Fairing-PIA: Progressive iterative approximation for fairing curve and surface generation

Yini Jiang\(^a\), Hongwei Lin\(^{a,b,*}\)

\(^a\)School of Mathematical Science, Zhejiang University, Hangzhou, 310027, China
\(^b\)State Key Laboratory of CAD\&CG, Zhejiang University, Hangzhou, 310058, China

Abstract

The fairing curves and surfaces are used extensively in geometric design, modeling, and industrial manufacturing. However, the majority of conventional fairing approaches, which lack sufficient parameters to improve fairness, are based on energy minimization problems. In this study, we develop a novel progressive-iterative approximation method for fairing curve and surface generation (fairing-PIA). Fairing-PIA is an iteration method that can generate a series of curves (surfaces) by adjusting the control points of B-spline curves (surfaces). In fairing-PIA, each control point is endowed with an individual weight. Thus, the fairing-PIA has many parameters to optimize the shapes of curves and surfaces. Not only a fairing curve (surface) can be generated globally through fairing-PIA, but also the curve (surface) can be improved locally. Moreover, we prove the convergence of the developed fairing-PIA and show that the conventional energy minimization fairing model is a special case of fairing-PIA. Finally, numerical examples indicate that the proposed method is effective and efficient.

Keywords: Curve and surface fairing, Progressive iterative approximation, Fairing-PIA, Energy minimization, Geometric iteration method

1. Introduction

The generation of fairing curves and surfaces from scattered data points is a fundamental problem in many fields. For example, in the area of geometric design, generating smooth airfoils [11], ship hulls forms [2], and car hoods [3, 4] is often necessary because these geometric properties can further affect their performance characteristics. Recently, many scholars have been interested in the applications of fairing curves in robot path planning [5, 6]; thus, designing effective and flexible fairing algorithms has become an urgent concern.

Traditionally, two kinds of fairing methods exist: global [7, 8] and local methods [9, 10, 11]. In both fairing methods, the fairing problem is usually modeled as an energy optimization problem or constrained energy optimization problem. However, the energy optimization model lacks sufficient parameters to adjust and improve the fairness of curves and surfaces finely. The capability of fine adjustment is a key ingredient in generating fairing curves and surfaces.

In this study, we develop a novel progressive-iterative approximation method for fairing curve and surface generation (fairing-PIA) to overcome the deficiency of the traditional fairing methods. A series of curves (surfaces) can be generated through fairing-PIA by updating the control points iteratively. In each iteration, we first construct the fitting vector for a control point and the fairing vector for a control point. Then, the weighted sum of the two vectors is taken as the difference vector for each control point. Finally, the new control points are generated by adding the difference vectors to the old ones. In this way, each control point is endowed with an individual weight. In total, considerable weights exist. These weights can be utilized to adjust and improve the fairness of curves and surfaces finely. Moreover, we prove the convergence of fairing-PIA. We also show that the traditional energy optimization fairing model is a special case of fairing-PIA. In conclusion, the following are the main contributions of this study:

- We develop the fairing-PIA for fairing curve and surface generation. In this method, many parameters can be utilized to improve the fairness finely.
- Fairing-PIA is both a global and local method; thus, it can generate a global fairing result and adjust the curves (surfaces) locally.

\(^*\)Corresponding author at: School of Mathematical Science, Zhejiang University, Hangzhou, 310027, China.

Email address: hwlin@zju.edu.cn (Hongwei Lin)
Fairing-PIA is a flexible method that allows users to change the weights for control points, the knot vectors, or data parametrization in each iteration.

In this paper, we present the fairing-PIA method with B-spline curves and surfaces. However, fairing-PIA is valid for other blending curves and surfaces, including Bezier curves and surfaces, NURBS curves and surface, T-spline surfaces.

The paper is organized as follows: In Section 1.1, we present a brief overview of the related work. Section 2 contains the details of our algorithm for curve fairing. Convergence is discussed in Section 3. Then, we propose a fairing algorithm for surfaces in Section 4. Moreover, numerical examples are presented in Section 5. Finally, this paper is concluded in Section 6.

1.1. Related work

In this section, we briefly review the related work on curve and surface fairing and progressive-iterative approximation (PIA).

Curve and surface fairing: The basic approaches to designing the fairing curve include poor point removal, energy minimization, and a combination of both methods. Kjellander et al. [9] proposed an interactive fairing algorithm to improve the smoothness of the curve by modifying the point where the offset vector is greater than the threshold value. Farin et al. [10] performed the knot removal and reinsertion tricks for the first time to increase the efficiency of replacing points. However, such a local fairing method is intrinsically inefficient when numerous abnormal points are encountered.

Energy minimization methods are proposed to overcome this limitation. The basic idea is to find the smoothest curve in the sense of minimal energy functional. The commonly used energy includes strain energy [11], tension energy [12], Holladay functional [13], jerk energy [14], and the variation of radius of curvature [15]. The energy methods can perform global optimization for all points; however, the calculation is time-consuming, and the fitting effect cannot be controlled well. Hence, another report focuses on integrating the aforementioned two approaches.

Vassilev [16] developed an autonomous fairing technique based on energy minimization and point insertion. Unlike the previous methods, the technique developed only adds new points when necessary. Thus, the number of control points to be solved is greatly decreased. Inspired by this work, researchers have presented numerous approaches [17, 18, 19] to identify bad points efficiently and update the control points derived from the energy minimization method.

Wavelet and multiresolution decomposition techniques have been recently employed to fair non-uniform rational B-splines (NURBS) [8, 20, 21]. For more details on curve and surface fairing, please refer to Ref. [22].

PIA: Inspired by Qi and de Boor’s work [23, 24], Lin et al. [25] initiated the study of PIA. Compared with classical methods of curve and surface fitting, PIA avoids the intrinsic lack of direct solving equations, such as high computational cost, by generating a series of fitting curves and surfaces after each iteration to approximate the curve and surface, which interpolate the given data points. Then, Lin et al. [26] proved that blending curves and surfaces with normalized totally positive bases achieve convergence. Shi et al. [27] popularized the result to NURBS curves and surfaces.

Additionally, several accelerating algorithms have been developed. Lu [28] proposed the weighted PIA, which can accelerate the convergence rate by introducing optimal weights for difference vectors. Wang [29] developed the Gauss-Seidel PIA to accelerate the traditional PIA method. Moreover, Deng and Lin [30] proposed the least square progressive iterative approximation (LSPIA) and analyzed its convergence. In that method, the number of control points is allowed to be less than that of the data points. This approach is particularly suitable for handling large data sets. Since then, numerous interesting applications have been found in diverse fields, including subdivision surfaces [31, 32, 33], implicit curve and surface reconstruction [34], and single image super-resolution [35]. For more details we refer the readers to [36].

2. Progressive-iterative approximate for curve fairing

In this section, we present the fairing-PIA method for B-spline curve fairing. Given an ordered data set \( \{Q_i\}_{i=1}^m \) with parameters \( \{t_i\}_{i=1}^m \) satisfying \( t_1 \leq t_2 \leq \cdots \leq t_m \), the initial B-spline curve can be constructed as follows:

\[
P^{[0]}(t) = \sum_{j=1}^{n} N_j(t) P_j^{[0]}, \quad t \in [t_1, t_m],
\]

where the initial control points \( \{P_j^{[0]}\}_{j=1}^n \) can be randomly selected, and \( N_j(t) \) is the B-spline basis function corresponding to the \( j \)-th control point. Suppose that the B-spline curve \( \bar{P}[1] \) is defined on the knot vector \( \{\bar{t}_1, \bar{t}_2, \cdots, \bar{t}_{n+p+1}\} \), where \( p \) is
the degree of the B-spline basis function, and \( n \) is the number of control points. In this case, we define the \textit{fairing functional} as follows to generate the fairing curve:

\[
\mathcal{F}_{r,j}(f) = \int_{t}^{t_n} N_{r,j}(t) f dt, \quad j = 1, 2, \ldots, n, \quad r = 1, 2, 3,
\]  

(2)

where \( N_{r,j}(t) \) represents the \( r \)-th derivative of \( N_j(t) \). Given the local support property of B-spline functions, the given data points \( \{q_i\}_{i=1}^{m} \) whose parameters fall in the support domain of the \( j \)-th basis function, i.e., \( N_j(t) \neq 0 \), are classified into the \( j \)-th group, corresponding to the \( j \)-th control point. The index set of the data points in the \( j \)-th group is denoted as \( I_j \).

First, we calculate the \textit{difference vectors for data points} in the first step of the iteration, i.e.,

\[
d_j^{[0]} = q_i - P_j^{[0]}(t_i), \quad i = 1, 2, \ldots, m.
\]

The \textit{fitting vectors for control points} are generated by taking a weighted sum of the difference vectors for the \( j \)-th group data points:

\[
d_j^{[0]} = \sum_{h \in I_j} N_j(t_h) d_h^{[0]}, \quad j = 1, 2, \ldots, n.
\]

Next, the \textit{fairing vectors for control points} of the first iteration can be calculated as

\[
\eta_j^{[0]} = \sum_{i=1}^{n} \mathcal{F}_{r,j}(N_j(t_i)) P_i^{[0]}, \quad j = 1, 2, \ldots, n.
\]

Then, the new control points \( \{P_j^{[1]}\}_{j=1}^{n} \) after the first iteration comprise the contribution of three terms:

\[
P_j^{[1]} = P_j^{[0]} + \mu_j \left[ (1 - \omega_j) d_j^{[0]} - \omega_j \eta_j^{[0]} \right],
\]

where \( \mu_j \) is a \textit{normalization weight}, and \( \omega_j \) is a \textit{smoothing weight} corresponding to the \( j \)-th control point. The larger the smoothing weight \( \omega_j \) is, the smoother the generated curve is. The new curve is obtained as follows by substituting the new control points into the curve (1):

\[
P_j^{[1]}(t) = \sum_{j=1}^{n} N_j(t) P_j^{[1]}, \quad t \in [t_1, t_m].
\]

This procedure is performed iteratively. After the \( k \)-th iteration, the \( k \)-th curve \( P_j^{[k]}(t) \) is generated:

\[
P_j^{[k]}(t) = \sum_{j=1}^{n} N_j(t) P_j^{[k]}, \quad t \in [t_1, t_m].
\]

To construct the \((k + 1)\)-st curve \( P_j^{[k+1]}(t) \), we first calculate the \((k + 1)\)-st difference vectors for data points,

\[
d_j^{[k]} = q_i - P_j^{[k]}(t_i), \quad i = 1, 2, \ldots, m.
\]

Then, the \textit{fitting vectors for control points} and the \textit{fairing vectors for control points} are calculated by

\[
d_j^{[k]} = \sum_{h \in I_j} N_j(t_h) d_h^{[k]}, \quad j = 1, 2, \ldots, n, \quad \text{and},
\]

\[
\eta_j^{[k]} = \sum_{i=1}^{n} \mathcal{F}_{r,j}(N_j(t_i)) P_i^{[k]}, \quad j = 1, 2, \ldots, n.
\]

Finally, the control points of the \((k + 1)\)-st curve are produced by

\[
P_j^{[k+1]} = P_j^{[k]} + \mu_j \left[ (1 - \omega_j) d_j^{[k]} - \omega_j \eta_j^{[k]} \right], \quad j = 1, 2, \ldots, n,
\]

(3)

leading to the \((k + 1)\)-st B-spline curve,

\[
P_j^{[k+1]}(t) = \sum_{j=1}^{n} N_j(t) P_j^{[k+1]}, \quad t \in [t_1, t_m].
\]
In this way, we obtain a sequence of curves \( \{P^{(k)}(t), \, k = 1, 2, 3, \cdots \} \). In the following section, we prove that the sequence of curves converges. In the fairing-PIA stated above (Eq. (3)), \( \mu_j \) is a normalization weight that guarantees the convergence of the algorithm, and \( \omega_j \) is a smoothing weight that balances the fitting vector and fairing vector. Fairing-PIA is much more flexible for adjusting the fairness than the traditional fairing methods because each control point has an individual smoothing weight.

3. Convergence analysis

In this section, we show that the fairing-PIA for curve fairing (3) is convergent, and the traditional energy minimization fairing model is a special case of fairing-PIA. It is rewritten in the following matrix form to show the convergence of the fairing-PIA method (3):

\[
P^{(k+1)} = P^{(k)} + \Lambda \left[ (I - \Omega) N^T (Q - NP^{(k)}) - \Omega D_r P^{(k)} \right]
\]

\[=(I - \Lambda \mu) P^{(k)} + \Lambda (I - \Omega) N^T Q,
\]

in which

\[
P^{(k)} = \left[ p_1^{(k)}, p_2^{(k)}, \cdots, p_n^{(k)} \right]^T, \quad Q^{(k)} = \left[ Q_1^{(k)}, Q_2^{(k)}, \cdots, Q_n^{(k)} \right]^T,
\]

\[
\Omega = \text{diag} (\omega_1, \omega_2, \cdots, \omega_n), \quad \Lambda = \text{diag} (\mu_1, \mu_2, \cdots, \mu_n),
\]

\[
A = (I - \Omega) N^T N + \Omega D_r.
\]

and \( N \) and \( D_r \) denote the collocation matrix and Gram matrix, respectively.

\[
N = \begin{bmatrix}
N_1(t_1) & N_2(t_1) & \cdots & N_n(t_1) \\
N_1(t_2) & N_2(t_2) & \cdots & N_n(t_2) \\
\vdots & \vdots & \ddots & \vdots \\
N_1(t_m) & N_2(t_m) & \cdots & N_n(t_m)
\end{bmatrix}
\]

\[
D_r = \begin{bmatrix}
\mathcal{F}_{r,1} (N_{r,1}(t_1)) & \mathcal{F}_{r,1} (N_{r,2}(t_1)) & \cdots & \mathcal{F}_{r,1} (N_{r,n}(t_1)) \\
\mathcal{F}_{r,2} (N_{r,1}(t_1)) & \mathcal{F}_{r,2} (N_{r,2}(t_1)) & \cdots & \mathcal{F}_{r,2} (N_{r,n}(t_1)) \\
\vdots & \vdots & \ddots & \vdots \\
\mathcal{F}_{r,n} (N_{r,1}(t_1)) & \mathcal{F}_{r,n} (N_{r,2}(t_1)) & \cdots & \mathcal{F}_{r,n} (N_{r,n}(t_1))
\end{bmatrix}
\]

Moreover, \( \text{diag}(\cdot) \) denotes the diagonal matrix, and \( r \) can be chosen from the set \( \{1, 2, 3\} \).

**Lemma 3.1.** Let \( A = [a_{ij}] \) be a strict diagonally dominant matrix. If the diagonal elements of matrix \( \Lambda \) satisfy \( 0 < \mu_i < \frac{2}{\sum_{j=1}^{n} |a_{ij}|}, \, i = 1, 2, \cdots, n \), then we obtain \( 0 < \|I - \Lambda A\|_\infty < 1 \).

**Proof.** Given that \( A \) is a strict diagonally dominant matrix with nonnegative diagonal entries, we obtain \( \sum_{j=1}^{n} |a_{ij}| < 2a_{ii} \). Consequently,

\[
0 < \frac{1}{a_{ii}} < \frac{2}{\sum_{j=1}^{n} |a_{ij}|}, \, i = 1, 2, \cdots, n.
\]

On the other hand, according to the definition of \( \infty \)-norm, we obtain

\[
\|I - \Lambda A\|_\infty = \max_i \left( 1 - \mu_i a_{ii} + \sum_{j=1, j\neq i}^{n} \mu_j |a_{ij}| \right).
\]

We discuss below the range of \( \|I - \Lambda A\|_\infty \) under three cases:

(i) If \( \mu_i = \frac{1}{a_{ii}}, \, i = 1, 2, \cdots, n \), then Eq. (7) can be simplified as

\[
\|I - \Lambda A\|_\infty = \max_i \left( \frac{\sum_{j=1, j\neq i}^{n} |a_{ij}|}{a_{ii}} \right) < 1.
\]
(ii) If \(0 < \mu_i < \frac{1}{a_{ii}}, \ i = 1, 2, \ldots, n\), then we obtain the following by substituting Eq. (6) into Eq. (7):

\[
\|I - \Lambda A\|_\infty = \max_i \left(1 - \mu_i a_{ii} + \sum_{j=1, j \neq i}^{n} \mu_i |a_{ij}|\right) = \max_i \left(1 + \mu_i \sum_{j=1}^{n} |a_{ij}| - 2 a_{ii}\right) < 1.
\]

(iii) If \(\frac{1}{a_{ii}} < \mu_i < \frac{2}{\sum_{j=1}^{n} |a_{ij}|}, \ i = 1, 2, \ldots, n\), we can compute Eq. (7) in the following way:

\[
\|I - \Lambda A\|_\infty = \max_i \left(\mu_i a_{ii} - 1 + \sum_{j=1, j \neq i}^{n} \mu_i |a_{ij}|\right) = \max_i \left(\mu_i \sum_{j=1}^{n} |a_{ij}| - 1\right) < 1.
\]

When \(0 < \mu_i < \frac{2}{\sum_{j=1}^{n} |a_{ij}|}, \ i = 1, 2, \ldots, n\), we obtain \(0 < \|I - \Lambda A\|_\infty < 1\) by combining the results of case (i)-(iii).

\[\text{Theorem 3.2. When } A \text{ is a strict diagonally dominant matrix, fairing-PIA} \quad (4) \text{ is convergent.}\]

\[\text{Proof. We obtain the iterative method from Eq. (4) as follows:}\]

\[
P^{(k+1)} = P^{(k)} + \Lambda \left[(I - \Omega) N^T (Q - N P^{(k)}) - \Omega D_r P^{(k)}\right]
\]

\[
= (I - \Lambda A) P^{(k)} + \Lambda (I - \Omega) N^T Q
\]

\[
= (I - \Lambda A)^2 P^{(k-1)} + \Lambda (I - \Omega) N^T Q + (I - \Lambda A) \Lambda (I - \Omega) N^T Q
\]

\[
\vdots
\]

\[
= (I - \Lambda A)^{k+1} P^{(0)} + \sum_{l=0}^{k} (I - \Lambda A)^l \Lambda (I - \Omega) N^T Q. 
\]

Based on Lemma 3.1, the spectral radius of \(I - \Lambda A\) satisfies

\[
0 < \rho(I - \Lambda A) < \|I - \Lambda A\|_\infty < 1,
\]

and we obtain

\[
\lim_{k \to \infty} (I - \Lambda A)^k = 0, \quad \sum_{l=0}^{\infty} (I - \Lambda A)^l = (\Lambda A)^{-1}.
\]

Hence, when \(k \to \infty\), Eq. (8) tends to

\[
P^{(\infty)} = (\Lambda A)^{-1} \Lambda (I - \Omega) N^T Q
\]

\[
= A^{-1} (I - \Omega) N^T Q,
\]

which is the solution of the linear system as follows,

\[
AP^{(\infty)} = (I - \Omega) N^T Q.
\]

\[\text{Remark 3.3. In practice, we select the diagonal elements of matrix } \Lambda \text{ as } \mu_i = \frac{1}{\sum_{j=1}^{n} |a_{ij}|}, \ i = 1, 2, \cdot \cdot \cdot, n.\]
In a special case of $\Omega = \omega I$, i.e., all smoothing weights $\omega_j$, $j = 1, 2, \cdots, n$ are equal to $\omega$, the fairing-PIA degenerates to

$$
P^{[k+1]} = P^{[k]} + \Delta \left[ (1 - \omega) N^T \left( Q - NP^{[k]} \right) - \omega D_r P^{[k]} \right]$$

where

$$B = (1 - \omega) N^T N + \omega D_r.$$  \hfill (10)

The matrices $N^T N$ and $D_r$ are real symmetric. The matrix $B$ is also a real symmetric matrix. In the following, we will prove that the special case ($9$) of fairing-PIA is equivalent to the traditional energy minimization fairing model.

In the traditional energy minimization fairing method, the goal is to find the minimum of the energy functional, which is defined as follows:

$$E = \frac{(1 - \omega)}{2} f_1 + \frac{\omega}{2} f_2,$$  \hfill (11)

where $f_1$ and $f_2$ represent the fitting term and the fairing term, respectively, defined as

$$f_1 = \sum_{i=1}^{m} \| P(t_i) - Q_i \|^2,$$  \hfill (12)

$$f_2 = \int_{t_1}^{t_m} \| P^{(r)}(t) \|^2 dt, \quad r = 1, 2, 3,$$  \hfill (13)

where $P^{(r)}(t)$ represents the $r$-th derivative of $P(t)$, and $\omega$ is the smoothing weight. The larger the smoothing weight $\omega$ is, the smoother the generated curve is.

**Remark 3.4.** When $r = 1, 2, 3$, the fairing term $f_2$ corresponds to three types of energy: stretch energy, strain energy, and jerk energy $[37, 38, 39]$.

The substitution of Eq. (13) into Eq. (11) makes the optimization problem become

$$\min_{P_j} E = \min_{P_j} \left[ \frac{(1 - \omega)}{2} \sum_{i=1}^{m} \| P(t_i) - Q_i \|^2 + \frac{\omega}{2} \int_{t_1}^{t_m} \| P^{(r)}(t) \|^2 dt \right].$$  \hfill (14)

We obtain the following by $\frac{\partial E}{\partial P_j} = 0$, with $j = 1, 2, \cdots, n$:

$$\frac{\partial E}{\partial P_j} = (1 - \omega) \sum_{i=1}^{m} N_j(t_i) \left( \sum_{i=1}^{m} N_j(t_i) P_j - Q_i \right)$$

$$+ \omega \int_{t_1}^{t_m} \left( \sum_{i=1}^{m} N_j(t) P_j \right) dt = 0,$$  \hfill (15)

where $\omega$ is a smoothing parameter.

Rearranging Eq. (15) yields

$$\sum_{i=1}^{m} (1 - \omega) N_j(t_i) Q_j = \sum_{i=1}^{m} (1 - \omega) N_j(t_i) P_j$$

$$+ \omega \int_{t_1}^{t_m} \left( \sum_{i=1}^{m} N_j(t) P_j \right) dt.$$  \hfill (16)

Then, Eq. (16) can be represented by a matrix as

$$BP = (1 - \omega) N^T Q,$$  \hfill (17)

where $B = (1 - \omega) N^T N + \omega D_r$. A fairing curve that minimizes the energy model (11) can be obtained after the equation is solved and the obtained control points are substituted into the curve equation.
Theorem 3.5. Let the diagonal elements of matrix $\Lambda$ satisfy $0 < \mu_i < \frac{2}{\lambda_{\text{min}}(B)}$, $i = 1, 2, \cdots, n$, where $\lambda_{\text{max}}$ is the largest eigenvalue of matrix $B$. If $B$ is a positive definite matrix, then the special case of the fairing-PIA (6) converges to the solution of the traditional energy minimization fairing model (14).

Proof. According to Eq. (9), we have

$$ P^{[k+1]} = (I - \Lambda B) P^{[k]} + (1 - \omega) A R^T Q $$

$$ = (I - \Lambda B)^{k+1} P^{[0]} + (1 - \omega) \sum_{l=0}^{k} (I - \Lambda B)^l A R^T Q, $$

where $B$ is a positive definite matrix.

A real orthogonal matrix $U$ and a positive diagonal matrix $S$ exist, such that $B = USU^T = US^2 S^2 U^T$. We obtain $B = C^T C$ by denoting $C = S^{1/2} U^T$. Suppose that $\beta$ is an arbitrary eigenvalue of the matrix $\Lambda B = \Lambda C^T C$ w.r.t. eigenvector $v$, i.e.,

$$ \Lambda B v = \Lambda C^T C v = \beta v. $$

In this case, we obtain the following by multiplying both sides of Eq. (19) by $C$:

$$ C \Lambda C^T (C v) = \beta (C v), $$

which means that $\beta$ is also an eigenvalue of the matrix $C \Lambda C^T$ w.r.t. eigenvector $C v$. Moreover, for all nonzero vectors $x \in R^n$, it holds

$$ x^T C \Lambda C^T x = (x^T C \Lambda \frac{1}{\sqrt{2}})(x^T C \Lambda \frac{1}{\sqrt{2}})^T > 0. $$

Therefore, we conclude that the matrix $C \Lambda C^T$ is a positive definite matrix, and all its eigenvalues are positive, i.e., $\beta > 0$.

Thus, the eigenvalues of matrix $\Lambda B = \Lambda C^T C$ are all positive real numbers.

On the other hand, given that $0 < \mu_i < \frac{2}{\lambda_{\text{min}}(B)}$, $i = 1, 2, \cdots, n$, the eigenvalues of $\Lambda B$ satisfy

$$ 0 < \lambda (\Lambda B) < \|\Lambda B\|_2 \leq \|A\|_2 \|B\|_2 = \lambda_{\text{max}}(A) \lambda_{\text{max}}(B) < 2. $$

Thus, the eigenvalues of $(I - \Lambda B)$ fulfill the condition $-1 < \lambda (I - \Lambda B) < 1$, leading to

$$ \rho(I - \Lambda B) < 1. $$

We obtain the following from Eq. (20):

$$ \lim_{k \to \infty} (I - \Lambda B)^k = 0, \quad \sum_{l=0}^{\infty} (I - \Lambda B)^l = (\Lambda B)^{-1}. $$

Hence, when $k \to \infty$, Eq. (18) tends to

$$ P^{[\infty]} = (1 - \omega) (\Lambda B)^{-1} A N^T Q $$

$$ = (1 - \omega) B^{-1} N^T Q. $$

The combination of Eqs. (21) and (17) indicate that the fairing-PIA (6) is convergent to the solution of the traditional energy minimization fairing model (14). $\square$

We deal with the case that matrix $B$ is positive definite in the above Theorem 3.5. When matrix $B$ is positive semidefinite, the convergence proof is complicated and deferred to Appendix for clarity of presentation.

Remark 3.6. For the case when matrix $\Omega$ (3) degrades to $\omega I$ (9), calculating the largest eigenvalue of the coefficient matrix $B$ is time-consuming. Therefore, we choose $\mu_i = \mu = \frac{2}{\lambda_{\text{max}}(B)}$, $i = 1, 2, \cdots, n$, which satisfies the condition in Theorem 3.5.
4. Progressive-iterative approximate for surface fairing

The fairing-PIA method for curve fairing can be easily extended to the surface case. In the following, we present the details of fairing-PIA for surface fairing.

Given an ordered data set \( \{Q_i\}_{i=1}^{m_i} \) with their parameters \( (s_i, t_i) \) satisfying \( s_1 \leq s_2 \leq \cdots \leq s_{m_i} \) and \( t_1 \leq t_2 \leq \cdots \leq t_{m_i} \), we rearrange the data points and their parameters in lexicographical order, i.e., \( \{Q_i\}_{i=1}^{m_i} \) and \( \{(s_i, t_i)\}_{i=1}^{m_i} \), where \( m = m_1 m_2 \). The initial B-spline surface \( P[0] \) can be constructed as follows:

\[
P[0](s, t) = \sum_{i=1}^{m} N_i(s)N_i(t)P[i], \quad 0 \leq s, t \leq 1,
\]

where \( \{P[i]\}_{i=1}^{m_i} \) are the initial control points, and \( N_i(s) \) and \( N_i(t) \) are the B-spline basis. Similarly, the basis functions and control points are arranged in lexicographical order as follows:

\[
\begin{align*}
N_1(s, t), N_2(s, t), \ldots, N_{m_2}(s, t) &= [N_1(s)N_1(t), N_1(s)N_2(t), \ldots, N_{m_1}(s)N_{m_2}(t)], \\
\{P_1, P_2, \ldots, P_{m_1m_2}\} &= [P_{11}, P_{12}, \ldots, P_{m_1m_2}].
\end{align*}
\]

Then, the initial surface can be rewritten as

\[
P[0](s, t) = \sum_{i=1}^{m_1m_2} N_i(s, t)P[i].
\] (22)

We define the fairing functionals analogous to Eq. (2). Let

\[
\begin{align*}
\mathcal{F}_{st,j}(f) &= \int_{s_1}^{s_n} \int_{t_1}^{t_n} N_{st,j}(s, t) f ds dt, \\
\mathcal{F}_{st,j}(f) &= \int_{s_1}^{s_n} \int_{t_1}^{t_n} N_{st,j}(s, t) f ds dt, \quad j = 1, 2, \ldots, n_1 n_2, \\
\mathcal{F}_{st,j}(f) &= \int_{s_1}^{s_n} \int_{t_1}^{t_n} N_{st,j}(s, t) f ds dt,
\end{align*}
\]

where \( N_{st,j}(s, t) \) represent the second-order partial derivatives of \( N_j(s, t) \) with respect to \( s \) and \( t \). Here, we take the fairing energy functional defined using the second-order derivatives as examples. The definition of the fairing functional can be easily extended to other cases [40].

Similar to the curve case, the \( k \)-th surface \( P[k] \) is assumed to be generated after the \( k \)-th iteration. In this case, we first calculate the \( (k + 1) \)-st difference vectors for data points as follows to construct the \( (k + 1) \)-st surface \( P[k+1] \) :

\[
d^{[k]} = Q_i - P^{[k]}(s_i, t_i), \quad i = 1, 2, \ldots, m.
\]

Then, the fitting vectors for control points and the fairing vectors for control points are calculated by

\[
\begin{align*}
\delta_j^{[k]} &= \sum_{i \in I_j} N_j(s_i, t_i) d_i^{[k]}, \quad j = 1, 2, \ldots, n_1 n_2, \quad \text{and}, \\
\mu_j^{[k]} &= \sum_{i \in I_j} [\mathcal{F}_{st,j}(N_{st,j}(s, t)) + 2\mathcal{F}_{st,j}(N_{st,j}(s, t))] P_i^{[k]},
\end{align*}
\]

where \( I_j \) is the index set of all parameters that fall in the support region of the \( j \)-th basis function. Finally, the new control points of the \( (k + 1) \)-st surface are produced by

\[
P_{j}^{[k+1]} = P_j^{[k]} + \mu_j^{[k]} \left[ (1 - \omega_j) \delta_j^{[k]} - \omega_j \mu_j^{[k]} \right],
\] (23)

leading to the \( (k + 1) \)-st B-spline surface:

\[
P^{[k+1]}(s, t) = \sum_{j=1}^{n_2} N_j(s, t)P_j^{[k+1]}.
\]

Here, the parameters \( \omega_j \) and \( \mu_j \) [23] are similar to those in Section 2.

In this way, we obtain a sequence of surfaces \( \{P[k] \} \), \( k = 1, 2, 3, \ldots \). The convergence analysis of the fairing-PIA for the surface case [23] is analogous to that of the curve case in Section 3.
5. Experiments and Discussions

In this section, some numerical examples are presented to demonstrate the effectiveness and efficiency of the developed fairing-PIA method. Moreover, we compare the fairing-PIA method with the traditional energy minimization method (11), which is a special case of the fairing-PIA method, i.e., the fairing-PIA method with equal weights.

In the numerical examples, cubic B-spline curves are considered the fitting curves. We choose the initial control points \( \{ P_{j}^{(0)} \}_{j=1}^{n} \) from the data points, i.e.,

\[
P_{j}^{(0)} = Q_{i}, \quad 1 = i_{1} < i_{2} < \cdots < i_{n} = m.
\]

Then, we construct a knot vector by using the following form:

\[
\{ 0, 0, 0, \tilde{t}_{5}, \cdots, \tilde{t}_{n}, 1, 1, 1, 1, \}
\]

where \( \tilde{t}_{j} = (t_{i_{j-1}} + t_{i_{j+1}} + t_{i_{j+2}}) / 3, \quad j = 5, \cdots, n. \) Moreover, bi-cubic B-spline surfaces are considered the fitting surfaces, and the initial control points and knot vectors are formed similarly.

On the one hand, we employ the root mean square as the *absolute fitting error*,

\[
E_{fit,a}^{[k]} = \sqrt{\frac{\sum_{i=1}^{m} (Q_{i} - P_{j}^{[k]}(t_{i}))^{2}}{m}}.
\]
and Eq. (13) as the absolute energy,

$$E_{eng,a} = \int_{t_i}^{t_f} \| P^r(t) \|^2 \, dt, \quad r = 1, 2, 3,$$

to measure the fitting error and the energy of the curve and surface. On the other hand, we define the relative fitting error $E_{fit,a}^{[k]}$, relative energy $E_{eng,a}^{[k]}$, i.e.,

$$E_{fit,a}^{[k]} = E_{fit,a}^{[0]} - E_{fit,a}^{[k]}, \quad E_{eng,a}^{[k]} = E_{eng,a}^{[0]} - E_{eng,a}^{[k]},$$

and the relative iteration error,

$$E_{iter,a}^{[k]} = \sqrt{\sum_{j=1}^{n} \| (I - \Omega^T N^T Q(j,:) - A^T P(j,:)) \|^2}, \quad k = 0, 1, 2, \ldots,$$

where matrices $\Omega$, $N$ and $A$ are given in Eq. (4) to measure the iteration procedure of the fairing-PIA. In our implementation, the fairing-PIA iteration stops when $|E_{iter,a}^{[k+1]} - E_{iter,a}^{[k]}| < 10^{-6}$. We implement the algorithms in MATLAB and run them on a PC with an Intel Core i7-10700 2.90GHz CPU and 16 GB RAM.

### 5.1. Planar curves

In this section, three examples of planar curve fairing are presented to illustrate the effectiveness of the fairing-PIA algorithm. The three models are as follows:

- **Note**: 245 points sampled from a note-shaped curve
- **Airfoil**: 49 points measured from a supercritical airfoil
- **Starfish**: 100 points sampled uniformly from an analytical curve expressed by

\[
\begin{align*}
x(t) &= \left(1 + \frac{1}{5} \cos(5t)\right) \cos(t), \quad t \in [0, 2\pi] \\
y(t) &= \left(1 + \frac{1}{5} \cos(5t)\right) \sin(t).
\end{align*}
\]

In the three examples, we take $r = 2$ in the fairing functional (2), corresponding to the strain energy in the traditional method (13).

In the example of the **Note** model, we first generate the fitting curve with the curvature comb (that is, the smoothing weight $\omega = 0$) is set in the energy functional (11), illustrated in Fig. 1(a). Then, the fairing curve using the traditional method is produced by setting $\omega = 1 \times 10^{-2}$ and $r = 2$ (strain energy) in the energy functional (11) (refer to Fig. 1(b)). The fairness of the curve segment in the red box (refer to Figs. 1(b) and 1(c)) is undesirable and needs to be improved. Each control point can be assigned with an individual smoothing weight in the fairing-PIA. Thus, we set the smoothing weights of the other control points as still set as $1 \times 10^{-5}$, thereby affecting the curve segment with undesirable fairness. The smoothing weights of the control points are still set as $1 \times 10^{-7}$, and the fairing curve is generated using fairing-PIA. As illustrated in Figs. 1(c) and 1(d), the fairness of the curve segment in the red box is clearly improved with different smoothing weights. Moreover, the diagrams of iterations v.s. relative iterative error, iterations v.s. relative fitting error, and iterations v.s. relative strain energy are shown in Fig. 1(e). The relative iterative error, relative fitting error, and relative strain energy are rapidly reduced at the early iterations.

In practice, the fairness of a curve is usually adjusted progressively and locally, i.e., segment by segment, using local fairing methods. In the example of the **Airfoil** model, we demonstrate the segment-by-segment fairing and local fairing capabilities of the fairing-PIA method. The fitting curve with curvature comb is illustrated in Fig. 2(a) where the smoothing weights for all control points are set as 0. We aim to improve the fairness of the three segments in the red boxes segment by segment. In Fig. 2(b), we first improve the fairness of the curve segment in the lower right box by setting the smoothing weights of the $4^{th}$ to $7^{th}$ control points as $1 \times 10^{-5}$. The fairness-PIA is performed by taking the control points of the fitting curve in Fig. 2(a) as the initial control points. The fairness of the curve segment in the lower right box is improved after the fairing-PIA iteration stops (Fig. 2(b)). Second, Fig. 2(c) illustrates that the fairness of the curve segment in the left box is improved using fairing-PIA by setting the smoothing weights of the $8^{th}$ to $11^{th}$ control points as $1 \times 10^{-5}$ and taking the control points of curve in Fig. 2(b) as the initial control points. Finally, the fairness of the curve segment in the upper right box can be improved similarly using fairing-PIA by setting the smoothing weights.
Figure 2: The Airfoil example. (a) Data sets, the fitting curve with its curvature comb, and its control polygon. (b) The first local fairing by adjusting the smoothing weights of the 4th-7th control points; (c) The second local fairing by adjusting the smoothing weights of the 8th-11th control points; (d) The third local fairing by adjusting the smoothing weights of the 19th-23rd control points.

Moreover, the flexibility of fairing-PIA is demonstrated in the example of the Starfish model (Fig. 3), where the new knots are inserted, and the smoothing weights are renewed in the iterations of fairing-PIA. In Fig. 3(a), the fitting curve of the 19th to 23rd as $1 \times 10^{-5}$ and taking the control points of the curve in Fig. 2(d) as the initial control points (refer to Fig. 2(d)).

Figure 3: The Starfish example. (a) The fitting curve and its curvature comb; (b) The fairing curve generated by fairing-PIA method and its curvature comb, where the fairness of the curve segments in the red boxes is improved, but the fitting errors are increased; (c) The fairing curve generated by fairing-PIA method by inserting knots and decreasing corresponding smoothing weights, where the fitting errors of the curve segments in the red boxes are reduced.

of the 19th to 23rd as $1 \times 10^{-5}$ and taking the control points of the curve in Fig. 2(d) as the initial control points (refer to Fig. 2(d)).

Moreover, the flexibility of fairing-PIA is demonstrated in the example of the Starfish model (Fig. 3), where the new knots are inserted, and the smoothing weights are renewed in the iterations of fairing-PIA. In Fig. 3(a), the fitting curve
with the curvature comb is presented, where the fairness of the curve segments in the red boxes is undesirable and needs to be improved. The fairness of the curve segments in the red boxes is improved (Fig. 3(b)) with fairing-PIA by setting the smoothing weights of the control points as $3 \times 10^{-5}$, thereby affecting the curve segments in the red boxes. The smoothing weights of the other control points are set as $1 \times 10^{-5}$. However, the fitting errors of the curve segments in the red boxes increase. We insert two knots at each curve segment in the red box to decrease the fitting error. We also reduce the smoothing weights of the control points to $6 \times 10^{-6}$, thereby affecting the curve segments in the red boxes. Then, a new round of fairing-PIA iterations is invoked by taking the curve generated after the last round of fairing-PIA iterations as the initial curve (refer to Fig. 3(b)). After the new round of iterations stops, a new curve is produced (Fig. 3(c)), where the fitting errors of the curve segments in the red boxes are reduced. Moreover, the fairness of these curve segments is comparable with the fairness of the curve in Fig. 3(b). The diagrams of iteration v.s. relative iterative error, iteration v.s. relative fitting error, and iteration v.s. the relative strain energy of the two rounds of iterations are shown in Fig. 4(b) where the fitting error is progressively decreased in the second round of iterations.

Additionally, the statistics of the above three examples are listed in Table 1, including the number of data points, number of control points, number of iterations, and running time. The absolute fitting error and absolute strain energy are given in Table 2.

![Figure 4](image-url)  
(a) The diagrams of iterations v.s. relative fitting error, relative iterative error and relative strain energy, respectively, for Note model (a), and Starfish model (b).

| Model               | # Data points a | # Control points b | # Iterations c | Time(seconds) | Energy minimization method | Fairing-PIA |
|---------------------|-----------------|-------------------|---------------|---------------|---------------------------|-------------|
| Note                | 245             | 83                | 64            | 0.1509        | 0.1035                    |             |
| Airfoil (The 1st segment) | 49              | 25                | 32            | 0.01144       | 0.01032                   |             |
| Starfish (The first round) | 100             | 34                | 23            | 0.02133       | 0.01911                   |             |
| Tooth               | 81 × 61         | 32 × 24           | 248           | 39.3429       | 35.4474                   |             |
| Fan_disk            | 41 × 61         | 16 × 24           | 85            | 7.4511        | 7.2305                    |             |
| Mannequin           | 121 × 161       | 48 × 64           | 143           | 1554.9496     | 1560.6648                 |             |

| a Number of the data points.  
| b Number of the control points.  
| c Number of iterations.  

5.2. Space Curves

In this section, an example of a space curve is presented to show the effects of the parameter $r$ in the fairing functional (2). As stated in Remark 3.4, different values of $r$, i.e., $r = 1, 2, 3$, correspond to different energies, i.e., stretch...
Table 2: The absolute fitting error and absolute energy of curve and surface examples.

| Model                  | Initial fitting error | Final fitting error$^a$ | Initial energy  | Final energy$^b$  |
|------------------------|-----------------------|--------------------------|-----------------|-------------------|
| Note                   | 0.01074               | 0.005778                 | 90438.9606      | 80448.1437        |
| Starfish (The first round) | 0.02725               | 0.02456                  | 15090.0488      | 13275.4486        |
| Viviani ($r = 1$)      | 0.02895               | 0.02210                  | 1455.8978       | 1442.3715         |
| Viviani ($r = 2$)      | 0.02895               | 0.01498                  | 2.4161 × 10$^5$ | 1.9570 × 10$^5$  |
| Viviani ($r = 3$)      | 0.02895               | 0.01712                  | 7.7733 × 10$^8$ | 3.8296 × 10$^7$  |
| Tooth                  | 0.006903              | 0.006180                 | 236.9693        | 70.3654           |
| Fan_disk               | 0.01002               | 0.005743                 | 73.3475         | 15.9637           |
| Mannequin              | 0.003143              | 0.001457                 | 787.0678        | 350.0640          |

$^a$ The fitting error obtained after the iteration stops.

$^b$ The energy obtained after the iteration stops.

$^c$ The stretch energy, strain energy and jerk energy correspond to $r = 1, 2, 3$, respectively.

Figure 5: The Viviani example. Comparison of different selections of $r$ (refer to Eq. (2)) in the fairing-PIA method. The fairing curve is in red, and the curvature comb is in green.

energy, strain energy and jerk energy. In the space curve example (Fig. 5), the 420 data points are sampled uniformly from the Viviani curve,

\[
\begin{align*}
\begin{cases}
    x^2 + y^2 + z^2 = 25, \\
    x^2 + y^2 - 5x = 0.
\end{cases}
\end{align*}
\]

Moreover, we add Gaussian noise with 0 mean and 0.005 variance into the data set. Then, we select 85 control points from the data points as the initial control points to perform the fairing-PIA iterations with $r = 1, 2, 3$. The smoothing
weights for the control points are assigned according to the following strategy. First, we calculate the absolute value of the second order difference (ASOD) of each data point. The initial control points are chosen from the data points. Thus, each control point has its own ASOD. The smoothing weights of the control points with the 20 highest ASODs are set as $2 \times 10^{-4}$, and the rest are set as $1 \times 10^{-5}$ for $r = 2$ and 3. When $r = 1$, the smoothing weights of the control points with the 20 highest ASODs are set as 0.022, and the rest are set as 0.02.

The fairing-PIA iteration procedures are illustrated in Fig. 5 for $r = 1, 2, 3.$ As shown in Fig. 5, the fairness of curves is improved with iterations, and the fairness for $r = 2$ and 3 is better than that for $r = 1.$ In addition, the diagrams of iteration v.s. relative iterative error, relative fitting error, and relative energy are presented in Fig. 6. Similar to those in the planar curve case, the iterative error, fitting error, and energy rapidly decrease in the early iterations. Moreover, the absolute fitting error and three types of absolute energies are shown in Table 2.

5.3. Surface examples

In this section, we compare the fairing-PIA method with the traditional energy minimization method (11) on three models, i.e., tooth, fan disk, mannequin (Fig. 7). The three data models consist of 81 $\times$ 61, 41 $\times$ 61 and 121 $\times$ 161 data points. The statistics of the fairing-PIA on the three surface examples are listed in Table 1.

For each model, we generate the fitting surface ($\omega = 0$ in Eq. (11)), the fairing surface by the traditional energy minimization method, and the fairing surface by the fairing-PIA developed in this study. The zebra and absolute value of mean curvature (AVMC) are rendered on the surfaces to evaluate the fairness of the surfaces. In each example, the AVMCs larger than the maximum AVMC of the fairing surface by the fairing-PIA method are mapped to the maximum AVMC of the fairing surface by fairing-PIA to generate a desirable mean curvature distribution image. The minimum and maximum mean curvatures of the surfaces are listed in Table 3. The absolute fitting error and absolute energy are given in Table 2.

For the tooth model, the smoothing weight $\omega$ (13) is taken as 0.002 (Figs. 8(b) and 8(e)) in the traditional energy minimization method. Some smoothing weights are increased to 0.006 in the fairing-PIA to improve the fairness of some
Figure 8: The tooth model. (a, b, c) Zebra on the fitting surface, fairing surface by the traditional energy minimization method, and fairing surface by fairing-PIA, respectively. (d, e, f) Absolute value of mean curvature on the fitting surface, fairing surface by the energy minimization method, and fairing surface by fairing-PIA, respectively.

Figure 9: The fan disk model. (a, b, c) Zebra on the fitting surface, fairing surface by the traditional energy minimization method, and fairing surface by fairing-PIA, respectively. (d, e, f) Absolute value of mean curvature on the fitting surface, fairing surface by the energy minimization method, and fairing surface by fairing-PIA, respectively.
regions, whereas the other smoothing weights remain 0.002 (Figs. [8](c) and [8](f)). For the fan disk model, the smoothing weight in the traditional energy minimization method is taken as $\omega = 0.001$ (Figs. [9](b) and [9](e)). Similarly, the corresponding smoothing weights are increased to 0.015 to improve the fairness of some regions (Figs. [9](c) and [9](f)).

For the mannequin model, the smoothing weight in the traditional energy minimization method (13) is set as $\omega = 1 \times 10^{-4}$. The generated fairing surface is illustrated in Figs. [10](b) and [10](e). To improve the fairness of the ear region, we increase the smoothing weights of the control points at the ear region to $5 \times 10^{-4}$, to keep the feature of the mouth region by reducing the fitting error, we decrease those at the mouth region to $2 \times 10^{-5}$. With the modified smoothing weights, the fairing surface is produced by fairing-PIA (Figs. [10](c) and [10](f)). The fairness of the ear region is improved, and the feature of the mouth region becomes clear.

Table 3: Minimum and maximum mean curvature value of surfaces.

| Type                 | Tooth   | Fan disk | Mannequin   |
|----------------------|---------|----------|-------------|
|                      | MinCur$^a$ | MaxCur$^a$ | MinCur | MaxCur | MinCur | MaxCur | MinCur | MaxCur |
| Fitting surface      | -15.8553 | 1791.7996 | -33.7973 | 42.6118 | -23.3849 | 27.0701 |
| Fairing surface by energy minimization | -11.9428 | 19.2055 | -27.8114 | 31.155 | -20.3517 | 27.0489 |
| Fairing surface by fairing-PIA      | -9.7458 | 14.9174 | -15.3867 | 15.7318 | -20.3517 | 23.3825 |

$^a$ Minimum mean curvature value.

$^b$ Maximum mean curvature value.
6. Conclusion

In this study, we present the fairing-PIA, a novel progressive iterative approximation for fairing curves and surface generation. The convergence analysis of the fairing-PIA is presented. Fairing-PIA is both a global and local method, where many parameters that can finely improve fairness exist. Moreover, fairing-PIA is flexible. Thus, the smoothing weights and knot vector can be changed in the iterations. Finally, many examples are presented to demonstrate the efficiency and effectiveness of the developed fairing-PIA method.

Acknowledgements

This work is supported by the National Natural Science Foundation of China under Grant nos. 61872316, 61932018, and the National Key R&D Plan of China under Grant no. 2020YFB1708900.

References

[1] W. Li, S. Krist, Spline-based airfoil curvature smoothing and its applications, Journal of aircraft 42 (4) (2005) 1065–1074.
[2] E. Sariöz, An optimization approach for fairing of ship hull forms, Ocean Engineering 33 (16) (2006) 2105–2118.
[3] G. Westgaard, H. Nowacki, A process for surface fairing in irregular meshes, Computer Aided Geometric Design 18 (7) (2001) 619–638.
[4] A. Hashemian, S. F. Hosseini, An integrated fitting and fairing approach for object reconstruction using smooth nurbs curves and surfaces, Computers & mathematics with applications (1987) 76 (7) (2018) 1555–1575.
[5] R. F. Gmänyanov, Recursive method of smoothing curvature of path in path planning problems for wheeled robots, Automation and remote control 72 (7) (2011) 1548–1556.
[6] B. Song, Z. Wang, L. Zou, An improved pso algorithm for smooth path planning of mobile robots using continuous high-degree bézier curve, Applied soft computing 100 (2021).
[7] H. Hagen, G.-P. Bonneau, Variational design of smooth rational bézier curves, Computer aided geometric design 8 (5) (1991) 393–399.
[8] W. Wang, Y. Zhang, Wavelets-based nurbs simplification and fairing: Computational geometry and analysis, Computer methods in applied mechanics and engineering 199 (5-8) (2010) 290–300.
[9] J. A. P. Kjellander, Smoothing of cubic parametric splines, Computer aided design 15 (3) (1983) 175–179.
[10] G. Farin, G. Rein, N. Sapidis, A. Worsey, Fairing cubic b-spline curves, Computer Aided Geometric Design 4 (1) (1987) 91–103.
[11] A. Wang, G. Zhao, Y.-D. Li, Wavelet based local fairing algorithm for surfaces with data compression, International journal of wavelets, multiresolution and information processing 13 (5) (2015) 1550041.
[12] H. Pottmann, Smooth curves under tension, Computer aided design 22 (4) (1990) 241–245.
[13] E. S. Cambell, Smoothest curve approximation, Mathematics of computation 11 (60) (1957) 233–243.
[14] H. Meier, H. Nowacki, Interpolating curves with gradual changes in curvature, Computer aided geometric design 4 (4) (1987) 297–305.
[15] T. C. Rando, Automatic fairness in computer-aided geometric design, Ph.D. thesis, aA9102032 (1990).
[16] I. V. Vassilev, Fair interpolation and approximation of b-splines by energy minimization and points insertion, Computer aided design 28 (9) (1996) 753–760.
[17] C. Zhang, P. Zhang, F. Cheng, Fairing spline curves and surfaces by minimizing energy, Computer aided design 33 (13) (2001) 913–923.
[18] F. Caihö, E. Miglio, M. Rasella, Curve fairing using integral spline operators, International journal for numerical methods in biomedical engineering 26 (12) (2010) 1674–1686.
[19] W. Li, S. Xu, J. Zheng, G. Zhao, Target curvature driven fairing algorithm for planar cubic b-spline curves, Computer Aided Geometric Design 21 (5) (2004) 499–513.
[20] A. Li, G. Wei, H. Tian, F. Kou, Wavelet-based multiresolution nurbs curve fairing, Advanced Materials Research 314-316 (2011) 1562–1565.
[21] A. Li, H. Tian, A multiresolution fairing approach for nurbs curves, Applied Mechanics and Materials 215-216 (2012) 1205–1208.
[22] J. Hoschek, D. Lasser, L. L. Schumaker, Fundamentals of Computer Aided Geometric Design, A. K. Peters, Ltd., USA, 1993.
[23] D. Qi, Z. Tian, Y. Zhang, The method of numeric polish in curve fitting, Acta Math. Sin. 18 (1975) 173–184.
[24] C. de Boor, How does agee’s smoothing method work?, Proceedings of the 1979 Army Numerical Analysis and Computers Conference (01 1979).
[25] H. Lin, G. Wang, C. Dong, Constructing iterative non-uniform b-spline curve and surface to fit data points, Science in China Series F: Information Sciences 47 (2004) 315–331.
[26] H. Lin, H. Bao, G. Wang, Totally positive bases and progressive iteration approximation, Computers & Mathematics with Applications 50 (3) (2005) 575–586.
[27] L. Shi, R. Wang, An iterative algorithm of nurbs interpolation and approximation, Journal of Mathematical Research & Exposition 26 (01 2006).
[28] Weighted progressive iteration approximation and convergence analysis, Computer Aided Geometric Design 27 (2) (2010) 129–137.
[29] Z. Wang, Y. Li, C. Deng, Convergence proof of gs-pia algorithm, Journal of Computer-Aided Design & Computer Graphics 30 (2018) 2035.
[30] C. Deng, H. Lin, Progressive and iterative approximation for least squares b-spline curve and surface fitting, Computer-Aided Design 47 (2014) 32–44.
[31] Z. Chen, X. Luo, L. Tan, B. Ye, J. Chen, Progressive interpolation based on catmull clark subdivision surfaces, Comput. Graph. Forum 27 (2008) 1823–1827.
[32] C. Deng, W. Ma, Weighted progressive interpolation of loop subdivision surfaces, Computer-Aided Design 44 (2012) 424–431.
[33] Y. Hamza, H. Lin, Conjugate-gradient-iterative-approximation for loop and catmull-clark subdivision surface interpolation, Journal of Computer Science and Technology 37 (2) (2022) 487–502.
[34] Y. Hamza, H. Lin, Z. Li, Implicit progressive-iterative approximation for curve and surface reconstruction, Computer Aided Geometric Design 77 (2020) 101817.
[35] Y. Zhang, P. Wang, F. Bao, X. Yao, C. Zhang, H. Lin, A single image super-resolution method based on progressive-iterative approximation, IEEE Transactions on Multimedia 22 (2020) 1407–1422.
[36] H. Lin, T. Maekawa, C. Deng, Survey on geometric iterative methods and their applications, Computer aided design 95 (2018) 40–51.
[37] R. C. Veltkamp, W. Wesselink, Modeling 3d curves of minimal energy, in: Computer Graphics Forum, Vol. 14, Wiley Online Library, 1995, pp. 97–110.
[38] C. Zhang, P. Zhang, F. F. Cheng, Fairing spline curves and surfaces by minimizing energy, Computer-Aided Design 33 (13) (2001) 913–923.
[39] H. Meier, H. Nowacki, Interpolating curves with gradual changes in curvature, Computer Aided Geometric Design 4 (4) (1987) 297–305.
[40] X. Wang, F. F. Cheng, B. A. Barsky, Energy and b-spline interproximation, Computer-Aided Design 29 (7) (1997) 485–496.
[41] H. Lin, Q. Cao, X. Zhang, The convergence of least-squares progressive iterative approximation for singular least-squares fitting system, Journal of systems science and complexity 31 (6) (2018) 1618–1632.
Appendix

In this section, we show the convergence of the fairing-PIA method when the diagonal elements of matrix $\Omega$ are equal to a constant value $\omega$, and coefficient matrix $B$ is a positive semidefinite matrix.

**Lemma Appendix .1.** Let the diagonal elements of matrix $\Lambda$ satisfy $0 < \mu_i < \frac{2}{\lambda_{\max}(B)}$, $i = 1, 2, \cdots, n$, where $\lambda_{\max}$ is the largest eigenvalue of $B$. The eigenvalues $\beta$ of the matrix $\Lambda B(9)$ are all real and satisfy $0 \leq \beta < 2$.

**Proof.** Matrix $B$ is a positive semidefinite matrix. Thus, real orthogonal matrix $U$ and diagonal matrix $S = \text{diag}(s_1, s_2, \cdots, s_n)$, $s_i \geq 0$, $i = 1, 2, \cdots, n$ exist, such that $B = USU^T$. We obtain the following by denoting $C = S^{1/2}U$:

$$B = C^T C.$$  \hspace{1cm} (.1)

Suppose that $\beta$ is an arbitrary eigenvalue of matrix $\Lambda C^T C$ w.r.t. eigenvector $v$, i.e.,

$$\Lambda C^T Cv = \beta v.$$ \hspace{1cm} (.2)

In this case, we obtain

$$C \Lambda C^T (Cv) = \beta (Cv),$$

by multiplying both sides of Eq. (2) by $C$, indicating that $\beta$ is also an eigenvalue of the matrix $C \Lambda C^T$ w.r.t. eigenvector $Cv$. Moreover, for all nonzero vectors $x \in \mathbb{R}^n$, it holds

$$x^T C \Lambda C^T x = (x^T C \Lambda^{1/2}) (x^T C \Lambda^{1/2})^T \geq 0.$$ 

Therefore, we conclude that the matrix $C \Lambda C^T$ is a positive semidefinite matrix. Thus, all its eigenvalues are nonnegative real numbers, i.e., $\beta \geq 0$. According to Eq. (2), the eigenvalues of matrix $\Lambda B = \Lambda C^T C$ are also all nonnegative real numbers.

On the other hand, given that $0 < \mu_i < \frac{2}{\lambda_{\max}(B)}$, $i = 1, 2, \cdots, n$, the eigenvalues of $\Lambda B$ meet

$$0 \leq \lambda(\Lambda B) < \|\Lambda B\|_2 \leq \|\Lambda\|_2 \|B\|_2 = \lambda_{\max}(\Lambda) \lambda_{\max}(B) < 2.$$

**Remark Appendix .2.** Let $n_0$ denote the dimension of the zero eigenspace of $B$ (10). Given that $\Lambda$ is nonsingular, the ranks of the matrix $B$ and $\Lambda B$ are the same, i.e.,

$$\text{rank}(\Lambda B) = \text{rank}(B) = n - n_0.$$ 

Moreover, in the proof of Theorem Appendix .5, a lemma similar to Lemma 2.5 in [41] is used as follows:

**Lemma Appendix .3.** The algebraic multiplicity of the zero eigenvalue of matrix $\Lambda B$ is equal to its geometric multiplicity.

**Proof.** Based on Eq. (1), matrix $\Lambda B$ can be written as $\Lambda C^T C$. Then, the proof of this lemma is similar to that of Lemma 2.5 provided in [41].

**Lemma Appendix .4.** The Jordan canonical form of the matrix $\Lambda B(9)$ can be represented as

$$J = \begin{bmatrix} J_n(\beta_1, 1) & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & J_n(\beta_k, 1) \end{bmatrix},$$ \hspace{1cm} (.3)
where

\[
J_n(\beta, 1) = \begin{bmatrix}
\beta & 1 & & \\
& \beta & \ddots & \\
& & \ddots & 1 \\
& & & \beta
\end{bmatrix} \in \mathbb{R}^{n \times n}
\]

is called a Jordan block of size \(n_i\) with eigenvalue \(\beta_i\), and \(0 \leq \beta_i < 2\) and \(i = 1, 2, \ldots, k\) are the nonzero eigenvalues of \(\Lambda B\).

**Proof.** Based on Lemma. Appendix .1, the eigenvalues of \(\Lambda B\) are all nonnegative real numbers. Thus, the Jordan canonical form of \(\Lambda B\) can be written as

\[
J = \begin{bmatrix}
J_{n_1}(\beta_1, 1) & & & \\
& J_{n_2}(\beta_2, 1) & & \\
& & \ddots & \\
& & & J_{n_k}(\beta_k, 1)
\end{bmatrix},
\]

where \(J_{n_i}(\beta_i, 1)\) is a Jordan block of size \(n_i\) with nonzero eigenvalue \(\beta_i\) of \(\Lambda B\), \(i = 1, 2, \ldots, k\), and \(J_{m_j}(0, 1)\) is a Jordan block of size \(m_j\) with zero eigenvalue of \(\Lambda B\), \(j = 1, 2, \ldots, l\).

Moreover, Lemma. Appendix .3 states that the algebraic multiplicity of the zero eigenvalue of the matrix \(\Lambda B\) is equal to its geometric multiplicity. This finding means that the Jordan blocks w.r.t the zero eigenvalues of the matrix \(\Lambda B\) are all equal to the matrix \((0)_{1 \times 1}\).

**Theorem Appendix .5.** If \(B\) is a positive semidefinite matrix, then the iterative method \([6]\) is convergent.

**Proof.** A positive semidefinite matrix \(B\) can be decomposed as

\[
B = V \text{diag}(\lambda_1, \lambda_2, \ldots, \lambda_{n-n_0}, 0, \ldots, 0)V^T,
\]

where \(V\) is an orthogonal matrix, and \(\lambda_i\) and \(i = 1, 2, \ldots, n-n_0\) are both the nonzero eigenvalues and nonzero singular values of \(B\). Then, the Moore-Penrose (M-P) inverse of \(B\) has the corresponding decomposition as follows:

\[
B^+ = V \text{diag}(\frac{1}{\lambda_1}, \frac{1}{\lambda_2}, \ldots, \frac{1}{\lambda_{n-n_0}}, 0, \ldots, 0)V^T.
\]

Thus,

\[
B^+ B = V \text{diag}(1, 1, \ldots, 1, 1, 0, \ldots, 0)V^T. \quad (4)
\]

Based on Lemma. Appendix .4, the Jordan canonical form of the matrix \(\Lambda B\) is \(J \[4]\). We obtain the following by using the Jordan decomposition: \(\Lambda B = T^{-1} JT\), where \(T\) is an invertible matrix. Hence,

\[
I - \Lambda B = T^{-1} \begin{bmatrix}
J_{n_1}(1-\beta_1, -1) & & & \\
& J_{n_2}(1-\beta_2, -1) & & \\
& & \ddots & \\
& & & J_{n_k}(1-\beta_k, -1)
\end{bmatrix} T.
\]

20
Based on Lemma Appendix 1, \( 1 < 1 - \beta_i < 1, i = 1, 2, \cdots, k \). Then, we obtain the following by combining Eq. (4):

\[
\lim_{h \to \infty} (I - \lambda B)^h = T^{-1} \text{diag}(0, \cdots, 0, 1, \cdots, 1)T
\]

\[
= T^{-1}(I - \text{diag}(1, 1, \cdots, 1, 0, \cdots, 0)) T
\]

\[
= I - (VT)^{-1} B^+ B (VT).
\]  

(5)

On the other hand, the coefficient matrix \( B \) is singular; thus, the linear system in Eq. (17) has its solution if and only if

\[
(1 - \omega) BB^+ N^T Q = (1 - \omega) N^T Q.
\]

Therefore, we obtain the following by subtracting \((1 - \omega) BB^+ N^T Q\) from both sides of the iterative scheme (9):

\[
P^{[k+1]} - (1 - \omega) B^+ N^T Q = (I - \lambda B) P^{[k]} + (1 - \omega) \lambda R^T Q - (1 - \omega) B^+ N^T Q
\]

\[
= (I - \lambda B) P^{[k]} + (1 - \omega) \lambda B B^+ N^T Q - (1 - \omega) B^+ N^T Q
\]

\[
= (I - \lambda B) P^{[k]} - (1 - \omega) (I - \lambda B) B^+ N^T Q
\]

\[
= (I - \lambda B) (P^{[k]} - (1 - \omega) B^+ N^T Q)
\]

\[
= (I - \lambda B) P^{[k+1]} - (1 - \omega) B^+ N^T Q.
\]

From Eq. (5), when \( k \to \infty \), the above equation tends to

\[
P^{[\infty]} - (1 - \omega) B^+ N^T Q = (I - (VT)^{-1} B^+ B (VT)) \left( P^{[0]} - (1 - \omega) B^+ N^T Q \right).
\]

Consequently,

\[
P^{[\infty]} = (I - (VT)^{-1} B^+ B (VT)) \left( P^{[0]} - (1 - \omega) B^+ N^T Q \right) + (1 - \omega) B^+ N^T Q
\]

\[
= (1 - \omega) (VT)^{-1} B^+ B (VT) B^+ N^T Q + (I - (VT)^{-1} B^+ B (VT)) P^{[0]},
\]

where \( P^{[0]} \) is the initial control point, which can be randomly selected. When \( P^{[0]} = 0 \), the iterative method (9) converges to

\[
P^{[\infty]} = (1 - \omega) (VT)^{-1} B^+ B (VT) B^+ N^T Q.
\]  

(6)

Remark Appendix 6. If matrix \( V \) is the inverse matrix of \( T \), that is, \( VT = TV = I \), then Eq. (6) becomes

\[
P^{[\infty]} = (1 - \omega) B^+ N^T Q,
\]

where \( B^+ \) is the M-P pseudo-inverse of the matrix \( B \). Then the iterative method (9) converges to the solution of the classical energy minimization equation (17).