Computer-Aided Design 95 (2018) 40–51

Contents lists available at ScienceDirect

Computer-Aided Design

journal homepage: www.elsevier.com/locate/cad

Survey on geometric iterative methods and their applications

Hongwei Lin [a,*], Takashi Maekawa [b], Chongyang Deng [c]

[a] School of Mathematical Science, State Key Lab. of CAD&CG, Zhejiang University, Hangzhou, 310027, China
[b] Department of Mechanical Engineering, Yokohama National University, Japan
[c] School of Science, Hangzhou Dianzi University, Hangzhou, 310018, China

ARTICLE INFO

Article history:
Received 8 June 2017
Accepted 9 October 2017

Keywords:
Geometric iterative method
Progressive–iterative approximation
Geometric interpolation/approximation
Geometric design

ABSTRACT

Geometric iterative methods (GIM), including the progressive–iterative approximation (PIA) and the geometric interpolation/approximation method, are a class of iterative methods for fitting curves and surfaces with clear geometric meanings. In this paper, we provide an overview of the interpolatory and approximate geometric iteration methods, present the local properties and accelerating techniques, and show their convergence. Moreover, because it is easy to integrate geometric constraints in the iterative procedure, GIM has been widely applied in geometric design and related areas. We survey the successful applications of geometric iterative methods, including applications in geometric design, data fitting, reverse engineering, mesh and NURBS solid generation.

1. Introduction

GIM is an iterative method with clear geometric meanings. From an initial curve (surface), through iterative adjustments of the control points, the curve (surface) in the limit can interpolate or approximate a given data point set. Most traditional fitting methods entail solving a global linear system and therefore local modifications are not possible. Specifically, even the change of a single point that is to be fitted could necessitate solving the global linear system from the start, thus considerably wasting computational resources. With GIM, the linear systems in most traditional fitting methods are avoided. If some data points are changed, the new fitting procedure can begin using the result generated in the previous round of GIM iterations. Moreover, because there are clear geometric meanings in the GIM iteration procedure, it is easy to integrate geometric constraints in each GIM iteration step, and then ensure the limit curve or surface satisfies these geometric constraints. Recently, owing to its desirable properties, GIM has been successfully applied in geometric design and related areas, including adaptive data fitting, large scale data fitting, symmetric surface fitting, generation of curves interpolating given positions, tangent vectors, and curvature vectors, generation of quadrilateral and hexahedral meshes with guaranteed quality, generation of trivariate B-spline solids.

Studies on iterations with geometric meanings began in the 1970s. In 1975, Qi and co-authors developed the "profit and loss" algorithm for uniform cubic B-spline curves, and showed its convergence [1]. Later, Yamaguchi independently proposed an algorithm similar to [1] in 1977 [2]. In 1979, de Boor re-invented the algorithm [3]. Over 20 years later, Lin et al. proved the convergence of the "profit and loss" algorithm for nonuniform cubic B-spline curves and surfaces in 2004 [4]. In 2005, he and co-authors showed that the algorithm is convergent for curves and surfaces with normalized and totally positive basis, and coined the terminology *progressive–iterative approximation* (PIA) to name the iterative method [5]. In PIA, the parameters of foot points in a fitting curve (surface) are fixed in iterations, where foot points are the points in a fitting curve (surface) with the same parameters as those of corresponding data points. Thus, iterations in PIA depend on the *parametric distance*. On the other hand, there are also GIM methods that rely on the *geometric distance*. In 2007, Maekawa et al., developed an iteration method, called *geometric interpolation* (GI) [6], that considers the points in a fitting curve (surface) closest to corresponding data points as the foot points. Thus, the geometric distance, i.e., the distance between a data point and its closest point in the fitting curve (surface), is employed in iterations of GI. Parameter correction is a widely used technique to adjust the parameter value iteratively in order to force error vectors to be perpendicular to the interpolating curve (surface) [7]. Therefore, it can be said that GI conducts parameter correction in each iteration in an indirect way by moving each control point parallel to the corresponding error vector. Moreover, the convergence of the GI iterations was proved in [8,9].

Because the PIA and GI methods are similar, both bearing clear geometric meanings, we collectively referred to them as *geometric iterative methods* (GIM).

[*] Corresponding author.
E-mail address: hwlin@zju.edu.cn (H. Lin).

https://doi.org/10.1016/j.cad.2017.10.002
0010-4485/© 2017 Elsevier Ltd. All rights reserved.

Table 1
GIMs for different types of curves and surfaces.

Type of curves or surfaces		Interpolation		Approximation	
		PD[a]	GD[b]	PD[a]	GD[b]
NURBS	Uniform cubic B-spline curves	[1–3,24]			
	Non-uniform B-spline curves/surfaces	[4]	[6,8,9]	[36,38]	[27]
	NURBS curve	[12]			
Normalized and totally positive basis		[5]			
Triangular Bernstein–Bézier surface		[13,16]			
Subdivision surfaces	Doo-Sabin surfaces	[18,20]			
	Loop surfaces	[19,21]	[6]		[44]
	Catmull-Clark surfaces	[22]	[6]		
Non-standard TP basis[c]	Wang-Ball curve	[32]			
	Said-Ball surface on triangular domain	[33]			
	Triangular Bézier surface	[34]			

[a] PD: parametric distance.

[b] GD: geometric distance.

[c] TP basis: totally positive basis.

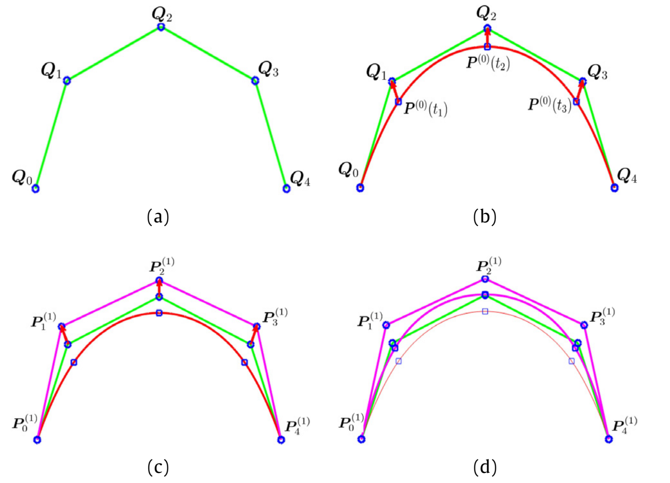


Fig. 1. The iteration procedure of the interpolatory GIM. (a) Data points $\boldsymbol{Q}_i$, $i = 0, 1, 2, 3, 4$ and the initial control polygon. (b) Generation of the difference vectors $\boldsymbol{\Delta}_i^{(0)} = \boldsymbol{Q}_i - \boldsymbol{P}^{(0)}(t_i)$, $i = 0, 1, 2, 3, 4$. (c) Generation of the new control points $\boldsymbol{P}_i^{(1)}$, $i = 0, 1, 2, 3, 4$ by adding the difference vectors to the corresponding control points. (d) Generation of the new curve.

In this paper, we survey the theories and applications of GIM comprehensively. The remainder of this paper is organized as follows. In Section 2, interpolatory GIMs are introduced. Next, approximating GIMs are reviewed in Section 3. Then, the applications of GIMs in geometric design and related areas are investigated in Section 4. Finally, Section 5 concludes the paper.

2. Interpolatory geometric iterative methods

In this section, we review interpolatory GIMs for some commonly employed curves and surfaces, including curves and surfaces with normalized and totally positive basis, NURBS curves and surfaces, triangular Bernstein–Bézier (B–B) curves and surfaces, and subdivision surfaces. Additionally, the iterative acceleration techniques for GIMs are also introduced. Finally, in conclusion to this section, GIMs for different types of curves/surfaces are listed in Table 1.

2.1. Interpolatory GIMs for curves and surfaces

Consider a data point sequence (Fig. 1(a)),

$$\{\boldsymbol{Q}_i \in \mathbb{R}^3, i = 0, 1, \ldots, n\}, \tag{1}$$

where each data point $\boldsymbol{Q}_i$ is assigned a parameter value $t_i, i = 0, 1, \cdots, n$, satisfying,

$$t_0 < t_1 < \cdots < t_n.$$

Considering point sequence (1) as the initial control point sequence, we can construct an initial curve,

$$\boldsymbol{P}^{(0)}(t) = \sum_{i=0}^{n} \boldsymbol{P}_i^{(0)} B_i(t), \tag{2}$$

where $\boldsymbol{P}_i^{(0)} = \boldsymbol{Q}_i$, $B_i(t), i = 0, 1, \ldots, n$ are basis functions. Suppose that the αth curve $\boldsymbol{P}^{(\alpha)}(t)$, generated after the αth iteration, is,

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

To perform the $(\alpha + 1)$th iteration, we first calculate the difference vectors (Fig. 1(b)),

$$\boldsymbol{\Delta}_i^{(\alpha)} = \boldsymbol{Q}_i - \boldsymbol{P}^{(\alpha)}(t_i), \ i = 0, 1, \ldots, n, \tag{3}$$

and add them to the corresponding control points of the αth curve $\boldsymbol{P}^{(\alpha)}(t)$, yielding the control points of the $(\alpha + 1)$th curve (Fig. 1(c)),

$$\boldsymbol{P}_i^{(\alpha+1)} = \boldsymbol{P}_i^{(\alpha)} + \boldsymbol{\Delta}_i^{(\alpha)}.$$

Thus, the $(\alpha + 1)$th curve is obtained (Fig. 1(d)):

$$\boldsymbol{P}^{(\alpha+1)}(t) = \sum_{i=0}^{n} \boldsymbol{P}_i^{(\alpha+1)} B_i(t).$$

The above interpolatory GIM procedure yields a curve sequence:

$$\{\boldsymbol{P}^{(\alpha)}(t), \ \alpha = 0, 1, \ldots\}. \tag{4}$$

Similarly, a surface sequence can be produced by the interpolatory GIM procedure [4,5].

Denote

$$\varDelta^{(\alpha)} = [\boldsymbol{\Delta}_0^{(\alpha)}, \boldsymbol{\Delta}_1^{(\alpha)}, \ldots, \boldsymbol{\Delta}_n^{(\alpha)}]^T.$$

The matrix form of the aforementioned interpolatory GIM can be represented as,

$$\varDelta^{(\alpha+1)} = (I - C)\varDelta^{(\alpha)},$$

where I is an identity matrix, and

$$C = \begin{bmatrix} B_0(t_0) & B_1(t_0) & \cdots & B_n(t_0) \\ B_0(t_1) & B_1(t_1) & \cdots & B_n(t_1) \\ \vdots & \vdots & & \vdots \\ B_0(t_n) & B_1(t_n) & \cdots & B_n(t_n) \end{bmatrix} \tag{5}$$

is the collocation matrix of the basis functions $B_i(t),\ i = 0, 1, \ldots, n$ at $t_j, j = 0, 1, \ldots, n$.

In Ref. [4], it was shown that, when the basis functions of the curves are nonuniform cubic B-spline basis functions, the curve sequence (4) converges to the curve interpolating the given data point sequence (1), i.e.,

$$\lim_{\alpha\to\infty} \boldsymbol{P}^{(\alpha)}(t_i) = \boldsymbol{Q}_i,\ i = 0, 1, \ldots, n.$$

The nonuniform bi-cubic B-spline surface sequence is also convergent to the surface interpolating the given data point array [4]. It is called the PIA property of curves and surfaces. Moreover, in Ref. [5], it was shown that the curve and surface sequence generated by the interpolatory GIM is convergent if the basis is normalized and totally positive. Therefore, Bézier curves and surfaces and B-spline curves and surfaces both have the PIA property. In Fig. 2, a PIA iteration procedure is illustrated, where a sequence of data points are fitted by a piece of cubic B-spline curve.

In particular, when the curve is a periodic uniform B-spline curve, the expression of the limit curve of the PIA iterations can be determined explicitly [10]. Additionally, the fitting error estimation formula for the PIA iterations was developed in [11]. Given the fitting precision and the parameters for data points, the iteration count can be calculated in advance, using the error estimation formula [11].

2.2. Interpolatory GIM for NURBS curves

The homogeneous expression of a NURBS curve in the projective space is,

$$\tilde{\boldsymbol{P}}(t) = \sum_{i=0}^{n} \tilde{\boldsymbol{P}}_i B_i(t), \tag{6}$$

where, $\tilde{\boldsymbol{P}}_i = (w_i\boldsymbol{P}_i, w_i) = (w_ix_i, w_iy_i, w_iz_i, w_i)$, $w_i \geq 0$ are the weights, $\boldsymbol{P}_i = (x_i, y_i, z_i) \in \mathbb{R}^3$, and $B_i(t), i = 0, 1, \ldots, n$ are the Bernstein basis or B-spline basis functions. The homogeneous expression (6) corresponds to a rational curve in the Euclidean space,

$$\boldsymbol{P}(t) = \frac{\sum_{i=0}^{n} w_i\boldsymbol{P}_iB_i(t)}{\sum_{i=0}^{n} w_iB_i(t)}.$$

Suppose we are given a data point sequence $\{\tilde{\boldsymbol{Q}}_i = (w_i^q\boldsymbol{Q}_i, w_i^q) = (w_i^qx_i^q, w_i^qy_i^q, w_i^qz_i^q, w_i^q), i = 0, 1, \ldots, n\}$ in the homogeneous projective space, where each point is assigned a parameter $t_i, i = 0, 1, \ldots, n$ with $t_0 < t_1 < \cdots < t_n$. By the homogeneous expression, the interpolatory GIM for NURBS curves is the same as that presented in Section 2.1. Then, a curve sequence in the homogeneous expression is generated, i.e.,

$$\{\tilde{\boldsymbol{P}}^{(\alpha)}(t) = \sum_{i=0}^{n} \tilde{\boldsymbol{P}}_i^{(\alpha)}B_i(t), \alpha = 0, 1, \ldots\}.$$

It corresponds to a rational curve sequence in the 3-dimensional Euclidean space,

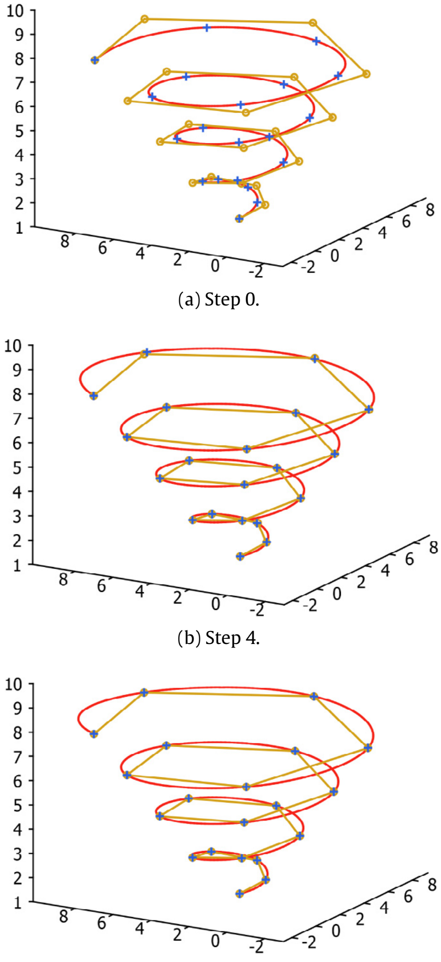
$$\left\{\boldsymbol{P}^{(\alpha)}(t) = \frac{\sum_{i=0}^{n} w_i^{(\alpha)}\boldsymbol{P}_i^{(\alpha)}B_i(t)}{\sum_{i=0}^{n} w_i^{(\alpha)}B_i(t)}, \alpha = 0, 1, \ldots\right\}, \tag{7}$$

and a weight function sequence,

$$\left\{w^{(\alpha)}(t) = \sum_{i=0}^{n} w_i^{(\alpha)}B_i(t), \alpha = 0, 1, \ldots\right\}. \tag{8}$$

Using inequality techniques, it was shown in [12] that the above two sequences (7) and (8) are both convergent, and,

$$\lim_{\alpha\to\infty} \boldsymbol{P}^{(\alpha)}(t_i) = \boldsymbol{Q}_i, \quad \lim_{\alpha\to\infty} w^{(\alpha)}(t_i) = w_i^q, \quad i = 0, 1, \ldots, n. \tag{9}$$



(a) Step 0.

(b) Step 4.

(c) Step 15.

Fig. 2. A sequence of data points are fitted by a cubic B-spline curve using the PIA method. The *circle* marks represent the data points and the blue *plus* marks illustrate the foot points on the B-spline curve with the same parameters as the corresponding data points. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

2.3. Interpolatory GIM for triangular Bernstein–Bézier surfaces

Consider a data point sequence $\{\boldsymbol{Q}_{ijk},\ i, j, k = 0, 1, \ldots, n,\ i + j + k = n\}$, where each point $\boldsymbol{Q}_{ijk}$ corresponds to the parameter (u_i, v_j, w_k) with $u_i + v_j + w_k = 1$. Considering these points as the initial control points, an initial triangular Bernstein–Bézier (B–B) surface can be constructed,

$$\boldsymbol{T}^{(0)}(u, v, w) = \sum_{i,j,k} \boldsymbol{T}_{ijk}^{(0)}B_i(u)B_j(v)B_k(w),$$

where, $\boldsymbol{T}_{ijk}^{(0)} = \boldsymbol{Q}_{ijk}, 0 \leq u, v, w \leq 1, u + v + w = 1$, and $B_i(u), B_j(v), B_k(w)$ are the Bernstein basis functions.

After the αth iteration, suppose the αth triangular B–B surface is generated, i.e.,

$$\boldsymbol{T}^{(\alpha)}(u, v, w) = \sum_{i,j,k} \boldsymbol{T}_{ijk}^{(\alpha)}B_i(u)B_j(v)B_k(w).$$

For the $(\alpha+1)$th triangular B–B surface, we calculate the difference vectors,

$$\boldsymbol{\Delta}_{ijk}^{(\alpha)} = \boldsymbol{Q}_{ijk} - \boldsymbol{T}^{(\alpha)}(u_i, v_j, w_k),\ i, j, k = 0, 1, \ldots, n,\ i + j + k = n,$$

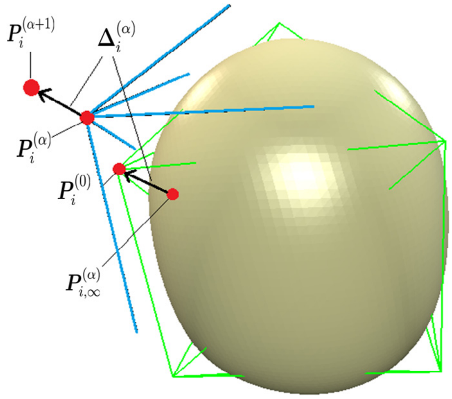


Fig. 3. One iteration of the interpolatory GIM for subdivision surfaces.

and add it to the control points $\boldsymbol{T}_{ijk}^{(\alpha)}$, yielding the control points of the $(\alpha+1)$th triangular B–B surface, i.e.,

$$\boldsymbol{T}_{ijk}^{(\alpha+1)} = \boldsymbol{T}_{ijk}^{(\alpha)} + \boldsymbol{\Delta}_{ijk}^{(\alpha)},\ i, j, k = 0, 1, \dots, n,\ i+j+k = n. \tag{10}$$

Accordingly, the $(\alpha+1)$th triangular B–B surface $\boldsymbol{T}^{(\alpha+1)}(u, v, w)$ is generated.

The iterative procedure stated above generates a sequence of triangular B–B surfaces,

$$\{\boldsymbol{T}^{(\alpha)}(u, v, w),\ 0 \le u, v, w \le 1,\ u+v+w = 1\}. \tag{11}$$

When the parameters corresponding to the data points are uniform, i.e.,

$$(u_i, v_j, w_k) = \left(\frac{i}{n}, \frac{j}{n}, \frac{k}{n}\right),\ i, j, k = 0, 1, \dots, n,\ i+j+k = n,$$

Chen and Wang [13] proved the convergence of the sequence of triangular B–B surfaces (11), namely,

$$\lim_{\alpha\to\infty} \boldsymbol{T}^{(\alpha)}\left(\frac{i}{n}, \frac{j}{n}, \frac{k}{n}\right) = \boldsymbol{Q}_{ijk},\ i, j, k = 0, 1, \dots, n,\ i+j+k = n,$$

by means of the properties of the Bernstein operator and the relation between the operator and matrix. Specifically, eigenvalues of the Bernstein operator lie in the interval (0, 1] [14,15]. It was shown in [13] that eigenvalues of the Bernstein operator and the collocation matrix of $B_i(u)B_j(v)B_k(w)$ on the uniform parameters

$$\left(\frac{i}{n}, \frac{j}{n}, \frac{k}{n}\right),\ i, j, k = 0, 1, \dots, n,\ i+j+k = n,$$

are the same. So the spectral radius of the iterative matrix for the iterative method (10) is less than 1, and then the sequence of triangular B–B surfaces (11) is convergent. In the general case, the convergence of sequence (11) was shown only for the triangular B–B surfaces with degree less than or equal to 4 [16].

2.4. Interpolatory GIMs for subdivision surfaces

In this section, we consider only approximate subdivision formats, including Catmull–Clark, Doo–Sabin, and Loop subdivision surfaces. Suppose that $Q = \{\boldsymbol{Q}_i,\ i = 0, 1, \dots, n\}$ is the vertex set of the control mesh of a subdivision surface. Each control mesh vertex $\boldsymbol{Q}_i$ has its limit position $\boldsymbol{Q}_{i,\infty},\ i = 0, 1, \dots, n$ on the subdivision limit surface, which is the linear combination of some control mesh vertices [17–19], i.e.,

$$\boldsymbol{Q}_{i,\infty} = c_1\boldsymbol{Q}_{i,1} + c_2\boldsymbol{Q}_{i,2} + \cdots + c_k\boldsymbol{Q}_{i,k},\ i = 0, 1, \dots, n.$$

Considering the given mesh Q as the initial control mesh $P^{(0)}$ (Fig. 3), i.e.,

$$P^{(0)} = \{\boldsymbol{P}_i^{(0)},\ i = 0, 1, \dots, n\},\ \text{where, } \boldsymbol{P}_i^{(0)} = \boldsymbol{Q}_i,$$

the initial subdivision surface $S^{(0)}$ can be generated.

As illustrated in Fig. 3, let the control mesh of the αth subdivision surface $S^{(\alpha)}$ after the αth iteration be $P^{(\alpha)} = \{\boldsymbol{P}_i^{(\alpha)},\ i = 0, 1, \dots, n\}$. Each mesh vertex $\boldsymbol{P}_i^{(\alpha)}$ corresponds to a limit position $\boldsymbol{P}_{i,\infty}^{(\alpha)},\ i = 0, 1, \dots, n$ on the subdivision surface $S^{(\alpha)}$. Then the difference vectors are constructed as,

$$\boldsymbol{\Delta}_i^{(\alpha)} = \boldsymbol{Q}_i - \boldsymbol{P}_{i,\infty}^{(\alpha)},\ i = 0, 1, \dots, n.$$

Adding the difference vectors to the corresponding control mesh vertices of $S^{(\alpha)}$, yields the control mesh $P^{(\alpha+1)} = \{\boldsymbol{P}_i^{(\alpha+1)},\ i = 0, 1, \dots, n\}$ of the $(\alpha+1)$th subdivision surface $S^{(\alpha+1)}$, i.e.,

$$\boldsymbol{P}_i^{(\alpha+1)} = \boldsymbol{P}_i^{(\alpha)} + \boldsymbol{\Delta}_i^{(\alpha)},\ i = 0, 1, \dots, n.$$

Accordingly, a sequence of subdivision surfaces $\{S^{(\alpha)}, \alpha = 0, 1, \dots\}$ is generated. When the subdivision surface is a Doo–Sabin subdivision surface [18,20], a Loop subdivision surface [19,21], or a Catmull–Clark subdivision surface without the mesh vertex of degree 3 [22], we have,

$$\lim_{\alpha\to\infty} \boldsymbol{P}_{i,\infty}^{(\alpha)} = \boldsymbol{Q}_i,\ i = 0, 1, \dots, n.$$

In other words, the subdivision surface sequence $\{S^{(\alpha)}, \alpha = 0, 1, \dots\}$ is convergent, and the subdivision limit surface interpolates the given data points $\{\boldsymbol{Q}_i,\ i = 0, 1, \dots, n\}$. Fig. 4 illustrates a PIA iterative procedure of a Loop subdivision surface.

2.5. Iteration speed of the interpolatory GIM and its acceleration

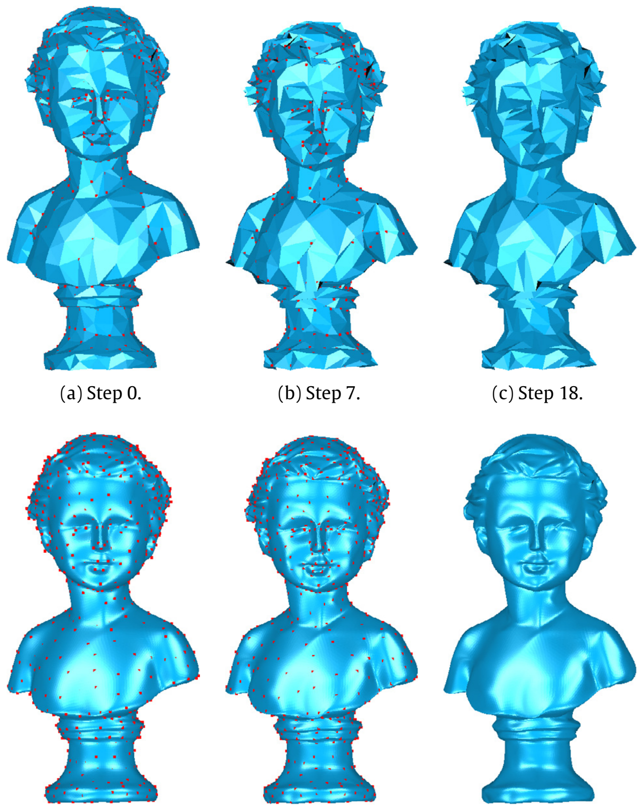
As stated in Section 2.1, the PIA method is convergent when the basis functions are normalized and totally positive. Therefore, the PIA methods for B-spline and Bernstein curves and surfaces are both convergent. Moreover, in the function space formed by a normalized and totally positive basis, the PIA method for the curve or surface with a normalized B-basis has the fastest convergent speed, where a normalized B-basis is the basis with optimal shape preserving properties in the space with a normalized and totally positive basis [23,24].

The PIA method can be accelerated by multiplying a weight ω in front of the difference vectors $\boldsymbol{\Delta}_i^{(\alpha)}, i = 0, 1, \dots, n$ (3). Then, the iterative method changes to,

$$\boldsymbol{P}_i^{(\alpha+1)} = \boldsymbol{P}_i^{(\alpha)} + \omega\boldsymbol{\Delta}_i^{(\alpha)},\ i = 0, 1, \dots, n.$$

Suppose the minimum eigenvalue of the collocation matrix (5) of the basis functions is λ_{min}. When $\omega = \frac{2}{1+\lambda_{min}}$, the PIA method reaches its fastest iteration speed [25,26]. Moreover, Kineri et al. proposed to accelerate the GI method by either fixing closest points or optimal repositioning of control points [27]. Fig. 5 shows the computational time versus the average error with respect to bounding box diagonal of the model with 10,000 and 40,000 data points for the accelerated GIM by [27] and the traditional interpolation method [7]. It can be clearly seen that the accelerated GIM is faster than the traditional interpolation method when the fit is coarse, while it can progressively obtain a finer fit by performing more iterations.

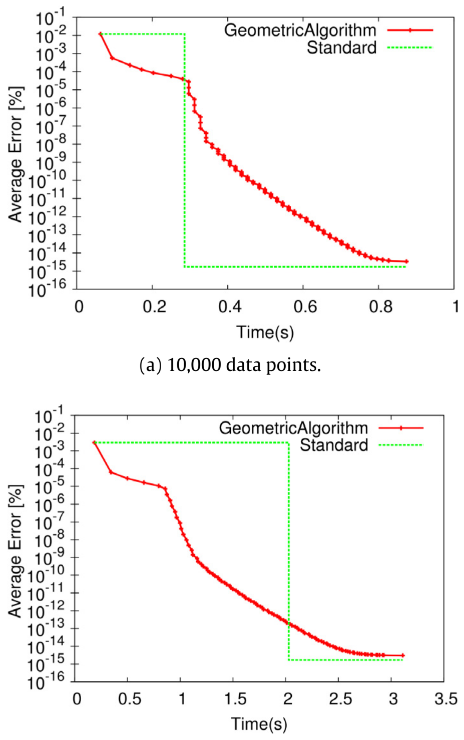
In addition, if the collocation matrix is ill-conditioned, the iterative speed will be very slow. In this case, the iterative speed can be accelerated by introducing the transformation matrix generated by the QR decomposition, and optimizing the spectral radius of the iterative matrix [28].



(a) Step 0. (b) Step 7. (c) Step 18.

(d) Step 0. (e) Step 7. (f) Step 18.

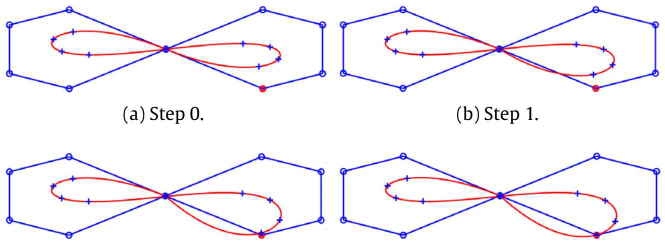
Fig. 4. A PIA iterative procedure of a Loop subdivision surface, where the points in red are the data points with fitting errors beyond the prescribed fitting precision. (a, b, c) The control meshes. (d, e, f) The Loop subdivision surfaces. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)



(a) 10,000 data points.

(b) 40,000 data points.

Fig. 5. Computational time versus average error for the accelerated GIM and the traditional interpolation method.
Source: Adapted from [27].



(a) Step 0. (b) Step 1.

(c) Step 5. (d) Step 10.

Fig. 6. The local property of the interpolatory GIM (adapted from [29]). The circles represent data points, and the red curve is a Bézier curve, where the marks '+' denote the points with the same parameters as those of the corresponding data points. In the iterations, only the control point corresponding to the red data point is adjusted, and the limit curve interpolate the red data point. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

2.6. Local property of the interpolatory GIM

In the interpolatory GIM, if the control points corresponding to a subset of data points are adjusted, and the other control points are fixed, the limit curve or surface will interpolate only the subset of data points. As illustrated in Fig. 6(a), the red curve is an initial Bézier curve, and the circles represent the data points. In the iterative procedure, only the control point corresponding to the data point in red is adjusted. In the limit, the fitting curve in red will interpolate only the data point in red (Fig. 6(d)). This property affords great flexibility to GIM [29,30]. For example, with the local property of GIM, the data points can be fitted adaptively, saving great amount of computations [31].

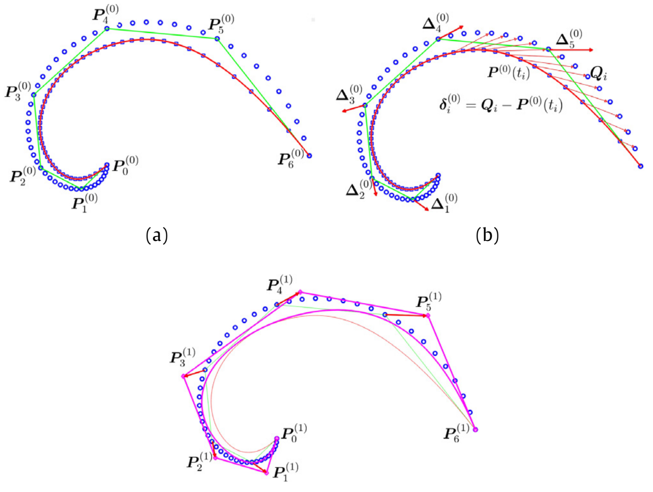
2.7. Generalization of the interpolatory GIM

The interpolatory GIM can be generalized to curves and surfaces with the non-standard totally positive basis [32] (refer to Table 1). If the collocation matrix of a basis is a strictly diagonally dominant matrix or generalized strictly diagonally dominant matrix at some parameter sequence, the basis is called a non-standard totally positive basis [32]. It has been shown that the interpolatory GIM is convergent when the basis is a non-standard totally positive basis [32,33]. Therefore, the interpolatory GIMs for Wang–Ball curves [32], Said-Ball surfaces on triangular domains [33], and T-Bézier triangular surfaces are all convergent [34]. Here, a T-Bézier triangular surface is a quadratic triangular surface defined on a triangular domain with a kind of quadratic quasi-Bernstein basis [34].

In addition, if the weights in front of the difference vectors are allowed to be different, the convergence rates of difference vectors can be controlled individually [32]. Moreover, by multiplying a preconditioning matrix in front of the collocation matrix, the classical PIA, weighted PIA, and the local PIA can be unified into one formula [35].

3. Approximate geometric iterative methods

In interpolatory GIMs, the number of control points of the curve or surface should be equal to that of data points. This requirement restricts the applications of GIMs especially in the cases where the



(a) (b)

(c)

Fig. 7. The iterative procedure of the approximate GIM. (a) Data points (in blue circles), the initial control polygon (in green), and the initial curve (in red). (b) Difference vectors for data points and difference vectors for control points. (c) Generation of the new control polygon (the purple polygon) and the new curve (the purple curve). (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

number of given data points is very large. In this section, we will review the approximate GIMs, where the number of control points can be less than that of the data points, and the limit curve or surface will approximate the given data point set.

Similar to the interpolatory GIM, the approximate GIM can be divided into two categories based on the distance of data points to curves/surfaces to be approximated, namely, the parametric distance and geometric distance. Thus far, on one hand, there are two types of approximate GIMs based on the parametric distance, that is, the extended progressive–iterative approximation (EPIA) [36], and the least-squares progressive–iterative approximation (LSPIA) [37,38]. On the other hand, approximate GIMs based on the geometric distance, called geometric approximation (GA), was studied by Kineri et al. [27].

3.1. Extended progressive–iterative approximation (EPIA)

Suppose we are given a data point sequence $\{\boldsymbol{Q}_i, i = 0, 1, \dots, n\}$, where each data point $\boldsymbol{Q}_i$ is assigned a parameter t_i, $i = 0, 1, \dots, n$ with $t_0 < t_1 < \cdots < t_n$. Then, we construct an initial curve $\boldsymbol{P}^{(0)}(t)$, and classify the data points into groups. Each group of data points corresponds to a control point. As illustrated in Fig. 7(a), there are seven control points, and the data points should be classified into seven groups correspondingly.

Let the αth curve generated after the αth iteration be,

$$\boldsymbol{P}^{(\alpha)}(t) = \sum_{j=0}^{m} \boldsymbol{P}_j^{(\alpha)} B_j(t), \quad m \le n.$$

In each step of the EPIA iterations, the *difference vector for the data point* (DVD) is first produced,

$$\boldsymbol{\delta}_i^{(\alpha)} = \boldsymbol{Q}_i - \boldsymbol{P}^{(\alpha)}(t_i), \quad i = 0, 1, \dots, n. \tag{12}$$

Next, the *difference vector for the control point* (DVC) is generated by weighted averaging the DVDs corresponding to each group of data points, i.e.,

$$\boldsymbol{\Delta}_j^{(\alpha)} = \frac{\sum_{i \in I_j} c_i^j \boldsymbol{\delta}_i^{(\alpha)}}{\sum_{i \in I_j} c_i^j}, \quad j = 0, 1, \dots, m, \tag{13}$$

where, I_j is the index set of the data points in the jth group, and c_i^j are weights. In Fig. 7(b), the group of DVDs $\boldsymbol{\delta}_i^{(0)}$ corresponding to the control point $\boldsymbol{P}_5^{(0)}$ are illustrated, which are weighted averaged to produce the DVC $\boldsymbol{\Delta}_5^{(0)}$. Finally, by adding the DVCs $\boldsymbol{\Delta}_j^{(\alpha)}$ to the corresponding control points $\boldsymbol{P}_j^{(\alpha)}$, the new control points $\boldsymbol{P}_j^{(\alpha+1)}, j = 0, 1, \dots, m$ are generated, as well as the new curve (refer to Fig. 7(c)).

In Ref. [36], we showed that, when $c_i^j = 1$, then $\boldsymbol{\Delta}_j^{(\alpha)}$ is the average vector of the DVDs in the jth group, $j = 0, 1, \dots, m$, EPIA is convergent. It should be pointed out that, in the EPIA method, the selection of the initial curve does not affect the convergence of the EPIA iterations. In addition, in the least-squares progressive–iterative approximation method (refer to Section 3.2), $c_i^j = B_j(t_i)$, that is, the value of the jth basis function $B_j(t)$ at t_i, $i = 0, 1, \dots, n, j = 0, 1, \dots, m$. However, the convergence analysis is an open problem when the weights $c_i^j, i = 0, 1, \dots, n, j = 0, 1, \dots, m$ are taken other values.

In Ref. [36], dominant points [39], which are the data points with local maximum curvatures, are taken as the control points of the initial curve. Denote γ as the curvature of a data point. Data points are classified into groups so that the sums of $|\gamma|$ of the data points in individual groups are as close as possible [36], where $|\gamma|$ is the absolute value of γ. Moreover, if a group contains only one data point, the limit curve will interpolate the data point. In this way, the approximate curve fitted to a data point set can be made to interpolate some data points [36].

3.2. Least-squares progressive–iterative approximation (LSPIA)

LSPIA is actually a special iterative form of EPIA with a special data point grouping manner. Suppose the given data points and their parameterizations are the same as specified in Section 3.1, and the initial curve is,

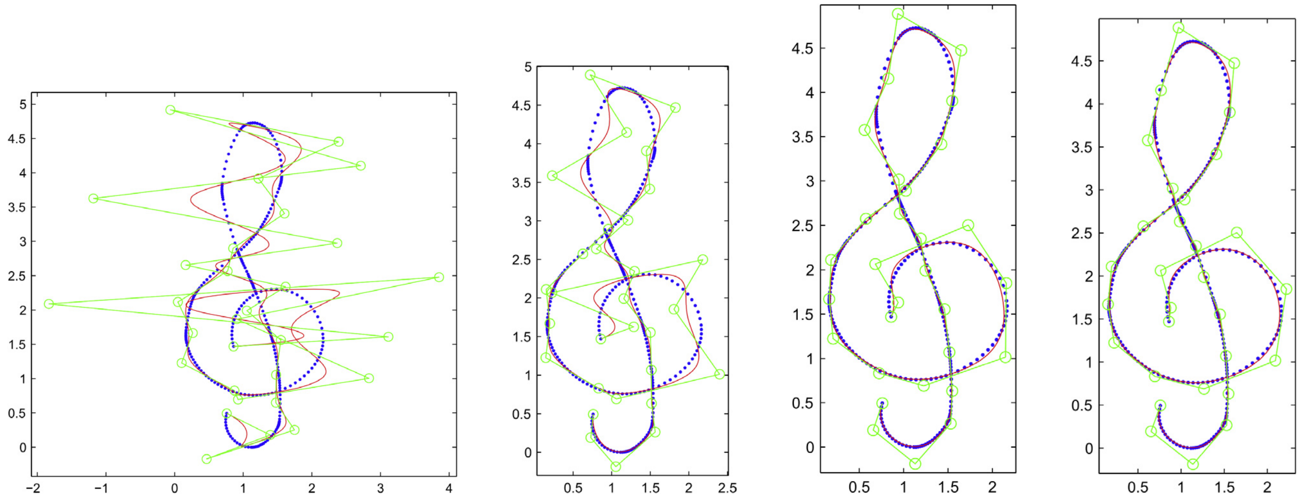
$$\boldsymbol{P}^{(0)}(t) = \sum_{j=0}^{m} \boldsymbol{P}_j^{(0)} B_j(t).$$

Then, all of the data points, whose parameters t_i lie in the local support region of the jth basis function $B_j(t)$, i.e., $B_j(t_i) \neq 0$, are classified into the jth group, corresponding to the jth control point, where $i = 0, 1, \dots, n$, $j = 0, 1, \dots, m$. In the LSPIA method, the weights c_i^j (13) are taken as $B_j(t_i)$, i.e., $c_i^j = B_j(t_i)$, $i = 0, 1, \dots, n$, $j = 0, 1, \dots, m$. Therefore, in LSPIA, the DVCs are generated using,

$$\boldsymbol{\Delta}_j^{(\alpha)} = \frac{\sum_{i \in I_j} B_j(t_i) \boldsymbol{\delta}_i^{(\alpha)}}{\sum_{i \in I_j} B_j(t_i)}, \quad i = 0, 1, \dots, n, \ j = 0, 1, \dots, m,$$

where, I_j is the index set of the data points in the jth group, and $\boldsymbol{\delta}_i^{(\alpha)}$ is a DVD, same as that in Eq. (12). It can be shown that, LSPIA is convergent, and it converges to the least-squares fitting result to the given data points [37,38]. Moreover, Zhang et al. showed that the LSPIA algorithm is also convergent for generalized B-spline curves with two different types of weights [40].

LSPIA has some desirable properties. First, because the convergence of the LSPIA method is unrelated to the initial curves or surfaces, the LSPIA method is insensitive to them. In Fig. 8, a LSPIA iterative procedure is demonstrated. Although the shape of the initial cubic B-spline curve is far different from that of the data point set, after 70 step LSPIA iterations, the B-spline curve approximates the data points faithfully (Fig. 8(d)). Second, when the basis functions of a curve or surface are locally supported, the computational complexity in each LSPIA iteration step is only related to the number of data points, and unrelated to the number of control points (i.e., the unknowns) [37]. Therefore, LSPIA is suitable to fit large data sets incrementally. Finally, LSPIA is so robust that whether its iterative matrix is singular or not, LSPIA is always convergent [37,38].



(a) Step 20.

(b) Step 40.

(c) Step 60.

(d) Step 70.

Fig. 8. A LSPIA iterative procedure with several iteration steps (adapted from [38]). Here, the blue dots are data points, the green circles are control points, and the red curve is a cubic B-spline curve. The two end control points are the same as the two end points of the input data points, and the other initial control points are all taken as the point with coordinates (100, 1), far from the data points within the range [0, 2.5] × [−0.5, 5]. Therefore, the shape of the initial B-spline curve is far different from that of the data point set. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

3.3. Geometric approximation

The algorithm [27] takes a set of randomly distributed data points $\boldsymbol{Q}_k$, $k = 0, \dots, N$ as input and generates a B-spline surface that approximates the data points. The Douglas–Peucker algorithm [41] is used to determine the initial control points of the boundary curves from a series of data points along the boundaries for the iterative geometric fitting algorithm. Farin's technique [42, Chapter 22] is used to generate the base surface for a simple surface, whereas dynamically fitted surfaces to geometries implied by point clouds are used to construct the base surface for a complex surface [43].

Once the base surface is constructed, they compute the error vector $\boldsymbol{Q}_k - \boldsymbol{P}^{(0)}(\hat{u}_k, \hat{v}_k)$ for each data point $\boldsymbol{Q}_k$ where $(\hat{u}_k, \hat{v}_k)$ denotes the parameter value of the closest point. At the parameter value of the closest point $(\hat{u}_k, \hat{v}_k)$ $(u_p < \hat{u}_k < u_{p+1}, v_q < \hat{v}_k < v_{q+1})$ corresponding to $\boldsymbol{Q}_k$, there are $K \times L$ nonzero multiples of basis functions $N_{i,K}(\hat{u}_k)N_{j,L}(\hat{v}_k)$, $i = p - K + 1, \dots, p$, $j = q - L + 1, \dots, q$. As B-spline surfaces are defined as a linear combination of control points and B-spline basis functions, the multiple of the basis functions $N_{i,K}(u)N_{j,L}(v)$ is associated with the control point $\boldsymbol{P}_{ij}$. Note that the sum of the weights is one. In order to distribute the error vector to the appropriate control points, Kineri et al. [27] distribute the error vector $\boldsymbol{Q}_k - \boldsymbol{P}^{(0)}(\hat{u}_k, \hat{v}_k)$ to the control points $\boldsymbol{P}_{ij}^{(0)}$, $i = p - K + 1, \dots, p, j = q - L + 1, \dots, q$ with weights $N_{i,K}(\hat{u}_k)N_{j,L}(\hat{v}_k)$. The difference vector for the control point with index (i, j) is,

$$\begin{aligned}\boldsymbol{\Delta}_{ij}^{(\alpha+1)} &= \sum_{k\in I_{ij}} w_k^{ij}\left(\boldsymbol{Q}_k - \boldsymbol{P}^{(\alpha+1)}(\hat{u}_k, \hat{v}_k)\right)\\ &= \boldsymbol{\Delta}_{ij}^{(\alpha)} - \sum_{g=0}^{n}\sum_{h=0}^{m}\boldsymbol{\Delta}_{gh}^{(\alpha)}\sum_{k\in I_{ij}} w_k^{ij} N_{g,K}(\hat{u}_k)N_{h,L}(\hat{v}_k),\end{aligned}$$

where w_k^{ij} denotes the weight for the error vector and I_{ij} denotes the set of data points $\boldsymbol{Q}_k$ that contribute to the repositioning vector $\boldsymbol{\Delta}_{ij}$. By arranging the difference vectors for the control points in a one-dimensional array,

$$\begin{aligned}\boldsymbol{\Delta}^{(\beta)} &= [\boldsymbol{\Delta}_{00}^{(\beta)}, \boldsymbol{\Delta}_{01}^{(\beta)}, \dots, \boldsymbol{\Delta}_{0m}^{(\beta)}, \dots, \boldsymbol{\Delta}_{10}^{(\beta)}, \dots, \boldsymbol{\Delta}_{1m}^{(\beta)}, \dots, \boldsymbol{\Delta}_{n0}^{(\beta)},\\ &\quad \dots, \boldsymbol{\Delta}_{nm}^{(\beta)}]^T, \qquad (14)\\ \beta &= \alpha, \alpha + 1,\end{aligned}$$

we obtain the iterative form,

$$\boldsymbol{\Delta}^{(\alpha+1)} = (I - C)\boldsymbol{\Delta}^{(\alpha)}. \qquad (15)$$

Since $\|C\|_\infty = 1$, all of the eigenvalues of C are less than or equal to 1. In conclusion, the eigenvalues of C satisfy,

$$0 < \lambda(C) \le 1. \qquad (16)$$

Therefore, the eigenvalues of $I - C$ fulfill,

$$0 \le \lambda(I - C) = 1 - \lambda(C) < 1, \qquad (17)$$

which means that the iterative method (15) is convergent.

Moreover, the geometric approximation can also be applied in the Loop subdivision fitting [44].

3.4. Discussion

In this section, three approximate GIMs, i.e., EPIA, LSPIA, and the geometric approximation, were presented. While EPIA and LSPIA rely on the parametric distance, geometric approximation depends on the geometric distance.

Compared with LSPIA, the grouping manner of EPIA is very flexible [36]. When there is only one point in a group, the limit curve (or surface) generated by EPIA will interpolate the point. Therefore, by EPIA, an approximate curve (or surface) can be made to interpolate some specified data points. However, the convergence of EPIA is proved just for a special case, that all of the weights are equal to 1. The convergence of EPIA in the generic case is an open problem.

Because the grouping manner of LSPIA is fixed, the convergence analysis of LSPIA is easy to be established [37,38], compared with EPIA. It was shown that LSPIA not only converges, but also converges to the least-squares fitting result. Recently, we showed that, LSPIA converges even when the iterative matrix is singular [45]. Because the iteration of LSPIA is related to the number of data points, and unrelated to the number of control points (the unknowns), it is suitable for the large scale data fitting.

Finally, because the geometric approximation method depends on the geometric distance [27], the computation of geometric approximation is more complicated than those of EPIA and LSPIA. However, the fairness of curves and surfaces generated by the geometric approximation method is usually better than those by LSPIA [27].

Table 2
Applications of GIMs.

Geometric design	Degree reduction of Bézier curve	[46,47]
	Rational triangular B–B patch approximation by polynomial	[48]
	Offset curve approximation	[49]
	Hermite interpolation	[50–52]
	Modifying Bézier curve/surface with interpolation constraints	[53]
	Modifying spline surfaces with interpolation constraints	[54]
	Fair curve and surface generation	[55]
Data fitting	Fitting large scale data set by T-spline	[37]
Reverse engineering	Enveloping strip-shaped point cloud	[56]
	Generating high quality class A surface	[57]
	Reconstructing symmetrical surface	[27]
	Reconstructing mesh model with correct topology	[58]
Mesh and solid generation	Quality-guaranteed hexahedron mesh generation	[59,60]
	Planar quadrilateral mesh generation	[61]
	Trivariate B-spline solid generation	[62,63]
Other applications	Satellite image processing	[64]
	Pattern recognition	[65]
	Handwritten curve approximation	[66]
	Rational curve approximation	[67]
	Watermarking NURBS surface	[68]
	Stem form modeling	[69]
	Parametric wing modeling	[70]
	Fitting of component characteristics of turbine engine	[71]
	Shape optimization of vehicle lights	[72]
	Energy absorption characteristics	[73]
	Heterogeneous object modeling	[74]

4. Applications of GIMs

Thus far, GIMs have been extensively applied in academic studies and engineering practices. With GIMs, desirable results are achieved not only in solving classical problems in geometric design, such as offset curves, degree reduction, and polynomial approximation to rational curves and surfaces, but also in related areas, such as adaptive data fitting, large scale data fitting, symmetric surface fitting, generation of curves interpolating given positions, tangent, and curvature vectors, generation of quality guaranteed quadrilateral and hexahedral meshes, generation of trivariate B-spline solids. In this section, we survey the applications of GIMs in these areas. For clarity, the applications of GIMs are listed in Table 2.

4.1. Applications in geometric design

GIMs have been successfully employed to solve various kinds of problems in geometric design. In order to improve the robustness of degree reduction for the Bézier curve, an iterative method was developed in Ref. [46]. Beginning with an initial Bézier curve, this method shifts the control points of the Bézier curve iteratively, and generates the approximate curve with the minimum L_2 error in the limit, thus improving the robustness of degree reduction greatly [46]. Moreover, starting with an initial Bézier curve, and taking $L_p(p = 1, 2, \infty)$ error as the measure, the optimal approximate Bézier curve in $L_p(p = 1, 2, \infty)$ norm can be produced by iteratively adjusting control points of the Bézier curve [47]. Similarly, PIA was also applied in calculating the polynomial surface approximating a rational triangular B–B surface [48].

Furthermore, by sampling points on the offset curve, and taking them as the initial control points, PIA can be utilized to generate the approximate offset curve [49]. This method not only unifies both the polynomial approximation and rational approximation to the offset curve, but also has superiorities with respect to both a lesser number of control points and better approximation error, compared with existing methods [49].

In addition, by adding suitable geometric constraints in the iteration procedure, GIM can be employed to generate a high quality curve that interpolates the given positions, tangent vectors, and curvature vectors simultaneously [50–52] (Fig. 9). Similarly, based on the local property of GIM, Bézier and B-spline surfaces can be modified in real time, and the surfaces after modification can interpolate prescribed data points [53]. In Ref. [54], GIM was designed to modify B-spline surfaces, making the B-spline surfaces interpolate specified positions, tangent vectors, and curvature vectors at knots.

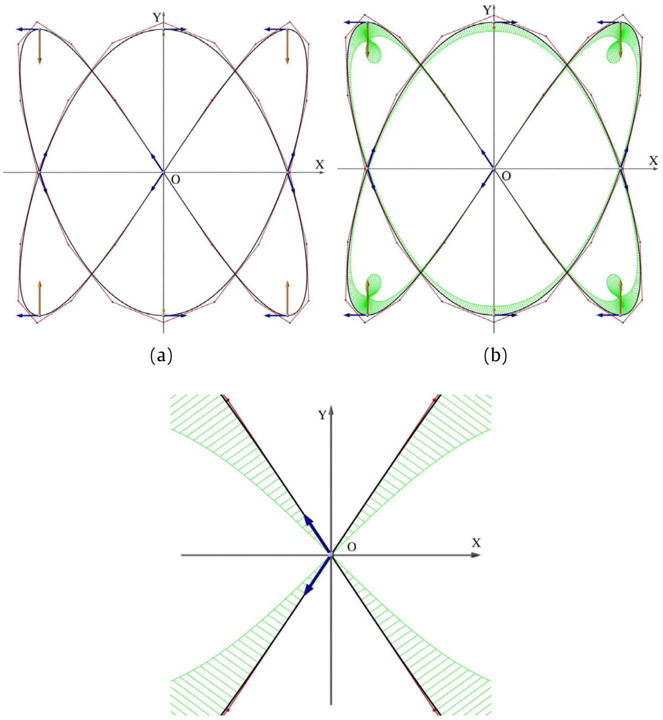


Fig. 9. Using the interpolatory GIM, a uniform B-spline curve can be constructed to interpolate given positions, tangent vectors, and curvature vectors (adapted from [51]). (a) The given positions, tangent vectors, curvature vectors, and the constructed B-spline curve by GIM. (b) The curvature comb plot of the B-spline curve. (c) Close-up view of the curvature comb plot.

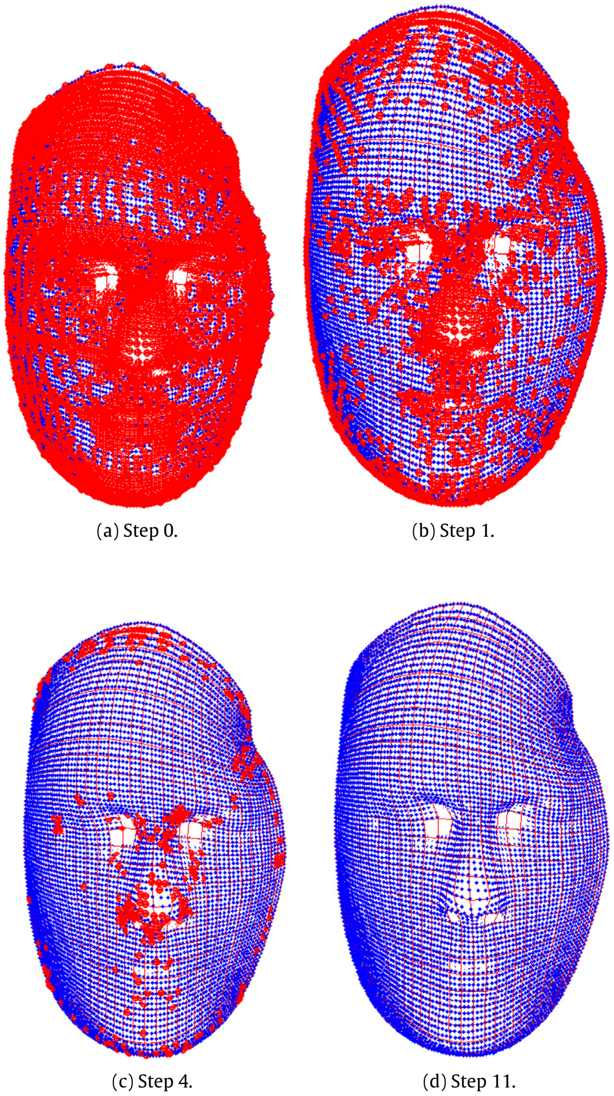


Fig. 10. Adaptive data fitting saves considerable amount of computational resources (adapted from [31]). The data points in red are active data points. While the control points corresponding to the active data points are adjusted in the iterations, the other control points are fixed in iterations. (a) Initially, nearly all of the control points are active. (b) After the first iteration, nearly half of the control points are fixed. (c) After the fourth iteration, only one tenth of the control points are active. (d) After the eleventh iteration, all of the control points are fixed. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

Moreover, the variational PIA method was developed to produce fair curves and surfaces [55].

Implementation of GIM is much easier than the traditional methods, and the fairness of curves and surfaces generated by GIM is usually better than that of the traditional methods. Moreover, the local property of GIM can be employed to improve the fairness of curves and surfaces by fine and local modifications.

4.2. Applications in data fitting

As stated above, the local property of GIM affords great flexibility to data fitting [29,30]. Based on this property, the fitting error for each data point can be controlled individually in the iteration procedure. If the fitting error of a data point achieves the prescribed fitting precision, the control point corresponding to the data point can be fixed in the next iterations. Accordingly, the data points are fitted adaptively. Theoretical analysis and numerical examples show that, adaptive data fitting saves considerable amount of computational resources (Fig. 10) [31]. Moreover, by virtue of the property that the iterative speed of LSPIA is only related to the number of data points (unrelated to the number of control points), and the property that LSPIA is easy to parallelize, a LSPIA algorithm for T-spline was presented in Ref. [37]. Together with the adaptivity of the representation of T-spline, this algorithm is suitable for fitting large scale data sets. For example, it is able to fit high resolution images with 50 million pixels in a common laptop [37]. The adaptivity and the independence of the number of unknowns endow GIM with the capability of large scale data fitting. Although GIM converges slower than traditional iteration methods when the number of unknowns is small, the convergence speed of GIM exceeds traditional methods when the number of unknowns is large enough [37].

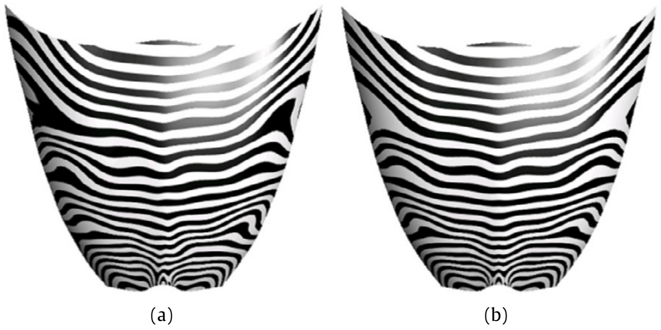


Fig. 11. An automobile hood surface with reflection symmetry (adapted from [27]). (a) Reconstructed surface without symmetry constraints. (b) Reconstructed surface with symmetry constraints.

4.3. Applications in reverse engineering

There are some successful applications of GIMs in reverse engineering. In the field of curve reconstruction, an interval B-spline curve was designed to envelop a strip-shaped point cloud, and the central curve of the interval B-spline curve is taken as the reconstructed curve [56]. In this method, the central curve and the boundary curves of the interval B-spline curve are all generated using PIA [56]. In the field of surface reconstruction, PIA is employed to slightly adjust the control points of a surface, for generating high quality class A surface [57]. Moreover, by continually checking and correcting the topological structure of the reconstructed model in the GIM iterations, it can be guaranteed that the topological structure of the reconstructed model by GIM is correct [58]. Additionally, using GIM, symmetrical surfaces can be reconstructed from given point clouds with symmetry [27]. Specifically, in each iteration step of GIM, the mirror mapping is performed to the left and right part of the control net of the reconstructed surface, and then the original control net and the mirrored control net are averaged to improve the symmetry [27]. Fig. 11(b) shows a reconstructed automobile hood surface with reflection symmetry, (with zebra mapping on it), generated by the GIM method stated above. As a comparison, Fig. 11(a) demonstrates the reconstructed surface without symmetry constraints. In conclusion, because each iteration step of GIM has geometric meanings, constraints are easy to be integrated into the iterations of GIM, such as the symmetry, topological conditions, and then the curves and surfaces generated by GIM can be made to satisfy complicated constraints.

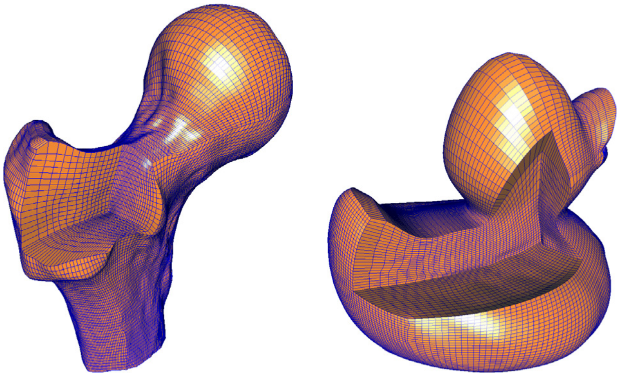

(a) Ball Joint. (b) Duck.

Fig. 12. Two trivariate B-spline solids generated by the LSPIA method.
Source: Adapted from [63].

4.4. Applications in the generation of meshes and NURBS solids

The generation of hexahedral meshes is a challenging task in the field of finite element analysis, because it is difficult to guarantee that there is no overlapping area in the generated hexahedral mesh, i.e., the Jacobian values at mesh vertices are all positive. In such cases, GIM can be employed to generate a hexahedral mesh with a quality guarantee that there is no overlapping area in the generated mesh [59,60]. Specifically, starting with an initial hexahedral mesh without the overlapping area, in each step of the GIM iterations, a feasible region for each mesh vertex is constructed, based on the evident geometric meaning of GIM and the condition ensuring the positive Jacobian values at the vertex and its adjacent vertices. If each mesh vertex does not exceed its feasible region in each iteration, it is guaranteed that there is no overlapping area in the hexahedral mesh [59,60]. A similar method can be utilized to produce the planar quadrilateral mesh without the overlapping area [61]. The generation of quality guaranteed hexahedral meshes by GIM once again demonstrates the capability of GIM in integrating complicated constraints.

With the development of isogeometric analysis [75], it is important to develop efficient methods for generating trivariate NURBS solids. A practical way for generating the trivariate NURBS solid is to fit a given tetrahedral mesh with a NURBS solid. However, the number of tetrahedral mesh vertices is usually very large, which leads to the following two problems: (1) The computations are very complicated; (2) more importantly, the coefficient matrix of the system of fitting equations is of large condition number, even singular. LSPIA is able to solve the system of equations for fitting a given tetrahedral mesh robustly and efficiently, producing desirable trivariate NURBS solids (Fig. 12) [62,63]. In Fig. 12, two trivariate B-spline solids generated by the LSPIA method are illustrated.

4.5. Other applications

In 2013, Natasha et al. published four papers. In these papers, PIA was applied in satellite image processing [64], pattern recognition [65], approximation of handwritten curve [66], and rational curve approximation [67], respectively. Moreover, PIA was also employed in watermarking NURBS surface [68], and stem form modeling [69]. More importantly, PIA was successfully employed in engineering practices, such as the parametric wing modeling [70], fitting of component characteristics of a turbine engine [71], and shape optimization of vehicle lights [72].

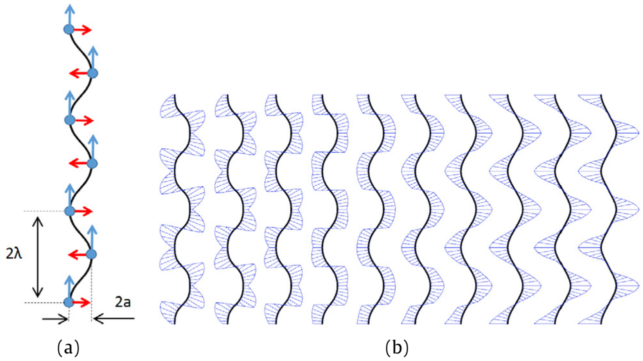


Fig. 13. (a) Generation of a profile curve. (b) The curvature values at the peaks of the wavelike profile curves were gradually increased from left to right, which leads to the formations of a valley between two mountains (Pattern I), a flat hill (Pattern II), and one large steep mountain (Pattern III) in the curvature comb plot for each half-wavelength.
Source: Adapted from [73].

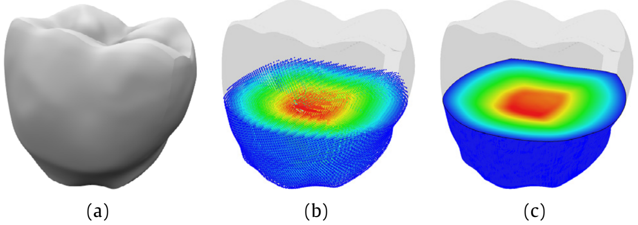


Fig. 14. Fitting of the volumetric attribute data of a tooth model [adapted from [74]]. (a) Geometric model. (b) Color-coded input data based on the distance function. (c) Fitted result. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

Imai et al. [73] studied the effects of curvatures on the energy absorption characteristics of cylindrical corrugated tubes under compression by isogeometric analysis and experiments as shown in Fig. 13. The profile curves of the corrugated tubes were constructed by interpolating a sequence of data points under tangent and curvature vector constraints using the geometric algorithm [51]. They found that curvature distribution patterns of the profile curve significantly affected the energy absorption characteristics of the corrugated tube. Pattern I is shown to have the lowest reaction force and tallest fold, whereas Pattern III has the highest reaction force and shortest fold. This leads to the fact that the Pattern III has the highest energy absorption profile. These results provide a new tool for controlling the energy absorption characteristics of cylindrical corrugated tubes.

Sasaki et al. [74] introduced a framework for modeling heterogeneous objects in terms of trivariate B-spline functions and a method for slicing them directly for additive manufacturing (see Fig. 14). The attribute data are approximated by B-spline ordinates of a graph function whose abscissae are (u, v, w) using the geometric approximation under the assumption that the geometric volume is already defined by the trivariate B-spline functions.

5. Conclusions and future work

Since PIA was presented in 2005, not only has its theory system been continually perfected, but it has also been successfully applied in a number of fields of academic studies and engineering practices. In this paper, on one hand, we surveyed the interpolatory GIMs, their convergence analysis, convergence speed and acceleration, local property, and some generalizations. On the other hand, three kinds of approximate GIMs, i.e., EPIA, LSPIA, and the geometric approximation, were introduced, with their convergence analysis. Moreover, the successful applications of GIMs in several fields were also reviewed in this paper, including applications in geometric design, data fitting, reverse engineering, generation of meshes and NURBS solids. Because GIMs have evident geometric meanings, geometric constraints are easy to be integrated in GIM iterations. Thereby, GIMs can be used to solve some problems that existing methods are unable to solve or cannot solve well. In summary, GIMs have wide applications in academic studies and engineering practices.

Acknowledgments

This paper is supported by the Natural Science Foundation of China (No. 61379072, 61370166), and the Fundamental Research Funds for the Central Universities (2017XZZX009-03).

References

[1] Qi D, Tian Z, Zhang Y, Feng J. The method of numeric polish in curve fitting. Acta Math Sinica 1975;18(3):173–84.

[2] Yamaguchi F. A method of designing free form surfaces by computer display (1st report), Precision machinery,43 (2), 1977, 168–173.

[3] de Boor C. How does agee's smoothing method work? Tech.rep., Washington D C: Army Research Office, 1979.

[4] Lin H, Wang G, Dong C. Constructing iterative non-uniform B-spline curve and surface to fit data points. Sci China Ser F 2004;47(3):315–31.

[5] Lin H, Bao H, Wang G. Totally positive bases and progressive iteration approximation. Comput Math Appl 2005;50(3/4):575–86.

[6] Maekawa T, Matsumoto Y, Namiki K. Interpolation by geometric algorithm. Comput Aided Des 2007;39(4):313–23.

[7] Hoschek J, Lasser D. Fundamentals of computer aided geometric design. Wellesley, (MA): A. K. Peters; 1993.

[8] Lin H. The convergence of the geometric interpolation algorithm. Comput Aided Des 2010;42(6):505–8.

[9] Xiong Y, Li G, Mao A. Convergence analysis for B-spline geometric interpolation. Comput Graph 2012;36(7):884–91.

[10] Deng C. An explicit formula for the control points of periodic uniform spline interpolants and its application. Comput Aided Geom Design 2013;30(4):389–97.

[11] Deng S, Wang G. Error estimation and application of a new class of graphics fitting method. J Zhejiang Univ Eng Sci 2014;48(5):942–56.

[12] Shi L, Wang R. An iterative algorithm of NURBS interpolation and approximation. J Math Res Exposition 2006;26(4):735–43.

[13] Chen J, Wang G. Progressive iterative approximation for triangular Bézier surfaces. Comput Aided Geom Design 2011;43(8):889–95.

[14] Cooper S, Waldron S. The eigenstructure of the Bernstein operator. J Approx Theory 2000;105(1):133–65.

[15] Cooper S, Waldron S. The diagonalisation of the multivariate Bernstein operator. J Approx Theory 2002;117(1):103–31.

[16] Zhao Y, Lin H. The PIA property of low degree non-uniform triangular B-B patches. In: Proceedings of 12th international conference on computer-aided design and computer graphics. IEEE; 2011. p. 239–42.

[17] Halstead M, Kass M, DeRose T. Efficient, fair interpolation using Catmull-Clark surfaces. In: Proceedings of the 20th annual conference on computer graphics and interactive techniques. ACM; 1993. p. 35–44.

[18] Fan F, Cheng F, Lai S. Subdivision based interpolation with shape control. Comput Aided Des Appl 2008;5(1–4):539–47.

[19] Cheng F-HF, Fan F-T, Lai S-H, Huang C-L, Wang J-X, Yong J-H. Loop subdivision surface based progressive interpolation. J Comput Sci Tech 2009;24(1):39–46.

[20] Cheng FF, Fan F, Huang C, Wang J, Lai S, Miura KT. Smooth surface reconstruction using Doo-Sabin subdivision surfaces. In: Proceedings of 3rd international conference on geometric modeling and imaging. IEEE; 2008. p. 27–33.

[21] Cheng FF, Fan F, Lai S, Huang C, Wang J, Yong J. Progressive interpolation using Loop subdivision surfaces. In: Proceedings of the 5th international conference on advances in geometric modeling and processing. Springer; 2008. p. 526–33.

[22] Chen Z, Luo X, Tan L, Ye B, Chen J. Progressive interpolation based on catmull-clark subdivision surfaces. Comput Graph Forum 2008;27(7):1823–7.

[23] Delgado J, Pena JM. Progressive iterative approximation and bases with the fastest convergence rates. Comput Aided Geom Design 2007;24(1):10–8.

[24] Delgado J, Manuel PJ. A comparison of different progressive iteration approximation methods. In: Proceedings of 7th international conference on mathematical methods for curves and surfaces. Springer; 2008. p. 136–52.

[25] Lu L. Weighted progressive iteration approximation and convergence analysis. Comput Aided Geom Design 2010;27(2):129–37.

[26] Deng C, Ma W. Weighted progressive interpolation of Loop subdivision surfaces. Comput Aided Des 2012;44(5):424–31.

[27] Kineri Y, Wang M, Lin H, Maekawa T. B-spline surface fitting by iterative geometric interpolation/approximation algorithms. Comput Aided Des 2012;44(7):697–708.

[28] Deng S, Wang G. Numerical analysis of the progressive iterative approximation method. J Comput Aided Des Comput Graph 2012;24(7):879–84.

[29] Lin H. Local progressive-iterative approximation format for blending curves and patches. Comput Aided Geom Design 2010;27(4):322–39.

[30] Zhao Y, Lin H, Bao H. Local progressive interpolation for subdivision surface fitting. J Comput Res Dev 2012;49(8):1699–707.

[31] Lin H. Adaptive fitting by the progressive-iterative approximation. Comput Aided Geom Design 2012;29(7):463–73.

[32] Chen J, Wang G, Jin C. Two kinds of generalized progressive iterative approximations. Acta Automat Sinica 2012;38(1):135–9.

[33] Zhang L, Li Y, Yang Y. Generalized progressive iterative approximation for Said-Ball bases on triangular domains. J Image Graph 2014;19(2):275–82.

[34] Chen S. Progressive iterative algorithm for triangular Bézier surfaces. Comput Eng Appl 2014;50(19):152–5.

[35] Deng S, Guan S, Wang G, Deng S. An Extension of a New Kind of Graphics Fitting Method. Appl Math Inf Sci 2013;7(2):741–7.

[36] Lin H, Zhang Z. An extended iterative format for the progressive-iteration approximation. Comput Graph 2011;35(5):967–75.

[37] Lin H, Zhang Z. An efficient method for fitting large data sets using T-splines. SIAM J Sci Comput 2013;35(6):A3052–68.

[38] Deng C, Lin H. Progressive and iterative approximation for least squares B-spline curve and surface fitting. Comput Aided Des 2014;47:32–44.

[39] Ray BK, Pandyan R. ACORD–an adaptive corner detector for planar curves. Pattern Recognit 2003;36(3):703–8.

[40] Zhang L, Ge X, Tan J. Least square geometric iterative fitting method for generalized B-spline curves with two different kinds of weights. Vis Comput 2015;1–12.

[41] Douglas DH, Peucker TK. Algorithms for the reduction of the number of points required to represent a line or its caricature. Canad Cartographer 1973;10(2):112–22.

[42] Farin G. Curves and surfaces for computer aided geometric design: a practical guide. fifth ed. Morgan Kaufmann; 2002 (Chapter 22).

[43] Azariadis P. Parameterization of clouds of unorganized points using dynamic base surfaces. Comput Aided Des 2004;36(7):607–23.

[44] Nishiyama Y, Morioka M, Maekawa T. Loop subdivision surface fitting by geometric algorithms. In: Igarashi T, Max N, Sillion F, editors. Poster proceedings of Pacific graphics 2008. 2008. p. 67–74.

[45] Lin H, Cao Q, Zhang X. The convergence of least-squares progressive iterative approximation with singular iterative matrix, ArXiv e-prints1707.09109.

[46] Lu L, Hu Q, Wang G. An iterative algorithm for degree reduction of Bézier curves. J Comput Aided Des Comput Graph 2009;21(12):1689–93.

[47] Lu L. Sample-based polynomial approximation of rational Bézier curves. J Comput Appl Math 2011;235(6):1559–63.

[48] Hu Q. An iterative algorithm for polynomial of rational triangular Bézier surfaces. Appl Math Comput 2013;219(17):9308–16.

[49] Zhang L, Wang H, Li Y. A progressive iterative approximation method in offset approximation. J Comput Aided Des Comput Graph 2014;26(10):1646–53.

[50] Gofuku S-i, Tamura S, Maekawa T. Point-tangent/point-normal B-spline curve interpolation by geometric algorithms. Comput Aided Des 2009;41(6):412–22.

[51] Okaniwa S, Nasri A, Lin H, Abbas A, Kineri Y, Maekawa T. Uniform B-spline curve interpolation with prescribed tangent and curvature vectors. IEEE Trans Vis Comput Graphics 2012;18(9):1474–87.

[52] Xing R, Pan R. PIA for uniform cubic B-spline curve interpolation with prescribed tangent vector. J Fujian Normal Univ Natur Sci Ed 2014;30(1):25–32.

[53] Zhao Y, Lin H. Real-time interactive modification of B-spline by PIA. J Comput Aided Des Comput Graph 2011;23(12):2013–8.

[54] Kineri Y, Endo S, Maekawa T. Surface design based on direct curvature editing. Comput Aided Des 2014;55(3):1–12.

[55] Lin H, Zhao Y. Variational progressive-iterative approximation for fairing curve and surface generation. In: Proceedings of 12th international conference on computer-aided design and computer graphics. IEEE; 2011. p. 258–61.

[56] Lin H, Chen W, Wang G. Curve reconstruction based on an interval B-spline curve. Vis Comput 2005;21(6):418–27.

[57] Liu Y, Fu H, Ju L. Application research of iterative minor adjustment in reverse engineering. In: ICMA 2009. International conference on mechatronics and automation, 2009. IEEE; 2009. p. 622–6.

[58] Yoshihara H, Yoshii T, Shibutani T, Maekawa T. Topologically robust B-spline surface reconstruction from point clouds using level set methods and iterative geometric fitting algorithms. Comput Aided Geom Design 2012;29(7):422–34.

[59] Lin H, Liao H, Deng C. Filling triangular mesh model with all-hex mesh by volume subdivision fitting, Technique report, TR-ZJUCAD-002, State Key Lab. of CAD & CG, Zhejiang University, 2012.

[60] Lin H, Jin S, Liao H, Jian Q. Quality guaranteed all-hex mesh generation by a constrained volume iterative fitting algorithm. Comput Aided Des 2015;67:107–17.

[61] Jian Q, Lin H, Cao Q, Lu X. Generating Quality Guaranteed Quadrilateral Mesh on an n-sided Region. J Comput Aided Des Comput Graph 2016;28(11):1811–20.

[62] Martin T, Cohen E, Kirby RM. Volumetric parameterization and trivariate B-spline fitting using harmonic functions. Comput Aided Geom Design 2009;26(6):648–64.

[63] Lin H, Jin S, Hu Q, Liu Z. Constructing B-spline solids from tetrahedral meshes for isogeometric analysis. Comput Aided Geom Design 2015;35:109–20.

[64] Boonyasith K, Akara P, Preesan R. Rice phenology monitoring using PIA time series MODIS imagery. In: Proceedings of 10th International conference on computer graphics, imaging and visualization. IEEE; 2013. p. 84–7.

[65] Suchada S, Natasha D. An approach to Thai decorative pattern recognition using Bézier curve representation with progressive iterative approximation. In: Proceedings of 10th international conference on computer graphics, imaging and visualization. IEEE; 2013. p. 46–9.

[66] Taweechai N, Natasha D. Approximating handwritten curve by using progressive-iterative approximation. In: Proceedings of 10th international conference on computer graphics, imaging and visualization. IEEE; 2013. p. 33–7.

[67] Anchisa C, Natasha D. Conversion of rational Bézier curves into non-rational Bézier curves using progressive iterative approximation. In: Proceedings of 10th international conference on computer graphics, imaging and visualization. IEEE; 2013. p. 38–41.

[68] Pan Z, Sun S, Zhang M, Zhang D. Watermarking NURBS surfaces. In: Proceedings of 6th pacific-rim conference on multimedia information processing. Springer; 2005. p. 325–36.

[69] Kuzelka K, Marusak R. Comparison of selected splines for stem form modeling : A case study in Norway spruce. Ann For Res 2014;57(1):137–48.

[70] Yu Z, Song W, Qian J. On technology of parametric wing modeling based on CATIA. Aircraft Des 2010;30(3):27–30.

[71] Sui Y, Zhang Z, Zhao J. Fitting of component characteristics of turbine engine based on non-uniform rational B-splines. J Aerosp Power 2008;23(8):1486–9.

[72] Wang G. Research on minor adjustment of freeform curves& surfaces and its applications. Hangzhou, China: Zhejiang University; 2006 (Ph.D. thesis).

[73] Imai T, Shibutani T, Matsui K, Kumagai S, Tran DT, Mu K, Maekawa T. Curvature sensitive analysis of axially compressed cylindrical tubes with corrugated surface using isogeometric analysis and experiment. Comput Aided Geom Design 2016;49:17–30.

[74] Sasaki Y, Takezawa M, Kim S, Kawaharada H, Maekawa T. Adaptive direct slicing of volumetric attribute data represented by trivariate B-spline function. Int J Adv Manuf Technol 2017;91(5–8):1791–807.

[75] Hughes T, Cottrell J, Bazilevs Y. Isogeometric analysis: CAD, finite elements, NURBS, exact geometry and mesh refinement. Comput Methods Appl Mech Engrg 2005;194(39):4135–95.