –55), (–8, 0, –55), (–8, –8, –55), (0, –8, –55).

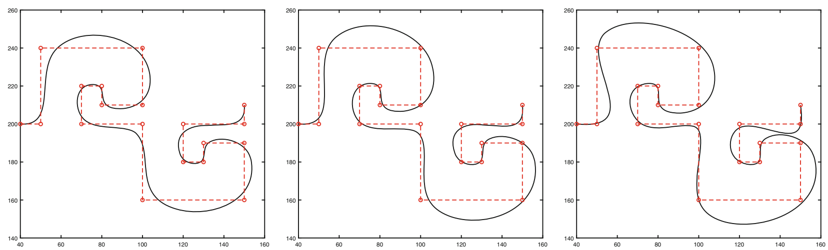


(a) Curve example (b) Surface example

Fig. 1. Test examples.

The experimental results of the first 15 iteration levels are shown in Tables 1 and 2, where PIA is the standard iterative interpolation method, WPIA is the PIA method with a acceleration weight, SOR-PIA_1 is our method with an arbitrary relaxation factor, SOR-PIA_2 is our method with a relaxation factor calculated by using equidistant search strategy. In Table 1, the relaxation factor is 1.1 in SOR-PIA_1 and it is 1.05 in SOR-PIA_2. In Table 2, the relaxation factor is 1.2 in SOR-PIA_1 and it is 1.1 in SOR-PIA_2. Based on the nature of the iterative process, it is generally believed that the smaller the error, the faster the iteration speed. By comparison, we can find that the SOR_PIA proposed in this paper is much faster than PIA and WPIA, and it is effective to calculate the relaxation factor by using the equidistant search strategy.

To visually show the performance of the three methods, we compare the results after the first iteration, as shown in Fig. 2 and Fig. 3. The reason for choosing to compare the visual results of the first iteration is that with the increase of the number of iterations, the visual results of the subsequent iterations can hardly be distinguished by the naked eye. And the relaxation factor of SOR-PIA is 1.05 in Fig. 2. In Fig. 3, the relaxation factor is 1.1. As can be seen from these figures, the iterative interpolation effect of SOR-PIA is significant.

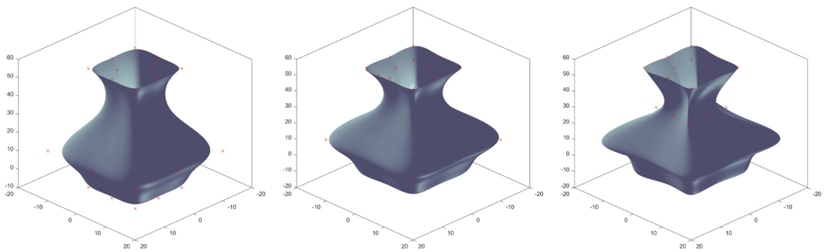


(a) PIA (b) WPIA (c) SOR-PIA

Fig. 2. Iterative interpolation in different methods of example 1.

Table 1. The iterative errors of Example 1

Level	PIA	WPIA	SOR-PIA_1	SOR-PIA_2
0	1.114 199e+02	1.114 199e+02	1.114 199e+02	1.114 199e+02
1	5.496 593e+01	3.267 479e+01	2.881 771e+01	2.745 255e+01
2	3.045 766e+01	1.312 380e+01	8.007 956e+00	7.344 945e+00
3	1.774 285e+01	6.003 406e+00	2.161 996e+00	2.020 399e+00
4	1.067 390e+01	2.843 402e+00	5.918 348e–01	5.670 205e–01
5	6.757 224e+00	1.412 444e+00	1.664 159e–01	1.432 362e–01
6	4.417 398e+00	7.102 530e–01	4.051 463e–02	3.272 078e–02
7	2.908 802e+00	3.617 117e–01	8.886 908e–03	6.845 107e–03
8	1.926 791e+00	1.851 538e–01	2.328 287e–03	1.383 686e–03
9	1.283 424e+00	9.518 904e–02	6.213 399e–04	2.749 991e–04
10	8.592 878e–01	4.899 593e–02	1.485 333e–04	6.815 204e–05
11	5.779 434e–01	2.528 939e–02	3.353 955e–05	1.988 929e–05
12	3.902 047e–01	1.305 820e–02	8.514 881e–06	5.739 107e–06
13	2.642 543e–01	6.756 562e–03	2.495 848e–06	1.405 999e–06
14	1.793 757e–01	3.495 290e–03	6.867 457e–07	2.930 517e–07
15	1.219 730e–01	1.811 265e–03	1.710 329e–07	5.940 200e–08



(a) PIA (b) WPIA (c) SOR-PIA

Fig. 3. Iterative interpolation in different methods of example 2.

Table 2. The iterative errors of Example 2

Level	PIA	WPIA	SOR-PIA_1	SOR-PIA_2
0	1.491 585e+02	1.491 585e+02	1.491 585e+02	1.491 585e+02
1	7.611 221e+01	4.922 341e+01	7.385 814e+01	6.129 211e+01
2	4.328 730e+01	2.795 687e+01	3.935 802e+01	2.789 917e+01
3	2.791 393e+01	1.486 221e+01	2.061 524e+01	1.217 103e+01
4	1.914 628e+01	9.058 060e+00	1.038 640e+01	5.080 306e+00
5	1.342 518e+01	5.846 255e+00	5.177 180e+00	2.045 256e+00
6	9.538 961e+00	3.848 654e+00	2.405 970e+00	7.489 668e–01
7	6.872 865e+00	2.644 782e+00	1.022 007e+00	2.500 636e–01
8	5.042 845e+00	1.833 794e+00	4.040 267e–01	7.804 021e–02
9	3.789 818e+00	1.283 230e+00	1.410 842e–01	1.974 009e–02
10	2.918 993e+00	9.107 264e–01	4.056 066e–02	4.886 369e–03
11	2.290 105e+00	6.444 096e–01	9.181 990e–03	1.281 902e–03
12	1.821 722e+00	4.623 262e–01	1.671 155e–03	5.012 722e–04
13	1.465 099e+00	3.298 248e–01	3.670 073e–04	1.940 344e–04
14	1.188 773e+00	2.379 626e–01	1.140 689e–04	6.683 711e–05
15	9.715 385e–01	1.708 819e–01	4.277 461e–05	2.057 718e–05

5 Conclusion

Above test results demonstrate that the successive over-relaxation iteration method (SOR-PIA) proposed in this paper for the purpose of iterative interpolation of non-uniform cubic B-spline curve and surface is much more accurate and therefore faster than the standard PIA interpolation method and the weighted PIA (WPIA) interpolation method. In the future, we will try to prove its convergence theoretically.

References

1. Lin, H., Wang, G., Dong, C.: Constructing iterative non-uniform B-spline curve and surface to fit data points. Sci. China Ser. F **47**(3), 315–331 (2004)
2. Lin, H., Bao, H., Wang, G.: Totally positive bases and progressive iteration approximation. Comput. Math. Appl **50**(3/4), 575–586 (2005)
3. Lin, H., Zhang, Z.: An extended iterative format for the progressive-iteration approximation. Comput. Graph. **35**(5), 967–975 (2011)
4. Deng, C., Lin, H.: Progressive and iterative approximation for least squares B-spline curve and surface fitting. Comput.-Aided Des. **47**(1), 32–44 (2014)
5. Lu, L.: Weighted progressive iteration approximation and convergence analysis. Comput. Aided Geom. Des. **27**(2), 129–137 (2010)
6. Lin, H., Zhang, Z.: An efficient method for fitting large data sets using T-splines. SIAM J. Sci. Comput. **35**(6), A3052–A3068 (2013)

7. Lin, H., Jin, S., Liao, H., et al.: Quality guaranteed all-hex mesh generation by a constrained volume iterative fitting algorithm. Comput.-Aided Des. **67–68**(C), 107–117 (2015)
8. Lin, H., Jin, S., Hu, Q., Liu, Z.: Constructing B-spline solids from tetrahedral meshes for isogeometric analysis. Comput. Aided Geom. Des. **35–36**, 109–120 (2015)
9. Galveza, A., Iglesiasa, A., Avilaa, A., Oteroc, C., Ariasc, R., Manchadoca, C.: Elitist clonal selection algorithm for optimal choice of free knots in B-spline data fitting. Appl. Soft Comput. **26**, 90–106 (2015)
10. Zhang, L., Ge, X., Tan, J.: Least square geometric iterative fitting method for generalized B-spline curves with two different kinds of weights. Visual Comput. **32**(9), 1109–1120 (2015). https://doi.org/10.1007/s00371-015-1170-3
11. Liu, M., Li, B., Guo, Q., Zhu, C., Hu, P., Shao, Y.: Progressive iterative approximation for regularized least square bivariate B-spline surface fitting. J. Comput. Appl. Math. **327**, 175–187 (2018)
12. Lin, H., Maekawa, T., Deng, C.: Survey on geometric iterative methods and their applications. Comput.-Aided Des. **95**, 40–51 (2018)
13. Hu, L., Shou, H., Dai, Z.: HSS-iteration-based iterative interpolation of curves and surfaces with NTP bases. Wirel. Netw., 1–13 (2020). https://doi.org/10.1007/s11276-019-02224-y