# The Convergence of Least-Squares Progressive Iterative Approximation for Singular Least-Squares Fitting System*

**LIN Hongwei · CAO Qi · ZHANG Xiaoting**

DOI:
Received: x x 20xx / Revised: x x 20xx

**Abstract** Data fitting is an extensively employed modeling tool in geometric design. With the advent of the big data era, the data sets to be fitted are made larger and larger, leading to more and more least-squares fitting systems with singular coefficient matrices. LSPIA (least-squares progressive iterative approximation) is an efficient iterative method for the least-squares fitting. However, the convergence of LSPIA for the singular least-squares fitting systems remains as an open problem. In this paper, we showed that LSPIA for the singular least-squares fitting systems is convergent. Moreover, in a special case, LSPIA converges to the Moore-Penrose (M-P) pseudo-inverse solution to the least-squares fitting result of the data set. This property makes LSPIA, an iterative method with clear geometric meanings, robust in geometric modeling applications. In addition, we discussed some implementation detail of LSPIA, and presented an example to validate the convergence of LSPIA for the singular least-squares fitting systems.

**Keywords** LSPIA, singular least-squares fitting system, data fitting, geometric modeling

## 1 Introduction

Least-squares fitting is a commonly employed approach in engineering applications and scientific research, including geometric modeling. With the advent of the *big data* era, least-squares fitting systems with *singular* coefficient matrices often appear, when the number of fitted data points is very large, or there are "holes" in the fitted data point set [1, 2]. LSPIA [3, 4] is an efficient iterative method for least-squares curve and surface fitting [5]. Although the convergence of LSPIA for the *nonsingular* least-squares fitting systems was shown in Refs. [3, 4], the convergence analysis of LSPIA for the *singular* least-squares fitting systems is not easy to perform, and remains as an open problem. In this paper, we will show that, when the coefficient matrix of a least-squares fitting system is *singular*, LSPIA is still convergent. This property of LSPIA will promote its applications in the large scale data fitting.

School of Mathematical Science, State Key Lab. of CAD&CG, Zhejiang University, Hangzhou, 310027, China
E-mail : hwlin@zju.edu.cn

*This research was supported by the Natural Science Foundation of China under Grant No.61379072

◇ *This paper was recommended for publication by Editor .*

The motivation of this paper comes from our research practices, where some singular least-squares fitting systems emerge. For examples, in generating trivariate B-spline solids by fitting tetrahedral meshes [6], and in fitting images with holes by T-spline surfaces [3], the coefficient matrices of least-squares fitting systems are singular. There, LSPIA was employed to solve the singular least-squares fitting systems, and converged to stable solutions. However, in Ref. [3, 6], the convergence of LSPIA for the singular least-squares fitting systems was not proved.

The progressive-iterative approximation (PIA) method was first developed in [7, 8]. Because PIA endows iterative methods with geometric meanings, it is suitable to handle geometric problems appearing in the field of geometric design. It has been shown that the PIA method is convergent for the B-spline fitting [4, 9], NURBS fitting [10], T-spline fitting [3], subdivision surface fitting [11–13], as well as curve and surface fitting with the totally positive basis [8]. The iterative method of the geometric interpolation (GI) [14] is similar as that of PIA. While PIA depends on the parametric distance, the iterations of GI rely on the geometric distance. Moreover, the PIA and GI methods have been employed in some practical applications, such as reverse engineering [15, 16], curve design [17], surface-surface intersection [18], trivariate B-spline solid generation [6], etc.

The structure of this paper is as follows. In Section 2, we show the convergence of LSPIA for the singular least-squares fitting systems. In Section 3, some implementation details of LSPIA are discussed, and an example is illustrated. Finally, Section 4 concludes the paper.

## 2 The LSPIA iterative format and its convergence analysis

To integrate the LSPIA iterative methods for B-spline curves, B-spline patches, trivariate B-spline solids, and T-splines, their representations are rewritten as the following form,

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

Specifically, T-spline patches [19] and trivariate T-spline solids [20] can be naturally written as the form (1). Moreover,

- If $\boldsymbol{P}(\boldsymbol{t})$ (1) is a B-spline curve, then, $\boldsymbol{t}$ is a scalar $u$, and $B_i(\boldsymbol{t}) = N_i(u)$, where $N_i(u)$ is a B-spline basis function.

- If $\boldsymbol{P}(\boldsymbol{t})$ (1) is a B-spline patch with $n_u \times n_v$ control points, then, $\boldsymbol{t} = (u, v)$, and $B_i(\boldsymbol{t}) = N_i(u)N_i(v)$, where $N_i(u)$ and $N_i(v)$ are B-spline basis functions. In the control net of the B-spline patch, the original index of $N_i(u)$ is $[\frac{i}{n_u}]$, and the original index of $N_i(v)$ is $(i \bmod n_u)$, where $[\frac{i}{n_u}]$ represents the maximum integer not exceeding $\frac{i}{n_u}$, and $(i \bmod n_u)$ is the module of $i$ by $n_u$.

- If $\boldsymbol{P}(\boldsymbol{t})$ is a trivariate B-spline solid with $n_u \times n_v \times n_w$ control points, then $\boldsymbol{t} = (u, v, w)$, and $B_i(\boldsymbol{t}) = N_i(u)N_i(v)N_i(w)$. In the control net of the trivariate B-spline solid, the original index of $N_i(w)$ is $[\frac{i}{n_u n_v}]$, the original index of $N_i(u)$ is $[\frac{(i \bmod n_u n_v)}{n_u}]$, and the original index of $N_i(v)$ is $((i \bmod n_u n_v) \bmod n_u)$.

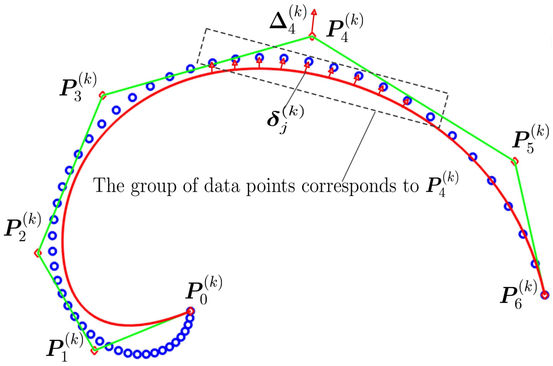


Figure 1: All of DVDs $\boldsymbol{\delta}_j^k$ distributed to the control point $\boldsymbol{P}_4^{(k)}$ are weighted averaged to generate the DVC $\boldsymbol{\Delta}_i^{(k)}$. Here, blue circles are the data points, and the red curve is the $k^{th}$ curve $\boldsymbol{P}^{(k)}(u)$.

Suppose we are given a data point set

$$\{\boldsymbol{Q}_j = (x_i, y_i, z_i), j = 0, 1, \cdots, m\}, \tag{2}$$

each of which is assigned a parameter $\boldsymbol{t}_j, j = 0, 1, \cdots, m$. Let the initial form be,

$$\boldsymbol{P}^{(0)}(\boldsymbol{t}) = \sum_{i=0}^{n} \boldsymbol{P}_i^{(0)} B_i(\boldsymbol{t}),\ n \leq m. \tag{3}$$

It should be noted that, though the initial control points $\boldsymbol{P}_i^{(0)}$ are usually chosen from the given data points, the initial control points are unrelated to the convergence of LSPIA. To perform LSPIA iterations, data points are classified into groups (Fig. 1). All of data points with parameters $\boldsymbol{t_j}$ satisfying $B_i(\boldsymbol{t_j}) \neq 0$ are classified into the $i^{th}$ group, corresponding to the $i^{th}$ control point (3).

After the $k^{th}$ iteration of LSPIA, the $k^{th}$ form $\boldsymbol{P}^{(k)}(\boldsymbol{t})$ is generated,

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

To produce the $(k+1)^{st}$ form $\boldsymbol{P}^{(k+1)}(\boldsymbol{t})$, we first calculate the *difference vectors for data points* (DVD) (Fig. 1),

$$\boldsymbol{\delta}_j^{(k)} = \boldsymbol{Q}_j - \boldsymbol{P}^{(k)}(\boldsymbol{t}_j),\ j = 0, 1, \cdots, m.$$

And then two procedures are performed, i.e., *vector distribution* and *vector gathering* (Fig. 1). In the vector distribution procedure, all of DVDs corresponding to data points in the $i^{th}$ group are distributed to the $i^{th}$ control point $\boldsymbol{P}_i^{(k)}$; in the vector gathering procedure, all of DVDs distributed to the $i^{th}$ control point $\boldsymbol{P}_i^{(k)}$ are weighted averaged to generate the *difference vectors for control points* (DVC) (Fig. 1),

$$\boldsymbol{\Delta}_i^{(k)} = \frac{\sum_{j \in I_i} B_i(\boldsymbol{t_j}) \boldsymbol{\delta}_j}{\sum_{j \in I_i} B_i(\boldsymbol{t_j})},\ i = 0, 1, \cdots, n,$$

where $I_i$ is the index set of the data points in the $i^{th}$ group. Then, the new control point $\boldsymbol{P}_i^{(k+1)}$ is produced by adding the $i^{th}$ DVC $\boldsymbol{\Delta}_i^{(k)}$ to $\boldsymbol{P}_i^{(k)}$, i.e.,

$$\boldsymbol{P}_i^{(k+1)} = \boldsymbol{P}_i^{(k)} + \boldsymbol{\Delta}_i^{(k)}, \ i = 0, 1, \cdots, n, \tag{4}$$

leading to the $(k+1)^{st}$ iteration form,

$$\boldsymbol{P}^{(k+1)}(\boldsymbol{t}) = \sum_{i=0}^{n} \boldsymbol{P}_i^{(k+1)} B_i(\boldsymbol{t}). \tag{5}$$

In this way, we get a sequence of iterative forms $\{\boldsymbol{P}^k(\boldsymbol{t}), \ k = 0, 1, \cdots\}$. Let,

$$P^{(k)} = [\boldsymbol{P}_0^{(k)}, \boldsymbol{P}_1^{(k)}, \cdots, \boldsymbol{P}_n^{(k)}]^T, \tag{6}$$

$$Q = [\boldsymbol{Q}_0, \boldsymbol{Q}_1, \cdots, \boldsymbol{Q}_m]^T. \tag{7}$$

From Eq. (4), it follows,

$$\begin{aligned}
\boldsymbol{P}_i^{(k+1)} &= \boldsymbol{P}_i^{(k)} + \frac{1}{\sum_{j\in I_i} B_i(\boldsymbol{t}_j)} \sum_{j\in I_i} B_i(\boldsymbol{t}_j)(\boldsymbol{Q}_j - \boldsymbol{P}^{(k)}(\boldsymbol{t}_j)) \\
&= \boldsymbol{P}_i^{(k)} + \frac{1}{\sum_{j\in I_i} B_i(\boldsymbol{t}_j)} \sum_{j\in I_i} B_i(\boldsymbol{t}_j) \left( \boldsymbol{Q}_j - \sum_{l=0}^{n} \boldsymbol{P}_l^{(k)} B_l(\boldsymbol{t}_j) \right)
\end{aligned}$$

Therefore, we get the LSPIA iterative method in matrix form,

$$P^{(k+1)} = P^{(k)} + \Lambda A^T (Q - AP^{(k)}), \ k = 0, 1, \cdots \tag{8}$$

where, $\Lambda = diag\left( \frac{1}{\sum_{j\in I_0} B_0(\boldsymbol{t}_j)}, \frac{1}{\sum_{j\in I_1} B_1(\boldsymbol{t}_j)}, \cdots, \frac{1}{\sum_{j\in I_n} B_n(\boldsymbol{t}_j)} \right)$ is a diagonal matrix, and $A$ is a collocation matrix,

$$A = \begin{bmatrix} B_0(\boldsymbol{t}_0) & B_1(\boldsymbol{t}_0) & \cdots & B_n(\boldsymbol{t}_0) \\ B_0(\boldsymbol{t}_1) & B_1(\boldsymbol{t}_1) & \cdots & B_n(\boldsymbol{t}_1) \\ \vdots & \vdots & & \vdots \\ B_0(\boldsymbol{t}_m) & B_1(\boldsymbol{t}_m) & \cdots & B_n(\boldsymbol{t}_m) \end{bmatrix}_{(m+1)\times(n+1)}.$$

**Remark 2.1** The iterative method (8) is the one presented in Ref. [3], and the iterative method developed in [4] is actually a special case of the method (8). In the special case [4], diagonal elements of the diagonal matrix $\Lambda$ are equal to each other.

**Remark 2.2** Because the diagonal elements of the diagonal matrix $\Lambda$ in the iterative method (8) are all positive, the diagonal matrix $\Lambda$ is nonsingular.

To show the convergence of the LSPIA iterative method (8), it is rewritten as,

$$P^{(k+1)} = (I - \Lambda A^T A)P^{(k)} + \Lambda A^T Q. \tag{9}$$

As stated above, it has been shown that, when the coefficient matrix $A^T A$ of a least-squares fitting system is *nonsingular*, the LSPIA iterative method is convergent [3, 4]. However, the convergence of LSPIA with *singular* coefficient matrix $A^TA$ still remains as an open problem. In the following, we will show that, even the matrix $A$ is not of full rank, and then $A^TA$ is *singular*, the LSPIA iterative method (8) (9) is still convergent.

We first show some lemmas.

**Lemma 2.3** *The eigenvalues $\lambda$ of the matrix $\Lambda A^TA$ are all real, and satisfy $0 \leq \lambda \leq 1$.*

*Proof* On one hand, suppose $\lambda$ is an arbitrary eigenvalue of the matrix $\Lambda A^TA$ with eigenvector $\boldsymbol{v}$, i.e.,

$$\Lambda A^TA\boldsymbol{v} = \lambda\boldsymbol{v}. \tag{10}$$

By multiplying $A$ at both sides of Eq. (10), we have,

$$A\Lambda A^T(A\boldsymbol{v}) = \lambda(A\boldsymbol{v}).$$

It means that $\lambda$ is also an eigenvalue of the matrix $A\Lambda A^T$ with eigenvector $A\boldsymbol{v}$. Moreover, $\forall\boldsymbol{x} \in R^{m+1}$, because,

$$\boldsymbol{x}^TA\Lambda A^T\boldsymbol{x} = \boldsymbol{x}^TA\Lambda^{\frac{1}{2}}(\Lambda^{\frac{1}{2}})^TA^T\boldsymbol{x} = (\boldsymbol{x}^TA\Lambda^{\frac{1}{2}})(\boldsymbol{x}^TA\Lambda^{\frac{1}{2}})^T \geq 0,$$

the matrix $A\Lambda A^T$ is a positive semidefinite matrix. Eigenvalues of a positive semidefinite matrix are all nonnegative, so $\lambda$ is real, and $\lambda \geq 0$.

On the other hand, because the B-spline basis functions are nonnegative and form a partition of unity, it holds, $\|A\|_\infty = 1$, where $\|\cdot\|_\infty$ denotes the $\infty$-norm of a matrix. Together with $\|\Lambda A^T\|_\infty = 1$, we have,

$$\|\Lambda A^TA\|_\infty \leq \|\Lambda A^T\|_\infty \|A\|_\infty = 1.$$

Therefore, the eigenvalue $\lambda$ of the matrix $\Lambda A^TA$ satisfies,

$$\lambda \leq \|\Lambda A^TA\|_\infty \leq 1.$$

In conclusion, the eigenvalues $\lambda$ of the matrix $\Lambda A^TA$ are all real, and satisfy $0 \leq \lambda \leq 1$. □

Because $A^TA$ (9) is singular, $\Lambda A^TA$ is also singular, and then $\lambda = 0$ is its eigenvalue. The following lemma deals with the relationship between the algebraic multiplicity and geometric multiplicity of the zero eigenvalue $\lambda = 0$ of $\Lambda A^TA$.

**Remark 2.4** In this paper, we assume that the dimension of the zero eigenspace of $A^TA$ is $n_0$. So, the rank of the $(n+1)\times(n+1)$ matrix $A^TA$ is

$$rank(A^TA) = n - n_0 + 1.$$

Because $\Lambda$ is nonsingular (refer to Remark 2.2), we have

$$rank(\Lambda A^TA) = n - n_0 + 1.$$

**Lemma 2.5** *The algebraic multiplicity of the zero eigenvalue of matrix $\Lambda A^TA$ is equal to its geometric multiplicity.*

*Proof* The proof consists of three parts.

(1) *The algebraic multiplicity of the zero eigenvalue of matrix $A^TA$ is equal to its geometric multiplicity.* Because $A^TA$ is a positive semidefinite matrix, it is a diagonalizable matrix. Then, for any eigenvalue of $A^TA$ (including the zero eigenvalue), its algebraic multiplicity is equal to its geometric multiplicity. In Remark 2.4, we assume that the dimension of the zero eigenspace of $A^TA$, i.e., the geometric multiplicity of zero eigenvalue of $A^TA$, is $n_0$. So, the algebraic multiplicity and geometric multiplicity of zero eigenvalue of $A^TA$ are both $n_0$.

(2) *The geometric multiplicity of the zero eigenvalue of the matrix $\Lambda A^TA$ is equal to that of the matrix $A^TA$.* Denote the eigenspaces of the matrices $\Lambda A^TA$ and $A^TA$ associated with the zero eigenvalue $\lambda = 0$ as $K_0(\Lambda A^TA)$ and $K_0(A^TA)$, respectively. The geometric multiplicities of the zero eigenvalue of matrices $\Lambda A^TA$ and $A^TA$ are dimensions of $K_0(\Lambda A^TA)$ and $K_0(A^TA)$, respectively.

Note that the matrix $\Lambda$ is nonsingular (Remark 2.2). On one hand, $\forall \boldsymbol{v} \in K_0(\Lambda A^TA), \Lambda A^TA\boldsymbol{v} = 0$, leading to $A^TA\boldsymbol{v} = \Lambda^{-1}0 = 0$. So, $\boldsymbol{v} \in K_0(A^TA)$. On the other hand, $\forall \boldsymbol{w} \in K_0(A^TA), A^TA\boldsymbol{w} = 0$, resulting in $\Lambda A^TA\boldsymbol{w} = 0$. So, $\boldsymbol{w} \in K_0(\Lambda A^TA)$. In conclusion, $K_0(\Lambda A^TA) = K_0(A^TA)$. Therefore, the geometric multiplicity of the zero eigenvalue of the matrix $\Lambda A^TA$ is equal to that of the matrix $A^TA$.

(3) *The algebraic multiplicity of the zero eigenvalue of the matrix $\Lambda A^TA$ is equal to that of the matrix $A^TA$.* Denote $I$ as an $(n+1)\times(n+1)$ identity matrix,

$$A^TA = (b_{ij})_{(n+1)\times(n+1)}, \text{ and, } \Lambda = diag(d_0, d_1, \cdots, d_n), \tag{11}$$

where $d_i > 0, i = 0, 1, \cdots, n$.

The characteristic polynomial of $A^TA$ and $\Lambda A^TA$ can be written as [21, pp.42],

$$p_{A^TA}(\lambda) = det(\lambda I - A^TA) = \lambda^{n+1} - E_1(A^TA)\lambda^n + E_2(A^TA)\lambda^{n-1} + \cdots + (-1)^{n+1}E_{n+1}(A^TA), \tag{12}$$

and,

$$p_{\Lambda A^TA}(\lambda) = det(\lambda I - \Lambda A^TA) = \lambda^{n+1} - E_1(\Lambda A^TA)\lambda^n + E_2(\Lambda A^TA)\lambda^{n-1} + \cdots + (-1)^{n+1}E_{n+1}(\Lambda A^TA), \tag{13}$$

where $E_k(A^TA), k = 1, 2, \cdots, n+1$ are the sums of the $k \times k$ principal minors of $A^TA$, and $E_k(\Lambda A^TA), k = 1, 2, \cdots, n+1$ are the sums of the $k \times k$ principal minors of $\Lambda A^TA$.

On one hand, because the algebraic multiplicity of the zero eigenvalue of $A^TA$ is $n_0$ (see Part (1)), its characteristic polynomial (12) can be represented as,

$$p_{A^TA} = \lambda^{n_0}\left(\lambda^{n-n_0+1} - E_1(A^TA)\lambda^{n-n_0} + \cdots + (-1)^{n-n_0+1}E_{n-n_0+1}(A^TA)\right).$$

where $E_{n-n_0+1}(A^TA) \neq 0$. Moreover, because $A^TA$ is positive semi-definite, all of its principal minors are nonnegative. Therefore, we have $E_{n-n_0+1}(A^TA) > 0$. Consequently, all of $(n-n_0+1)\times(n-n_0+1)$ principal minors of $A^TA$ are nonnegative, and there is at least one positive $(n-n_0+1)\times(n-n_0+1)$ principal minor of $A^TA$.

On the other hand, because $rank(\Lambda A^T A) = n - n_0 + 1$ (Remark 2.4), all of $l \times l$ $(l > n - n_0 + 1)$ principal minors of $\Lambda A^T A$ are zero. Therefore,

$$E_l(\Lambda A^T A) = 0,\ l > n - n_0 + 1. \tag{14}$$

Denote $M_{A^T A}(i_1, i_2, \cdots, i_k)$ and $M_{\Lambda A^T A}(i_1, i_2, \cdots, i_k)$ are the $k \times k$ principal minors of $A^T A$ and $\Lambda A^T A$, respectively. Now, considering a $k \times k$ principal minor of $\Lambda A^T A$ (refer to Eq. (11)), we have,

$$M_{\Lambda A^T A}(i_1, i_2, \cdots, i_k) = det \begin{pmatrix} d_{i_1} b_{i_1, i_1} & \cdots & d_{i_1} b_{i_1, i_k} \\ \cdots & \cdots & \cdots \\ d_{i_k} b_{i_k, i_1} & \cdots & d_{i_k} b_{i_k, i_k} \end{pmatrix} = (\prod_{j=1}^{k} d_{i_j}) M_{A^T A}(i_1, i_2, \cdots, i_k),$$

where $\prod_{j=1}^{k} d_{i_j} > 0$ (Eq. (11)). In other words, the principal minor $M_{\Lambda A^T A}(i_1, i_2, \cdots, i_k)$ of $\Lambda A^T A$ is the product of a principal minor $M_{A^T A}(i_1, i_2, \cdots, i_k)$ of $A^T A$ and a positive value $\prod_{j=1}^{k} d_{i_j}$. Together with the result that all of $(n - n_0 + 1) \times (n - n_0 + 1)$ principal minors of $A^T A$ are nonnegative, and there is at least one positive $(n - n_0 + 1) \times (n - n_0 + 1)$ principal minor of $A^T A$, the sum of all $(n - n_0 + 1) \times (n - n_0 + 1)$ principal minors of $\Lambda A^T A$, namely, $E_{n-n_0+1}(\Lambda A^T A)$, is positive. That is,

$$E_{n-n_0+1}(\Lambda A^T A) > 0. \tag{15}$$

By Eqs. (14) and (15), the characteristic polynomial of $\Lambda A^T A$ (13) can be transformed as,

$$p_{\Lambda A^T A} = \lambda^{n_0} \left( \lambda^{n-n_0+1} - E_1(\Lambda A^T A) \lambda^{n-n_0} + \cdots + (-1)^{n-n_0+1} E_{n-n_0+1}(\Lambda A^T A) \right),$$

where $E_{n-n_0+1}(\Lambda A^T A) > 0$. It means that the algebraic multiplicity of the zero eigenvalue of $\Lambda A^T A$ is $n_0$, equal to the algebraic multiplicity of the zero eigenvalue of $A^T A$.

Combining the results of part (1)-(3), we have shown that the algebraic multiplicity of the zero eigenvalue of matrix $\Lambda A^T A$ is equal to its geometric multiplicity. □

Denote $J_r(a, b)$ as a $r \times r$ matrix block,

$$J_r(a, b) = \begin{pmatrix} a & b & & & \\ & a & b & \text{\Large 0} & \\ & & \ddots & \ddots & \\ & \text{\Large 0} & & \ddots & b \\ & & & & a \end{pmatrix}_{r \times r}. \tag{16}$$

Specifically, $J_r(\lambda, 1)$ is a $r \times r$ *Jordan block*. Lemma 2.3 and 2.5 result in Lemma 2.6 as follows.

**Lemma 2.6** *The Jordan canonical form of the matrix $\Lambda A^T A$ (9) can be written as,*

$$J = \begin{pmatrix} J_{n_1}(\lambda_1, 1) & & & & & & \\ & J_{n_2}(\lambda_2, 1) & & \mathit{0} & & & \\ & & \ddots & & & & \\ & & & J_{n_k}(\lambda_k, 1) & & & \\ & & & & 0 & & \\ & \mathit{0} & & & & \ddots & \\ & & & & & & 0 \end{pmatrix}_{(n+1)\times(n+1)}, \tag{17}$$

*where $0 < \lambda_i \leq 1, i = 1, 2, \cdots, k$ are the nonzero eigenvalues of $\Lambda A^T A$, which need not be distinct, and $J_{n_i}(\lambda_i, 1)$ (16) is an $n_i \times n_i$ Jordan block, $i = 1, 2, \cdots, k$.*

*Proof* Based on Lemma 2.3, the eigenvalues $\lambda_i$ of $\Lambda A^T A$ are all real and lie in the interval $[0, 1]$, so the Jordan canonical form of $\Lambda A^T A$ can be written as,

$$\begin{pmatrix} J_{n_1}(\lambda_1, 1) & & & & & \\ & \ddots & & 0 & & \\ & & J_{n_k}(\lambda_k, 1) & & & \\ & & & J_{m_1}(0, 1) & & \\ & 0 & & & \ddots & \\ & & & & & J_{m_l}(0, 1) \end{pmatrix}_{(n+1)\times(n+1)},$$

where $0 < \lambda_i \leq 1, i = 1, 2, \cdots, k$ are the nonzero eigenvalues of $\Lambda A^T A$, which need not be distinct, and $J_{n_i}(\lambda_i, 1)$ (16) is an $n_i \times n_i$ Jordan block, $i = 1, 2, \cdots, k$; $J_{m_j}(0, 1)$ is an $m_j \times m_j$ Jordan block corresponding to the zero eigenvalue of $\Lambda A^T A$, $j = 1, 2, \cdots, l$.

According to the theory on the Jordan canonical form [21, p.129], the number of the Jordan blocks corresponding to an eigenvalue is the geometric multiplicity of the eigenvalue, and the sum of the orders of all Jordan blocks corresponding to an eigenvalue equals its algebraic multiplicity. Based on Lemma 2.5, the algebraic multiplicity of the zero eigenvalue of the matrix $\Lambda A^T A$ is equal to its geometric multiplicity, so the Jordan blocks $J_{m_j}(0, 1),\ j = 1, 2, \cdots, l$ corresponding to the zero eigenvalue of $\Lambda A^T A$ are all $1 \times 1$ matrix $(0)_{1\times1}$. This proves Lemma 2.6. □

Denote $(A^T A)^+$ as the Moore-Penrose (M-P) pseudo-inverse of the matrix $A^T A$. We have the following lemma.

**Lemma 2.7** *There exists an orthogonal matrix $V$, such that,*

$$V^T (A^T A)^+ (A^T A) V = diag(\underbrace{1, 1, \cdots, 1}_{n-n_0+1}, \underbrace{0, 0, \cdots, 0}_{n_0}). \tag{18}$$

*Proof* Because $rank(A^TA) = n - n_0 + 1$ (Remark 2.4), and $A^TA$ is a positive semidefinite matrix, it has the singular value decomposition (SVD),

$$A^TA = Vdiag(\delta_1, \delta_2, \cdots, \delta_{n-n_0+1}, \underbrace{0, \cdots, 0}_{n_0})V^T, \tag{19}$$

where $V$ is an orthogonal matrix, $\delta_i, i = 1, 2, \cdots, n - n_0 + 1$ are the singular values of $A^TA$. Then, the M-P pseudo-inverse of $A^TA$ is,

$$(A^TA)^+ = Vdiag(\frac{1}{\delta_1}, \frac{1}{\delta_2}, \cdots, \frac{1}{\delta_{n-n_0+1}}, \underbrace{0, \cdots, 0}_{n_0})V^T.$$

Therefore,

$$(A^TA)^+(A^TA) = Vdiag(\underbrace{1, 1, \cdots, 1}_{n-n_0+1}, \underbrace{0, 0, \cdots, 0}_{n_0})V^T,$$

where $V$ is an orthogonal matrix. It leads to Eq. (18). □

Based on the lemmas stated above, we can show the convergence of the LSPIA iterative method (9) when $A^TA$ is singular.

**Theorem 2.8** *When $A^TA$ (9) is singular, the LSPIA iterative method (9) is convergent.*

*Proof* By Lemma 2.6, the Jordan canonical form of the matrix $\Lambda A^TA$ (9) is $J$ (17). Then, there exists an invertible matrix $W$, such that,

$$\Lambda A^TA = W^{-1}JW.$$

Therefore (refer to Eq. (17)),

$$I - \Lambda A^TA =$$

$$W^{-1}\begin{pmatrix} J_{n_1}(1-\lambda_1, -1) & & & & & & \\ & J_{n_2}(1-\lambda_2, -1) & & & & 0 & \\ & & \ddots & & & & \\ & & & J_{n_k}(1-\lambda_k, -1) & & & \\ & & & & 1 & & \\ & 0 & & & & \ddots & \\ & & & & & & 1 \end{pmatrix}W,$$

where $0 \leq 1 - \lambda_i < 1, i = 1, 2, \cdots, k$. Together with Lemma 2.7, it holds,

$$\begin{aligned} \lim_{l\to\infty}(I - \Lambda A^TA)^l &= W^{-1}diag(\underbrace{0, \cdots, 0}_{n-n_0+1}, \underbrace{1, \cdots, 1}_{n_0})W \\ &= I - W^{-1}diag(\underbrace{1, \cdots, 1}_{n-n_0+1}, \underbrace{0, \cdots, 0}_{n_0})W \\ &= I - W^{-1}V^T(A^TA)^+(A^TA)VW \\ &= I - (VW)^{-1}(A^TA)^+(A^TA)(VW). \end{aligned} \tag{20}$$

Now, consider the linear system $A^TAX = A^TQ$ (refer to Eq. (7)). It has solutions if and only if [22],

$$(A^TA)(A^TA)^+(A^TQ) = A^TQ. \tag{21}$$

Subtracting $(A^TA)^+A^TQ$ from both sides of the iterative method (9), together with Eq. (21), we have,

$$\begin{aligned}
&P^{(k+1)} - (A^TA)^+A^TQ \\
&= (I - \Lambda A^TA)P^{(k)} + \Lambda A^TQ - (A^TA)^+A^TQ \\
&= (I - \Lambda A^TA)P^{(k)} + \Lambda(A^TA)(A^TA)^+(A^TQ) - (A^TA)^+A^TQ \\
&= (I - \Lambda A^TA)P^{(k)} - (I - \Lambda A^TA)(A^TA)^+A^TQ \\
&= (I - \Lambda A^TA)(P^{(k)} - (A^TA)^+A^TQ) \\
&= (I - \Lambda A^TA)^{k+1}(P^{(0)} - (A^TA)^+A^TQ).
\end{aligned} \tag{22}$$

Owing to Eq. (20), it follows,

$$\begin{aligned}
&P^{(\infty)} - (A^TA)^+A^TQ = \lim_{k\to\infty}(P^{(k+1)} - (A^TA)^+A^TQ) \\
&= \lim_{k\to\infty}(I - \Lambda A^TA)^{k+1}(P^{(0)} - (A^TA)^+A^TQ) \\
&= (I - (VW)^{-1}(A^TA)^+(A^TA)(VW))(P^{(0)} - (A^TA)^+A^TQ)
\end{aligned} \tag{23}$$

By simple computation, Eq. (23) changes to,

$$P^{(\infty)} = (VW)^{-1}(A^TA)^+(A^TA)VW(A^TA)^+A^TQ + (I - (VW)^{-1}(A^TA)^+(A^TA)(VW))P^{(0)}. \tag{24}$$

Therefore, the iterative method (9) is convergent when $A^TA$ is singular. Theorem 2.8 is proved. □

**Remark 2.9** Returning to Eq. (24), if $V$ is the inverse matrix of $W$, i.e., $VW = I$, it becomes,

$$\begin{aligned}
P^{(\infty)} &= (A^TA)^+(A^TA)(A^TA)^+A^TQ + (I - (A^TA)^+(A^TA))P^{(0)} \\
&= (A^TA)^+A^TQ + (I - (A^TA)^+(A^TA))P^{(0)},
\end{aligned} \tag{25}$$

where, $P^{(0)}$ is an arbitrary initial value. Eq. (25) is the M-P pseudo-inverse solution of the linear system $A^TAX = A^TQ$, which is the normal equation of the least-squares fitting to the data points (2). Because $P^{(0)}$ is an arbitrary value, there are infinite solutions to the normal equation $A^TAX = A^TQ$. Within these solutions, $(A^TA)^+(A^TQ)$ is the one with the minimum Euclidean norm [21].

As mentioned in Remark 2.1, the iterative method developed in [4] is a special case of the method (8). In the special case [4], diagonal elements of the diagonal matrix $\Lambda$ (Eq. (9)) are equal to each other. Denoting $\alpha$ as the diagonal elements of the diagonal matrix $\Lambda$, the iterative method developed in [4] can be written as (refer to Eq. (9)),

$$P^{(k+1)} = (I - \alpha A^TA)P^{(k)} + \alpha A^TQ. \tag{26}$$

In this special case, we have the following theorem.

**Theorem 2.10** *If $A^TA$ is singular, and the spectral radius $\rho(\alpha A^TA) \leq 1$, the iterative method (26) converges to the M-P pseudo-inverse solution of the linear system $A^TAX = A^TQ$ (refer to Eq. (7)). Moreover, if the initial value $P^{(0)} = 0$, the iterative method (26) converges to $(A^TA)^+(A^TQ)$, i.e., the M-P pseudo-inverse solution of the linear system $A^TAX = A^TQ$ with the minimum Euclidean norm.*

*Proof* Because $A^TA$ is both a normal matrix and a positive semidefinite matrix, its eigen decomposition is the same as its singular value decomposition [21], and has the form presented in Eq. (19). So, we have,

$$\alpha A^TA = V diag(\alpha\delta_1, \alpha\delta_2, \cdots, \alpha\delta_{n-n_0+1}, \underbrace{0, \cdots, 0}_{n_0})V^T,$$

where $V$ is an orthogonal matrix, and $\alpha\delta_i, i = 1, 2, \cdots, n - n_0 + 1$ are both the nonzero eigenvalues and nonzero singular values of $\alpha A^TA$. Because $\rho(\alpha A^TA) \leq 1$, it holds

$$0 < \alpha\delta_i \leq 1, i = 1, 2, \cdots, n - n_0 + 1.$$

Then, based on Lemma 2.7, we have,

$$\begin{aligned}\lim_{l\to\infty}(I - \alpha A^TA)^l &= V diag(\underbrace{0, \cdots, 0}_{n-n_0+1}, \underbrace{1, \cdots, 1}_{n_0})V^T \\ &= I - V diag(\underbrace{1, \cdots, 1}_{n-n_0+1}, \underbrace{0, \cdots, 0}_{n_0})V^T \\ &= I - VV^T(A^TA)^+(A^TA)VV^T \\ &= I - (A^TA)^+(A^TA).\end{aligned}$$

Similar as the deduction in the proof of Theorem 2.8 (Eqs. (22) (23)), we have,

$$P^{(k+1)} - (A^TA)^+A^TQ = (I - \alpha A^TA)^{k+1}(P^{(0)} - (A^TA)^+A^TQ),$$

and,

$$\begin{aligned}P^{(\infty)} - (A^TA)^+A^TQ &= \lim_{k\to\infty}(P^{(k+1)} - (A^TA)^+A^TQ) \\ &= \lim_{k\to\infty}(I - \alpha A^TA)^{k+1}(P^{(0)} - (A^TA)^+A^TQ) \\ &= (I - (A^TA)^+(A^TA))(P^{(0)} - (A^TA)^+A^TQ).\end{aligned}$$

Therefore,

$$P^{(\infty)} = (A^TA)^+A^TQ + (I - (A^TA)^+(A^TA))P^{(0)},$$

where $P^{(0)}$ is an arbitrary initial value. It is the M-P pseudo-inverse solution of the linear system $A^TAX = A^TQ$. Moreover, when $P^{(0)} = 0$, $P^{(k)}$ converges to $P^{(\infty)} = (A^TA)^+A^TQ$, i.e., the M-P pseudo-inverse solution with the minimum Euclidean norm. □

## 3 Discussion and Example

In this section, we will discuss some implementation detail on LSPIA (8), and illustrate an example for validating the convergence of LSPIA when the matrix $A^T A$ is singular. The numerical examples demonstrated in Section 3.2 and 3.3 are implemented by Matlab 2013b, and run on a PC with $2.9G$ CPU, and $8G$ memory.

### 3.1 Computational complexity and memory cost

First, we will measure the computational complexity and memory cost per iteration step of the LSPIA iteration method (8). Suppose the matrix $A$ is of $(m+1) \times (n+1)$, and $Q$ is of $(m+1) \times 1$. Then, each iteration step of the LSPIA method (8) needs:

- once matrix-vector multiplication $M_1 = AP^{(k)}$, cost $(m+1)(n+1)$ multiplications and $(m+1)n$ additions,
- once vector-vector subtraction $M_2 = Q - M_1$, cost $(m+1)$ subtractions,
- once matrix-vector multiplication $M_3 = A^T M_2$, cost $(m+1)(n+1)$ multiplications, and $(n+1)m$ additions,
- once matrix-vector multiplications $M_4 = \Lambda M_3$, cost $n+1$ multiplications, and,
- once vector-vector subtraction $P^{(k)} - M_4$, cost $n+1$ subtractions.

In total, one step iteration of the LSPIA method needs $(2m+3)(n+1)$ multiplications, and $2(m+1)(n+1)$ additions (subtractions). Moreover, in each iteration, it is required to store two $(n+1) \times 1$ vectors, $P^{(k)}$ and $P^{(k+1)}$, one $(m+1) \times 1$ vector $Q$, one $(m+1) \times (n+1)$ matrix $A$, and a diagonal matrix $\Lambda$ with $n+1$ diagonal elements. Therefore, each LSPIA iteration requires $(m+4)(n+1) + m + 1$ unit memory. It should be pointed out that, in the large scale data fitting, the matrix $A$ (8) is usually sparse, and then the computational complexity and memory requirement will be greatly reduced.

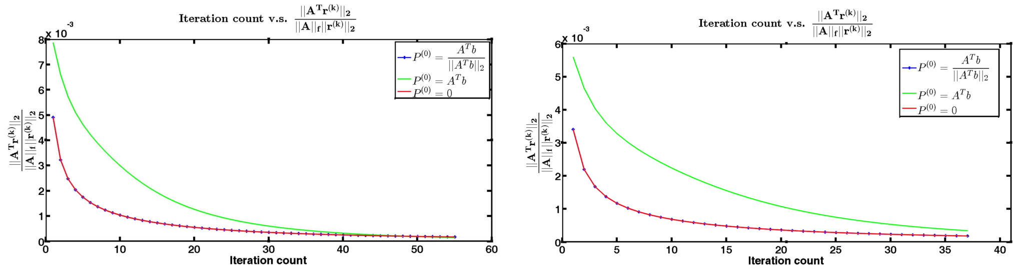
### 3.2 Influence of the initial value $P^{(0)}$

In this section, the influence of the initial value $P^{(0)}$ (6) is considered. We select three commonly employed initial values, i.e.,

$$P^{(0)} = 0,\ P^{(0)} = \frac{A^T Q}{\|A^T Q\|_2},\ \text{ and } P^{(0)} = A^T Q, \tag{27}$$

where, $\|\cdot\|_2$ is the Euclidean norm of a vector. As stated in Section 2 (also refer to Fig. 1), at the beginning of the LSPIA iterations, the data points are classified into groups. In geometry, $A^T Q$ means that the weighted sum of the data points in each group is taken as an initial control point. With the initial values (27), the LSPIA method (8) is employed to fit some synthesized or practical data sets. In nearly all of the examples, the convergence speeds of the LSPIA method with the two initial values $P^{(0)} = 0$ and $P^{(0)} = \frac{A^T Q}{\|A^T Q\|_2}$ are nearly the same, faster than that with the initial value $P^{(0)} = A^T Q$.

Specifically, Fig. 2 illustrates the curve plots of iteration count v.s. $\frac{\left\|A^T r^{(k)}\right\|_2}{\|A\|_f \left\|r^{(k)}\right\|_2}$ of LSPIA in fitting two practical data sets (Fig. 3), using uniform bi-cubic B-spline patches. Here, $r^{(k)} = Q - AP^{(k)}$, and $\|\cdot\|_f$ is the Frobenius norm of a matrix. As illustrated in Fig. 3, one of the data sets is the geological image *Calabria* (Fig. 3(a)) with resolution $17992 \times 22510$ and 2296014 data points, downloaded from *AimatShape*, and the other is the red-component of the image *flower* (Fig. 3(b)) with resolution $2048 \times 3072$. While the coefficient matrix for fitting the data set *Calabria* is singular, that for fitting the data set *flower* is nonsingular. From Figs. 2(a) and 2(b), we can see that the curve plots for the 2 initial values $P^{(0)} = 0$ and $P^{(0)} = \frac{A^T Q}{\|A^T Q\|_2}$ are nearly overlapped. They are better than the curve plot for the initial value $P^{(0)} = A^T Q$.



(a) (b)

Figure 2: Curve plots of iteration count v.s. $\frac{\left\|A^T r^{(k)}\right\|_2}{\|A\|_f \left\|r^{(k)}\right\|_2}$ of LSPIA in fitting the geological image *Calabria* (a) and the image *Flower* (b) with the 3 different initial values (27). Note that the curves for $P^{(0)} = \frac{A^T Q}{\|A^T Q\|_2}$ and $P^{(0)} = 0$ are nearly overlapped.

(a) (b)

Figure 3: Two data sets for checking the influence of the initial values. (a) The geological image *Calabria* with resolution $17992 \times 22510$ and 2296014 data points; (b) the image *Flower* with resolution $2048 \times 3072$.

### 3.3 Example

In this example, LSPIA is employed for least-squares fitting a practical data point set, i.e., the 43994 mesh vertices of a tetrahedral mesh model *balljoint* (Fig. 4(a)), by a tri-cubic trivariate B-spline solid (Fig. 4(b)), which has $30 \times 21 \times 33$ control points and uniform knot vectors along the three parametric directions with Bézier end conditions, uniformly distributed in the interval $[0, 1]$, respectively. The tetrahedral mesh vertices are parameterized with the method developed in Ref. [6], and the initial value of LSPIA iteration is taken as $P^{(0)} = 0$. In this example, the order of the matrix $A^T A$ (9) is $20790 \times 20790$, and its rank is 12628, so the matrix $A^T A$ (9) is singular. Although $A^T A$ is singular, LSPIA converges to a stable solution in 5.50 seconds (Fig. 4(b)) with least-squares fitting error $9.97 \times 10^{-5}$. Fig. 4(a) illustrates the input model, i.e., a tetrahedral mesh model with six segmented patches on its boundary, and Fig. 4(b) is the cut-away view of the generated trivariate B-spline solid.

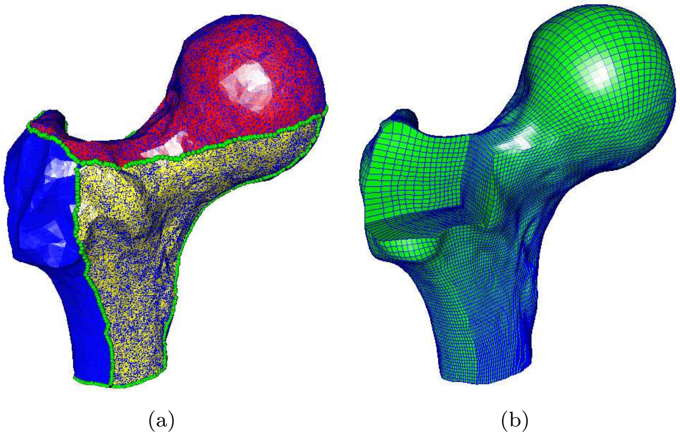


Figure 4: In least-squares fitting a tetrahedral mesh with six segmented patches on its boundary (a), though the matrix $A^T A$ (9) is singular, LSPIA converges to a stable solution, i.e., a trivariate B-spline solid (cut-away view) (b) in 5.50 seconds with least-squares fitting error $9.97 \times 10^{-5}$.

## 4 Conclusions

In this paper, we showed that the LSPIA method is convergent when the coefficient matrix $A^T A$ of a least-squares fitting system is singular. Moreover, when the diagonal elements of the diagonal matrix $\Lambda$ (8) are equal to each other, it converges to the M-P pseudo-inverse solution of the least-squares fitting to the given data points. Therefore, together with the previous result proved in Refs. [3, 4], LSPIA is convergent whatever the least-squares fitting system is singular or not. This property of LSPIA greatly extends the scope of the applications of LSPIA, especially in solving geometric problems in the big data processing, where singular linear systems frequently appear.

## References

[1] Pereyra V and Scherer G, Least squares scattered data fitting by truncated svds, *Applied Numerical Mathematics*, 2002, **40**(1): 73–86.

[2] Pereyra V and Scherer G, Large scale least squares scattered data fitting, *Applied Numerical Mathematics*, 2003, **44** (1): 225–239.

[3] Lin H and Zhang Z, An efficient method for fitting large data sets using T-splines, *SIAM Journal on Scientific Computing*, 2013, **35** (6): A3052–A3068.

[4] Deng C and Lin H, Progressive and iterative approximation for least squares B-spline curve and surface fitting, *Computer-Aided Design*, 2014, **47**: 32–44.

[5] Brandt C, Seidel H-P, and Hildebrandt K, Optimal spline approximation via $l_0$-minimization, *Computer Graphics Forum*, 2015, **34**; 617–626.

[6] Lin H, Jin S, Hu Q, and Liu Z, Constructing b-spline solids from tetrahedral meshes for isogeometric analysis, *Computer Aided Geometric Design*, 2015, **35**: 109–120.

[7] Lin H, Wang G, and Dong C, Constructing iterative non-uniform B-spline curve and surface to fit data points, *SCIENCE IN CHINA, Series F*, 2004, **47** (3): 315–331.

[8] Lin H, Bao H, and Wang G, Totally positive bases and progressive iteration approximation, *Computers and Mathematics with Applications*, **50** (3-4): 575–586.

[9] Lin H and Zhang Z, An extended iterative format for the progressive-iteration approximation, *Computers & Graphics*, 2011, **35**(5): 967–975.

[10] Shi L and Wang R, An iterative algorithm of nurbs interpolation and approximation, *Journal of Mathematical Research and Exposition*, 2006, **26**(4): 735–743.

[11] Cheng F, Fan F, Lai S, Huang C, Wang J, and Yong J, Loop subdivision surface based progressive interpolation, *Journal of Computer Science and Technology*, 2009, **24**(1): 39–46.

[12] Fan F, Cheng F, and Lai S, Subdivision based interpolation with shape control, *Computer Aided Design & Applications*, 2008, **5**(1-4): 539–547.

[13] Chen Z, Luo X, Tan L, Ye B, and Chen J, Progressive interpolation based on catmull-clark subdivision surfaces, *Computer Grahics Forum*, 2008, **27**(7): 1823–1827.

[14] Maekawa T, Matsumoto Y, and Namiki K, Interpolation by geometric algorithm, *Computer-Aided Design*, 2007, **39**: 313–323.

[15] Kineri Y, Wang M, Lin H, and Maekawa T, B-spline surface fitting by iterative geometric interpolation/approximation algorithms, *Computer-Aided Design*, 2012, **44**(7): 697–708.

[16] Yoshihara H, Yoshii T, Shibutani T, and Maekawa T, Topologically robust b-spline surface reconstruction from point clouds using level set methods and iterative geometric fitting algorithms, *Computer Aided Geometric Design*, 2012, **29**(7): 422–434.

[17] Okaniwa S, Nasri A, Lin H, Abbas A, Kineri Y, and Maekawa T, Uniform b-spline curve interpolation with prescribed tangent and curvature vectors, *IEEE Transactions on Visualization and Computer Graphics*, 2012, **18**(9): 1474–1487.

[18] Lin H, Qin Y, Liao H, and Xiong Y, Affine arithmetic-based b-spline surface intersection with gpu acceleration, *IEEE Transactions on Visualization and Computer Graphics*, 2014, **20**(2): 172–181.

[19] Sederberg T W, Cardon D L, Finnigan G T, North N S, Zheng J, and Lyche T, T-spline simplification and local refinement, *ACM Transactions on Graphics*, 2004, **23**: 276–283.

[20] Zhang Y, Wang W, and Hughes T J, Solid T-spline construction from boundary representations for genus-zero geometry, *Computer Methods in Applied Mechanics and Engineering*, 2012, **249**: 185–197.

[21] Horn R A and Johnson C R, *Matrix analysis (volume 1)*, Cambridge University Press, Cambridge, 1985.

[22] James M, The generalised inverse, *The Mathematical Gazette*, 1978, **62**(420): 109–114.