



Research article

On a weighted ALSPIA method for tensor-product B-spline surface fitting with missing data

Xianglan Ge and Lijuan Hu*

School of Mathematics and Data Science, Hunan University of Humanities, Science and Technology, Loudi 417000, China

* **Correspondence:** Email: 3464@huhst.edu.cn.

Abstract: This paper develops a tensor-structured asynchronous least-squares progressive-iterative approximation method (ALSPIA) for least-squares fitting and reconstruction of incomplete tensor-product B-spline surface data. A binary mask matrix is introduced to impose fitting constraints only on the observed entries. The resulting matrix-form iteration is algebraically equivalent to applying ALSPIA to the vectorized least-squares system containing only the observed data. However, it avoids explicitly constructing the row-restricted Kronecker-product collocation matrix and retains the original two-dimensional tensor-product representation. The convergence properties of the proposed tensor-structured implementation are analyzed for both singular and nonsingular weighted normal matrices. Numerical experiments with different missing-data patterns demonstrate the effectiveness of the method for surface reconstruction.

Keywords: B-spline; tensor-product; ALSPIA; weighted least-squares; missing-data reconstruction

Mathematics Subject Classification: 65D07, 65D10, 65F10

1. Introduction

B-spline curves and surfaces have been widely used in computer-aided geometric design, data fitting, and data reconstruction because of their local support, numerical stability, and flexible geometric representation [1, 2]. In particular, tensor-product B-spline surfaces provide a natural representation for two-dimensional gridded data and constitute an important tool for surface approximation and reconstruction [3]. When all grid data are available, the corresponding least-squares fitting problem can be expressed in a compact matrix form, allowing the separable structure of the basis functions in the two parameter directions to be exploited efficiently.

The fitting and reconstruction of B-spline curves and surfaces have been extensively studied. Classical least-squares B-spline fitting methods provide a stable approximation framework, but

they usually require solving linear systems. Moreover, for tensor-product spline fitting of large, scattered data sets, the formation of the corresponding least-squares system matrices can itself be computationally expensive. Merchel et al. [4] accelerated this assembly process by projecting the data parameters onto an auxiliary regular grid and applying sum factorization. To avoid solving large-scale systems explicitly, progressive-iterative approximation (PIA) and its variants have attracted considerable attention [5]. For large-scale fitting problems, Deng and Lin proposed the least-squares progressive-iterative approximation (LSPIA) method for B-spline curve and surface fitting [6]. Subsequently, various LSPIA-type methods have been developed to improve the efficiency, stability, and flexibility of the original framework, including constrained LSPIA [7], asynchronous LSPIA (ALSPIA) [8], full LSPIA for nonuniform rational B-spline (NURBS) fitting [9], and several randomized, momentum-accelerated, and local variants [10–14].

Most conventional tensor-product B-spline fitting formulations are developed for complete observations on a tensor-product parameter grid. When some entries are missing, the parameter grid itself may remain regular, but the separable least-squares structure associated with the complete set of observations is generally lost. A common strategy is to first estimate the missing entries using interpolation, imputation, or data-completion techniques and then apply a conventional surface-fitting method to the completed data. This two-stage strategy makes the final fitting result dependent on the accuracy of the preliminary completion step. Among these approaches, low-rank matrix and tensor completion methods constitute an important class of techniques for recovering incomplete structured data. These methods infer missing entries by exploiting assumed low-rank structures. Representative studies include tensor completion for visual-data recovery [15], low- n -rank tensor recovery through convex optimization [16], and low-rank tensor completion based on Riemannian optimization [17]. Such methods have been widely applied to the recovery of images, videos, and other multiway data. Nevertheless, they rely primarily on low-rank assumptions and do not directly incorporate the tensor-product B-spline representation used in gridded surface fitting. Low-rank approximation has also been introduced to improve the computational efficiency of tensor-product spline surface fitting for complete gridded data [3]. However, this approach is not specifically designed to model general missing-data patterns or to impose fitting constraints only on the observed entries.

Instead of estimating the missing entries in advance, one may formulate the fitting problem directly using only the observed data. An important property of LSPIA-type methods is their applicability to singular least-squares fitting systems. Lin et al. [18] established the convergence of LSPIA for singular least-squares systems, and the numerical results of Wu and Liu [8] demonstrated that ALSPIA remains applicable to data sets containing holes. Therefore, the presence of missing data does not, by itself, prevent LSPIA or ALSPIA from converging. The explicit treatment of missing data in these studies is mainly presented in the curve-fitting setting. For B-spline curve fitting, missing data points can be handled by removing the corresponding equations from a single collocation system, after which LSPIA or ALSPIA can be applied directly. However, these studies do not provide an explicit structure-preserving matrix formulation for arbitrary missing entries in tensor-product surface data.

For tensor-product surface fitting with complete gridded observations, the separable basis functions in the two parameter directions lead to a compact matrix representation. General missing-data patterns select an irregular subset of the tensor grid and therefore destroy the separable structure of the complete observation constraints. One possible treatment is to flatten the surface data and formulate the problem as a conventional vector least-squares system. The vector-form LSPIA framework can then be applied

in essentially the same manner as in the curve case. However, this treatment may require the explicit construction or storage of a large-scale row-restricted coefficient matrix involving the Kronecker product, and the original matrix-form tensor-product structure is no longer directly available.

The main contribution in this work is therefore not whether LSPIA or ALSPIA can converge for data sets containing holes but rather how to implement the observed-data ALSPIA iteration while preserving the tensor-product matrix structure. To this end, this paper develops a tensor-structured ALSPIA formulation for incomplete tensor-product B-spline surface data. A binary mask matrix is used to impose fitting constraints only on the observed entries. The resulting formulation is algebraically equivalent to applying ALSPIA to the vectorized least-squares system containing only the observed data, but it evaluates the residual correction directly in matrix form. Consequently, it avoids the explicit construction of the row-selection matrix and the Kronecker-product coefficient matrix and can accommodate general missing-data patterns. The convergence properties of the resulting tensor-structured iteration are also analyzed under the corresponding weighted least-squares formulation.

The remainder of this paper is organized as follows. Section 2 introduces the preliminaries. Section 3 presents the ALSPIA-based method for incomplete-data fitting and provides its convergence analysis. Section 4 reports numerical experiments demonstrating the effectiveness of the proposed method. Finally, Section 5 concludes the paper and outlines possible future research directions.

2. Preliminaries

2.1. Least-squares fitting of tensor-product B-spline surfaces

Given a tensor-product grid of data points

$$\{q_{ij} \in \mathbb{R}^3 : i = 0, 1, \dots, m_1, j = 0, 1, \dots, m_2\},$$

let

$$0 \leq u_0 < u_1 < \dots < u_{m_1} \leq 1 \text{ and } 0 \leq v_0 < v_1 < \dots < v_{m_2} \leq 1$$

be the parameter values associated with the data points in the two parameter directions.

Let $\{N_h^d(u)\}_{h=0}^{n_1}$ and $\{N_l^d(v)\}_{l=0}^{n_2}$ be the degree- d B-spline basis functions defined on the open uniform knot vectors

$$\mathcal{U} = (\xi_0, \xi_1, \dots, \xi_{n_1+d+1}) \text{ and } \mathcal{V} = (\eta_0, \eta_1, \dots, \eta_{n_2+d+1}),$$

respectively, where

$$\xi_r = \begin{cases} 0, & 0 \leq r \leq d, \\ \frac{r-d}{n_1-d+1}, & d+1 \leq r \leq n_1, \\ 1, & n_1+1 \leq r \leq n_1+d+1, \end{cases}$$

and

$$\eta_r = \begin{cases} 0, & 0 \leq r \leq d, \\ \frac{r-d}{n_2-d+1}, & d+1 \leq r \leq n_2, \\ 1, & n_2+1 \leq r \leq n_2+d+1. \end{cases}$$

The tensor-product B-spline surface is then defined by

$$\mathcal{S}(u, v) = \sum_{h=0}^{n_1} \sum_{l=0}^{n_2} N_h^d(u) N_l^d(v) p_{hl}, \quad (2.1)$$

where p_{hl} are the control points to be determined.

To make the surface approximate the given data points in the least-squares sense, the following optimization problem is obtained:

$$\min_{p_{hl}} \sum_{i=0}^{m_1} \sum_{j=0}^{m_2} \|q_{ij} - \mathcal{S}(u_i, v_j)\|^2. \quad (2.2)$$

For simplicity, the following matrix formulation is presented for a single coordinate component. For surface data in $\mathbb{R}^3$, the same formulation is applied independently to the three coordinate components. The observed target data and the control points are arranged in matrix form as

$$\mathbf{Q} = [q_{ij}] \in \mathbb{R}^{(m_1+1) \times (m_2+1)}, \quad \mathbf{P} = [p_{hl}] \in \mathbb{R}^{(n_1+1) \times (n_2+1)}.$$

Then, the least-squares problem can be equivalently expressed in the following matrix form:

$$\min_{\mathbf{P}} \|\mathbf{A}\mathbf{P}\mathbf{B}^T - \mathbf{Q}\|_F^2, \quad (2.3)$$

where $\|\cdot\|_F$ denotes the Frobenius norm. Specifically, for a matrix $\mathbf{M} = [m_{ij}] \in \mathbb{R}^{m \times n}$, it is defined by

$$\|\mathbf{M}\|_F = \left(\sum_{i=1}^m \sum_{j=1}^n |m_{ij}|^2 \right)^{1/2} = (\text{trace}(\mathbf{M}^T \mathbf{M}))^{1/2}.$$

The matrices

$$\mathbf{A} = \begin{bmatrix} N_0^d(u_0) & N_1^d(u_0) & \cdots & N_{n_1}^d(u_0) \\ N_0^d(u_1) & N_1^d(u_1) & \cdots & N_{n_1}^d(u_1) \\ \vdots & \vdots & \ddots & \vdots \\ N_0^d(u_{m_1}) & N_1^d(u_{m_1}) & \cdots & N_{n_1}^d(u_{m_1}) \end{bmatrix} \text{ and } \mathbf{B} = \begin{bmatrix} N_0^d(v_0) & N_1^d(v_0) & \cdots & N_{n_2}^d(v_0) \\ N_0^d(v_1) & N_1^d(v_1) & \cdots & N_{n_2}^d(v_1) \\ \vdots & \vdots & \ddots & \vdots \\ N_0^d(v_{m_2}) & N_1^d(v_{m_2}) & \cdots & N_{n_2}^d(v_{m_2}) \end{bmatrix}$$

are the collocation matrices of the basis functions defined at the parameter values. Correspondingly, its first-order optimality condition of (2.3) can be written in the following matrix equation form:

$$\mathbf{A}^T (\mathbf{A}\mathbf{P}\mathbf{B}^T - \mathbf{Q}) \mathbf{B} = 0, \quad (2.4)$$

which constitutes the basic algebraic model for least-squares fitting of tensor-product B-spline surfaces.

Remark 2.1. Let $\mathbf{M} = [m_{ij}] \in \mathbb{R}^{m \times n}$. Its columnwise vectorization is defined as

$$\text{vec}(\mathbf{M}) = (m_{11}, m_{21}, \dots, m_{m1}, m_{12}, m_{22}, \dots, m_{m2}, \dots, m_{1n}, \dots, m_{mn})^T.$$

Then, using the Kronecker product identity

$$\text{vec}(\mathbf{A}\mathbf{P}\mathbf{B}^T) = (\mathbf{B} \otimes \mathbf{A})\text{vec}(\mathbf{P}),$$

the least-squares problem (2.3) can be rewritten as

$$\min_{\mathbf{x}} \|(\mathbf{B} \otimes \mathbf{A})\mathbf{x} - \text{vec}(\mathbf{Q})\|_2^2, \quad \mathbf{x} = \text{vec}(\mathbf{P}).$$

The corresponding normal equation is

$$(\mathbf{B} \otimes \mathbf{A})^T (\mathbf{B} \otimes \mathbf{A})\mathbf{x} = (\mathbf{B} \otimes \mathbf{A})^T \text{vec}(\mathbf{Q}).$$

2.2. The ALSPIA method for tensor-product B-spline surfaces

Following [8], we briefly review the ALSPIA method for the least-squares fitting of complete tensor-product surface data. This formulation also provides the basis for the missing-data reconstruction method developed in the next section.

The ALSPIA iteration is initialized with the control points

$$\{p_{hl}^{(0)} : h = 0, \dots, n_1, l = 0, \dots, n_2\},$$

which define the initial tensor-product B-spline surface

$$\mathcal{S}^{(0)}(u, v) = \sum_{h=0}^{n_1} \sum_{l=0}^{n_2} N_h^d(u) N_l^d(v) p_{hl}^{(0)}.$$

At the k th iteration, let

$$\{p_{hl}^{(k)} : h = 0, \dots, n_1, l = 0, \dots, n_2\}$$

denote the current control points. The corresponding tensor-product B-spline surface is given by

$$\mathcal{S}^{(k)}(u, v) = \sum_{h=0}^{n_1} \sum_{l=0}^{n_2} N_h^d(u) N_l^d(v) p_{hl}^{(k)}.$$

The fitting residual at the parameter point (u_i, v_j) is defined as

$$r_{ij}^{(k)} = q_{ij} - \mathcal{S}^{(k)}(u_i, v_j), \quad i = 0, \dots, m_1, \quad j = 0, \dots, m_2.$$

The correction associated with the control point $p_{hl}^{(k)}$ is then computed by

$$\Delta_{hl}^{(k)} = \sum_{i=0}^{m_1} \sum_{j=0}^{m_2} N_h^d(u_i) N_l^d(v_j) r_{ij}^{(k)}, \quad h = 0, \dots, n_1, \quad l = 0, \dots, n_2.$$

Accordingly, the control points are updated according to

$$p_{hl}^{(k+1)} = p_{hl}^{(k)} + \mu_k \Delta_{hl}^{(k)}, \quad h = 0, \dots, n_1, \quad l = 0, \dots, n_2, \quad (2.5)$$

where $\mu_k > 0$ is the acceleration parameter at the k th iteration.

According to (2.5), the update of the control points is obtained by weighted accumulation of the residuals from all data points according to the basis functions. This method inherits the local support property of B-spline basis functions. When a large error occurs at a certain data point, only a small number of associated control points are significantly affected, which ensures good local adjustment capability and numerical stability.

To express the iteration more compactly, arranging all control points into the matrix

$$\mathbf{P}^{(k)} = [p_{hl}^{(k)}] \in \mathbb{R}^{(n_1+1) \times (n_2+1)},$$

the ALSPIA update scheme (2.5) for the control points can be written in the matrix form

$$\mathbf{P}^{(k+1)} = \mathbf{P}^{(k)} + \mu_k \mathbf{A}^\top (\mathbf{Q} - \mathbf{A} \mathbf{P}^{(k)} \mathbf{B}^\top) \mathbf{B}. \quad (2.6)$$

Thus, ALSPIA can be interpreted as a nonstationary Richardson iteration for the tensor-product least-squares problem. When $\mu_k \equiv \mu$ is fixed, the iteration reduces to the standard LSPIA scheme; ALSPIA employs a nonstationary parameter sequence $\{\mu_k\}$ to accelerate convergence.

The matrix formulation (2.6) requires only multiplications involving the two collocation matrices $\mathbf{A}$ and $\mathbf{B}$. It neither requires the direct solution of the normal equations nor the explicit construction of the Kronecker-product matrix $\mathbf{B} \otimes \mathbf{A}$. Consequently, it preserves the tensor-product structure and provides a convenient basis for the missing-data formulation introduced in the next section.

3. Missing data fitting of tensor-product B-splines

In this section, we develop a structure-preserving matrix formulation of the observed-data least-squares problem for incomplete tensor-product B-spline surfaces. The binary mask provides a convenient representation of the observed entries and allows the corresponding ALSPIA correction to be evaluated without explicitly forming the row-restricted Kronecker-product collocation matrix.

3.1. Missing data modeling

To effectively utilize the original structural information in the presence of missing data, it is necessary to introduce a weighting mechanism to model the observed data while preserving the tensor-product form. For this purpose, let

$$\Omega = \{(i, j) \mid q_{ij} \text{ is observed}\}$$

denote the index set of known data, and introduce the weight matrix $\mathbf{W} = [w_{ij}] \in \mathbb{R}^{(m_1+1) \times (m_2+1)}$:

$$w_{ij} = \begin{cases} 1, & (i, j) \in \Omega, \\ 0, & \text{otherwise.} \end{cases} \quad (3.1)$$

Based on the above weighting mechanism, the original least-squares fitting problem is generalized into the following weighted least-squares model:

$$\min_{\mathbf{P}} \sum_{i=0}^{m_1} \sum_{j=0}^{m_2} w_{ij} \|\mathcal{S}(u_i, v_j) - q_{ij}\|^2, \quad (3.2)$$

where $\mathcal{S}(u, v)$ is the tensor-product B-spline surface.

Using the matrix representation, the problem can be equivalently written in the following weighted matrix form:

$$\min_{\mathbf{P}} \|\mathbf{W} \odot (\mathbf{A}\mathbf{P}\mathbf{B}^\top - \mathbf{Q})\|_F^2, \quad (3.3)$$

where $\odot$ denotes the Hadamard elementwise product.

Remark 3.1. Model (3.3) imposes constraints only on the observed positions while preserving the tensor-product structure, thereby providing a natural modeling approach for missing data. When $\mathbf{W}$ is an all-one matrix, this model reduces to the standard tensor-product B-spline least-squares fitting problem.

3.2. Tensor-structured iterative algorithm based on ALSPIA

For the weighted least-squares model established in the previous subsection, we construct an ALSPIA-based iterative method while preserving the tensor-product structure.

First, we initialize the control points $p_{hl}^{(0)}$ and construct the initial B-spline surface

$$\mathcal{S}^{(0)}(u, v) = \sum_{h=0}^{n_1} \sum_{l=0}^{n_2} N_h^d(u) N_l^d(v) p_{hl}^{(0)}.$$

Recursively, denote the B-spline surface corresponding to the k th iteration by

$$\mathcal{S}^{(k)}(u, v) = \sum_{h=0}^{n_1} \sum_{l=0}^{n_2} N_h^d(u) N_l^d(v) p_{hl}^{(k)}.$$

Define the residual vector as

$$r_{ij}^{(k)} = q_{ij} - \mathcal{S}^{(k)}(u_i, v_j), \quad i = 0, 1, \dots, m_1, j = 0, 1, \dots, m_2.$$

When missing data are considered, corrections are performed only at the observed positions. Hence, the weighted residual vector is given by

$$\tilde{r}_{ij}^{(k)} = w_{ij} \cdot r_{ij}^{(k)}.$$

Furthermore, the adjustment vector of the control points is defined as

$$\Delta_{hl}^{(k)} = \sum_{i=0}^{m_1} \sum_{j=0}^{m_2} N_h^d(u_i) N_l^d(v_j) \tilde{r}_{ij}^{(k)} \quad h = 0, \dots, n_1, \quad l = 0, \dots, n_2.$$

Thus, for $h = 0, \dots, n_1, \quad l = 0, \dots, n_2$, the ALSPIA update scheme for the control points is

$$p_{hl}^{(k+1)} = p_{hl}^{(k)} + \mu_k \Delta_{hl}^{(k)}, \quad (3.4)$$

where $\mu_k > 0$ is the k th relaxation parameter. Its specific Chebyshev-type choice will be introduced in the convergence analysis.

The above update scheme shows that, at each iteration, the control points are corrected through a weighted accumulation of the residuals. Owing to the local support property of the B-spline basis functions, each residual affects only the control points associated with the corresponding local basis functions. Consequently, the fitted surface is progressively adjusted toward the observed data.

Define the residual matrix at the k th iteration by

$$\mathbf{R}^{(k)} = \mathbf{Q} - \mathbf{A} \mathbf{P}^{(k)} \mathbf{B}^\top.$$

Then, the iterative process (3.4) can be written in matrix form as

$$\mathbf{P}^{(k+1)} = \mathbf{P}^{(k)} + \mu_k \mathbf{A}^\top (\mathbf{W} \odot \mathbf{R}^{(k)}) \mathbf{B}, \quad (3.5)$$

where $\odot$ denotes the Hadamard elementwise product. This form evaluates the observed-data residual correction directly through matrix operations involving $\mathbf{A}$, $\mathbf{B}$, and $\mathbf{W}$. It is algebraically equivalent to the vectorized observed-row ALSPIA iteration but avoids explicitly constructing the Kronecker-product matrix $\mathbf{B} \otimes \mathbf{A}$.

Remark 3.2. When $\mathbf{W}$ is an all-one matrix, Iteration (3.5) reduces to the conventional ALSPIA scheme (2.6) for complete tensor-product data. For incomplete data, it provides a structure-preserving matrix implementation of ALSPIA applied to the observed-data least-squares system.

Remark 3.3. Li et al. [19] introduced data weights into LSPIA curve fitting to satisfy different interpolation and approximation requirements. In their method, the weights control the relative importance of the data points. In our work, the entries of $\mathbf{W}$ are binary and represent data availability: An observed entry has weight one, whereas a missing entry has weight zero.

More importantly, the present work considers incomplete tensor-product surface data. The binary mask allows the observed-row ALSPIA correction to be evaluated in two-dimensional matrix form, without explicitly constructing the large row-restricted Kronecker-product coefficient matrix.

3.3. Convergence analysis

In this subsection, we analyze the convergence of the proposed tensor-structured ALSPIA iteration for incomplete data. We first reformulate the weighted least-squares problem and the matrix iteration (3.5) in vector form. We then consider separately the singular and nonsingular cases of the resulting weighted normal matrix.

Let

$$\mathbf{x}^{(k)} = \text{vec}(\mathbf{P}^{(k)}), \quad \mathbf{q} = \text{vec}(\mathbf{Q}),$$

and define

$$\mathbf{K} = \mathbf{B} \otimes \mathbf{A}, \quad \mathbf{D} = \text{diag}(\text{vec}(\mathbf{W})).$$

Because $\mathbf{W}$ is binary, $\mathbf{D}$ is a symmetric diagonal projection matrix satisfying $\mathbf{D}^\top \mathbf{D} = \mathbf{D}^2 = \mathbf{D}$. Using the vectorization identity [20]

$$\text{vec}(\mathbf{A}\mathbf{P}^{(k)}\mathbf{B}^\top) = (\mathbf{B} \otimes \mathbf{A}) \text{vec}(\mathbf{P}^{(k)}) = \mathbf{K}\mathbf{x}^{(k)},$$

together with

$$\text{vec}(\mathbf{W} \odot \mathbf{R}^{(k)}) = \mathbf{D} \text{vec}(\mathbf{R}^{(k)}),$$

the weighted least-squares problem (3.3) can be written as

$$\min_{\mathbf{x}} \frac{1}{2} (\mathbf{K}\mathbf{x} - \mathbf{q})^\top \mathbf{D} (\mathbf{K}\mathbf{x} - \mathbf{q}) = \min_{\mathbf{x}} \frac{1}{2} \|\mathbf{D}^{1/2}(\mathbf{K}\mathbf{x} - \mathbf{q})\|_2^2. \quad (3.6)$$

The normal equation associated with (3.6) is

$$\mathbf{H}\mathbf{x} = \mathbf{g}, \quad (3.7)$$

where

$$\mathbf{H} = \mathbf{K}^\top \mathbf{D} \mathbf{K} = (\mathbf{B} \otimes \mathbf{A})^\top \mathbf{D} (\mathbf{B} \otimes \mathbf{A}), \quad \mathbf{g} = \mathbf{K}^\top \mathbf{D} \mathbf{q}.$$

By vectorizing (3.5), the proposed iteration becomes

$$\begin{aligned} \mathbf{x}^{(k+1)} &= \mathbf{x}^{(k)} + \mu_k \mathbf{K}^\top \mathbf{D} (\mathbf{q} - \mathbf{K}\mathbf{x}^{(k)}) \\ &= \mathbf{x}^{(k)} - \mu_k (\mathbf{H}\mathbf{x}^{(k)} - \mathbf{g}). \end{aligned} \quad (3.8)$$

Thus, the proposed tensor-structured ALSPIA iteration is equivalent to a nonstationary Richardson iteration applied to the weighted normal equation (3.7). The matrix form (3.5) is used in the actual implementation, whereas (3.8) is introduced only for the convergence analysis.

Let $\mathbf{x}^*$ be a solution of $\mathbf{H}\mathbf{x} = \mathbf{g}$, and define

$$\mathbf{e}^{(k)} = \mathbf{x}^{(k)} - \mathbf{x}^*.$$

Subtracting $\mathbf{H}\mathbf{x}^* = \mathbf{g}$ from (3.8) yields

$$\mathbf{e}^{(k+1)} = (\mathbf{I} - \mu_k \mathbf{H})\mathbf{e}^{(k)}.$$

Therefore, after s iterations,

$$\mathbf{e}^{(s)} = \prod_{\ell=0}^{s-1} (\mathbf{I} - \mu_\ell \mathbf{H})\mathbf{e}^{(0)} = f_s(\mathbf{H})\mathbf{e}^{(0)}, \quad (3.9)$$

where

$$f_s(t) = \prod_{\ell=0}^{s-1} (1 - \mu_\ell t).$$

Consequently, the convergence behavior of the proposed iteration is determined by the values of $f_s(t)$ on the spectrum of $\mathbf{H}$.

Lemma 3.1. *Let*

$$\mathbf{H} = \mathbf{K}^\top \mathbf{D} \mathbf{K}, \quad \mathbf{K} = \mathbf{B} \otimes \mathbf{A}, \quad \mathbf{D} = \text{diag}(\text{vec}(\mathbf{W})),$$

where $\mathbf{W}$ is a binary mask matrix. Let $\mathbf{K}_\Omega$ denote the submatrix of $\mathbf{K}$ formed by the rows corresponding to the observed entries. Then,

$$\mathbf{H} = \mathbf{K}_\Omega^\top \mathbf{K}_\Omega.$$

Consequently, $\mathbf{H}$ is symmetric positive semidefinite. Moreover, $\mathbf{H}$ is positive definite if and only if $\mathbf{K}_\Omega$ has full column rank.

Proof. Because the diagonal entries of $\mathbf{D}$ are either zero or one, multiplication by $\mathbf{D}$ removes the contributions corresponding to the missing entries. It follows that

$$\mathbf{K}^\top \mathbf{D} \mathbf{K} = \mathbf{K}_\Omega^\top \mathbf{K}_\Omega.$$

For any vector $\mathbf{z}$, we have

$$\mathbf{z}^\top \mathbf{H} \mathbf{z} = \mathbf{z}^\top \mathbf{K}_\Omega^\top \mathbf{K}_\Omega \mathbf{z} = \|\mathbf{K}_\Omega \mathbf{z}\|_2^2 \geq 0.$$

Hence, $\mathbf{H}$ is symmetric positive semidefinite. Furthermore,

$$\mathbf{z}^\top \mathbf{H} \mathbf{z} = 0 \iff \mathbf{K}_\Omega \mathbf{z} = \mathbf{0}.$$

Therefore, $\mathbf{H}$ is positive definite if and only if

$$\ker(\mathbf{K}_\Omega) = \{0\},$$

which is equivalent to $\mathbf{K}_\Omega$ having full column rank. □

Lemma 3.1 shows that missing entries do not necessarily make the weighted normal matrix singular. Singularity occurs only when the observed entries do not provide sufficient independent constraints to determine all the control points.

Moreover, because $\|\mathbf{D}\|_2 \leq 1$, we have

$$\lambda_{\max}(\mathbf{H}) \leq \|\mathbf{K}\|_2^2 = \|\mathbf{B} \otimes \mathbf{A}\|_2^2 = \|\mathbf{A}\|_2^2 \|\mathbf{B}\|_2^2.$$

Thus, an upper bound for the largest eigenvalue of $\mathbf{H}$ can be chosen as

$$\beta = \|\mathbf{A}\|_2^2 \|\mathbf{B}\|_2^2.$$

We first consider the case in which $\mathbf{H}$ is symmetric positive semidefinite and may be singular. Following the Chebyshev-type acceleration strategy in [8], let $s \in \mathbb{N}^+$ be a prescribed integer that determines the degree of the Chebyshev polynomial and the number of acceleration parameters. In [8], s is referred to as the prescribed maximum iteration number. For a fixed s , define

$$r_s = \cos\left(\frac{(2s+1)\pi}{2s+2}\right)$$

and

$$c_\ell = \cos\left(\frac{(2\ell+1)\pi}{2s+2}\right), \quad \ell = 0, 1, \dots, s-1.$$

The corresponding set of s Chebyshev-type acceleration parameters is given by

$$\mu_\ell = \frac{1 - r_s}{\beta(c_\ell - r_s)}, \quad \ell = 0, 1, \dots, s-1, \quad (3.10)$$

where

$$\beta \geq \lambda_{\max}(\mathbf{H}) > 0.$$

Here, the parameter μ_k is used at the k th iteration for $k = 0, 1, \dots, s-1$. Therefore, s should be no smaller than the number of iterations to be performed. The iteration may terminate before reaching s when the prescribed stopping criterion is satisfied.

Theorem 3.1. *Let the sequence $\{\mathbf{x}^{(k)}\}$ be generated by the proposed weighted tensor-structured ALSPIA iteration (3.8). Suppose that $\mathbf{H}$ is symmetric positive semidefinite and that the parameters μ_ℓ are given by (3.10). Then,*

$$\|\mathbf{H}\mathbf{e}^{(s)}\|_2 \leq \frac{\beta\pi}{2(s+1)^2} \|\mathbf{e}^{(0)}\|_2. \quad (3.11)$$

Moreover, for every positive eigenvalue $\lambda_i(\mathbf{H}) > 0$,

$$f_s(\lambda_i(\mathbf{H})) \longrightarrow 0 \text{ as } s \longrightarrow \infty,$$

whereas

$$f_s(0) = 1.$$

Consequently, the error components associated with the positive eigenvalues of $\mathbf{H}$ converge to zero, whereas the component in $\ker(\mathbf{H})$ remains unchanged.

Proof. Define

$$\widetilde{f}_{s+1}(t) = tf_s(t) = t \prod_{\ell=0}^{s-1} (1 - \mu_\ell t).$$

Let T_{s+1} denote the Chebyshev polynomial of the first kind. Because r_s and c_ℓ , $\ell = 0, \dots, s-1$, are the $s+1$ roots of T_{s+1} , the parameters in (3.10) give

$$\widetilde{f}_{s+1}(t) = \frac{\beta T_{s+1} \left(r_s + \frac{1-r_s}{\beta} t \right)}{(1-r_s)T'_{s+1}(r_s)}.$$

For $t \in [0, \beta]$,

$$r_s + \frac{1-r_s}{\beta} t \in [r_s, 1] \subseteq [-1, 1].$$

Hence,

$$\left| T_{s+1} \left(r_s + \frac{1-r_s}{\beta} t \right) \right| \leq 1.$$

Let

$$\theta_s = \frac{(2s+1)\pi}{2s+2} = \pi - \frac{\pi}{2s+2}.$$

Because $r_s = \cos \theta_s$, the derivative identity for Chebyshev polynomials gives

$$|T'_{s+1}(r_s)| = \frac{s+1}{\sin\left(\frac{\pi}{2s+2}\right)}.$$

Moreover,

$$1 - r_s = 1 + \cos\left(\frac{\pi}{2s+2}\right).$$

It follows that

$$\max_{0 \leq t \leq \beta} |\widetilde{f}_{s+1}(t)| \leq \frac{\beta}{(1-r_s)|T'_{s+1}(r_s)|} = \frac{\beta}{s+1} \tan\left(\frac{\pi}{4s+4}\right) \leq \frac{\beta\pi}{2(s+1)^2}.$$

From (3.9), we have

$$\mathbf{H}\mathbf{e}^{(s)} = \mathbf{H}f_s(\mathbf{H})\mathbf{e}^{(0)} = \widetilde{f}_{s+1}(\mathbf{H})\mathbf{e}^{(0)}.$$

Because the spectrum of $\mathbf{H}$ is contained in $[0, \beta]$,

$$\|\mathbf{H}\mathbf{e}^{(s)}\|_2 \leq \max_{0 \leq t \leq \beta} |\widetilde{f}_{s+1}(t)| \|\mathbf{e}^{(0)}\|_2,$$

which proves (3.11).

For every positive eigenvalue $\lambda_i(\mathbf{H}) > 0$,

$$|f_s(\lambda_i(\mathbf{H}))| = \frac{|\widetilde{f}_{s+1}(\lambda_i(\mathbf{H}))|}{\lambda_i(\mathbf{H})} \leq \frac{\beta\pi}{2\lambda_i(\mathbf{H})(s+1)^2} \rightarrow 0.$$

For a zero eigenvalue,

$$f_s(0) = 1.$$

Therefore, the error components corresponding to the positive eigenvalues vanish, whereas the component in $\ker(\mathbf{H})$ remains unchanged. $\square$

We next consider the case in which the observed entries provide sufficient independent constraints so that $\mathbf{K}_\Omega$ has full column rank. According to Lemma 3.1, $\mathbf{H}$ is then symmetric positive definite, and the weighted least-squares problem has a unique control-point solution.

Theorem 3.2. *Let the sequence $\{\mathbf{x}^{(k)}\}$ be generated by the proposed weighted tensor-structured ALSPIA iteration (3.8). Suppose that $\mathbf{K}_\Omega$ has full column rank and that*

$$0 < \alpha \leq \lambda_i(\mathbf{H}) \leq \beta,$$

where

$$\alpha \leq \lambda_{\min}(\mathbf{H}), \quad \beta \geq \lambda_{\max}(\mathbf{H}), \quad \alpha < \beta.$$

For a prescribed number s of iterations, choose

$$\mu_\ell = \frac{2}{(\beta + \alpha) + (\beta - \alpha) \cos\left(\frac{(2\ell + 1)\pi}{2s}\right)}, \quad \ell = 0, 1, \dots, s - 1. \quad (3.12)$$

Then,

$$\|\mathbf{x}^{(s)} - \mathbf{x}^*\|_2 \leq 2 \left(\frac{\sqrt{\kappa} - 1}{\sqrt{\kappa} + 1} \right)^s \|\mathbf{x}^{(0)} - \mathbf{x}^*\|_2, \quad (3.13)$$

where $\kappa = \beta/\alpha$, and $\mathbf{x}^*$ is the unique solution of $\mathbf{H}\mathbf{x} = \mathbf{g}$. Equivalently,

$$\|\mathbf{P}^{(s)} - \mathbf{P}^*\|_F \leq 2 \left(\frac{\sqrt{\kappa} - 1}{\sqrt{\kappa} + 1} \right)^s \|\mathbf{P}^{(0)} - \mathbf{P}^*\|_F.$$

Consequently, the proposed weighted tensor-structured ALSPIA method converges linearly to the unique weighted least-squares solution.

Proof. Because $\mathbf{K}_\Omega$ has full column rank, Lemma 3.1 implies that

$$\mathbf{H} = \mathbf{K}_\Omega^\top \mathbf{K}_\Omega$$

is symmetric positive definite. Therefore, the normal equation $\mathbf{H}\mathbf{x} = \mathbf{g}$ has a unique solution $\mathbf{x}^*$.

From (3.9),

$$\mathbf{e}^{(s)} = f_s(\mathbf{H})\mathbf{e}^{(0)}, \quad f_s(t) = \prod_{\ell=0}^{s-1} (1 - \mu_\ell t).$$

For the parameters in (3.12), the roots of $f_s(t)$ are

$$\frac{\alpha + \beta}{2} + \frac{\beta - \alpha}{2} \cos\left(\frac{(2\ell + 1)\pi}{2s}\right), \quad \ell = 0, 1, \dots, s - 1.$$

Hence,

$$f_s(t) = \frac{T_s\left(\frac{2t - (\alpha + \beta)}{\beta - \alpha}\right)}{T_s\left(-\frac{\alpha + \beta}{\beta - \alpha}\right)},$$

where T_s denotes the Chebyshev polynomial of the first kind.

For $t \in [\alpha, \beta]$,

$$\frac{2t - (\alpha + \beta)}{\beta - \alpha} \in [-1, 1],$$

and therefore,

$$\left| T_s \left(\frac{2t - (\alpha + \beta)}{\beta - \alpha} \right) \right| \leq 1.$$

Let

$$z = \frac{\alpha + \beta}{\beta - \alpha} > 1.$$

Using

$$T_s(z) = \frac{1}{2} \left[\left(z + \sqrt{z^2 - 1} \right)^s + \left(z - \sqrt{z^2 - 1} \right)^s \right],$$

we obtain

$$\frac{1}{|T_s(z)|} \leq 2 \left(z + \sqrt{z^2 - 1} \right)^{-s}.$$

A direct calculation gives

$$z + \sqrt{z^2 - 1} = \frac{\sqrt{\beta} + \sqrt{\alpha}}{\sqrt{\beta} - \sqrt{\alpha}}.$$

It follows that

$$\max_{t \in [\alpha, \beta]} |f_s(t)| \leq 2 \left(\frac{\sqrt{\beta} - \sqrt{\alpha}}{\sqrt{\beta} + \sqrt{\alpha}} \right)^s = 2 \left(\frac{\sqrt{\kappa} - 1}{\sqrt{\kappa} + 1} \right)^s.$$

Because $\mathbf{H}$ is symmetric positive definite, and its spectrum is contained in $[\alpha, \beta]$, the spectral theorem gives

$$\|f_s(\mathbf{H})\|_2 = \max_i |f_s(\lambda_i(\mathbf{H}))|.$$

Therefore,

$$\|\mathbf{e}^{(s)}\|_2 \leq \|f_s(\mathbf{H})\|_2 \|\mathbf{e}^{(0)}\|_2 \leq 2 \left(\frac{\sqrt{\kappa} - 1}{\sqrt{\kappa} + 1} \right)^s \|\mathbf{e}^{(0)}\|_2.$$

Finally, because $\mathbf{x}^{(k)} = \text{vec}(\mathbf{P}^{(k)})$, we have

$$\|\mathbf{x}^{(k)} - \mathbf{x}^*\|_2 = \|\mathbf{P}^{(k)} - \mathbf{P}^*\|_F.$$

This completes the proof. □

Remark 3.4. If $\alpha = \beta$, then $\mathbf{H} = \alpha \mathbf{I}$, and the choice $\mu_0 = 1/\alpha$ gives the exact solution in one iteration. Moreover, when $\mathbf{W}$ is an all-one matrix, $\mathbf{D} = \mathbf{I}$, and the proposed method reduces to the conventional ALSPIA method for complete tensor-product data.

4. Numerical experiments

To evaluate the effectiveness of the proposed tensor-structured ALSPIA implementation, numerical experiments are conducted under several missing-data patterns, including mixed local holes and strip-shaped missing regions.

All numerical experiments were implemented in MATLAB R2022a and performed on a desktop computer equipped with a 13th Gen Intel(R) Core(TM) i7-13700F processor running at 2.10 GHz and 32 GB of RAM. The computational time was measured using the MATLAB `tic` and `toc` functions, and the elapsed time of a single run is reported.

4.1. Evaluation metrics

Assume that the original complete surface data and the reconstructed surface data are denoted by

$$\mathbf{Q} = \{q_{ij}\}_{i=0, j=0}^{m_1, m_2}, \quad \widehat{\mathbf{Q}} = \{\widehat{q}_{ij}\}_{i=0, j=0}^{m_1, m_2},$$

respectively, where $\widehat{q}_{ij}$ is obtained from the fitted tensor-product B-spline surface.

Let

$$\Omega_{\text{miss}} = \{(i, j) \mid q_{ij} \text{ is missing}\}$$

be the index set of missing data points. The total number of sampled points is

$$N = (m_1 + 1)(m_2 + 1),$$

and the number of missing points is denoted by $|\Omega_{\text{miss}}|$. The missing rate is defined as

$$\text{Missing rate} = \frac{|\Omega_{\text{miss}}|}{N} \times 100\%.$$

The global root mean square error (Global RMSE) is defined by

$$\text{Global RMSE} = \sqrt{\frac{1}{N} \sum_{i=0}^{m_1} \sum_{j=0}^{m_2} \|q_{ij} - \widehat{q}_{ij}\|^2}.$$

The root mean square error over the missing region (Missing RMSE) is defined as

$$\text{Missing RMSE} = \sqrt{\frac{1}{|\Omega_{\text{miss}}|} \sum_{(i,j) \in \Omega_{\text{miss}}} \|q_{ij} - \widehat{q}_{ij}\|^2}.$$

The mean absolute error over the missing region (Missing MAE) is defined by

$$\text{Missing MAE} = \frac{1}{|\Omega_{\text{miss}}|} \sum_{(i,j) \in \Omega_{\text{miss}}} \|q_{ij} - \widehat{q}_{ij}\|.$$

The maximum absolute reconstruction error over the missing region (Missing Max Error) is defined as

$$\text{Missing Max Error} = \max_{(i,j) \in \Omega_{\text{miss}}} \|q_{ij} - \widehat{q}_{ij}\|.$$

For scalar surface data, the above error metrics are computed directly from the scalar values. For parametric surfaces in $\mathbb{R}^3$, the absolute difference is replaced by the Euclidean norm between the corresponding reconstructed and exact points.

4.2. Numerical examples

Example 4.1. Tensor-product cubic B-spline surfaces with $(n_1 + 1) \times (n_2 + 1)$ control points are employed to reconstruct 513×521 points sampled from the peaks function

$$z = 3(1 - x)^2 e^{-x^2 - (y+1)^2} - 10(x/5 - x^3 - y^5) e^{-x^2 - y^2} - 1/3 e^{-(x+1)^2 - y^2}, \quad (x, y) \in [-3, 3] \times [-2.5, 2.5].$$

Example 4.2. Using bi-cubic B-spline surfaces with 37×40 control points to fit 61×81 points, the dataset is sampled from a real tooth surface model with locally missing regions.

Example 4.3. Tensor-product cubic B-spline surfaces with 41×36 control points are employed to reconstruct 161×161 points sampled from the parametric equation

$$\begin{cases} x(u, v) = \frac{A(\cos u \cos(2v) + B \sin u \cos v) \cos u}{B - \sin(2u) \sin(3v)}, \\ y(u, v) = \frac{A(\cos u \sin(2v) - B \sin u \sin v) \cos u}{B - \sin(2u) \sin(3v)}, \\ z(u, v) = \frac{B \cos^2 u}{B - \sin(2u) \sin(3v)}, \end{cases} \quad (u, v) \in [0, \pi] \times [0, \pi],$$

where $A = 2/3$, and $B = \sqrt{2}$.

4.3. Numerical results and discussion

4.3.1. Results under different missing patterns

We first test Example 4.1 using the proposed method on several types of missing region, including irregular, rectangular, and strip-shaped patterns. Table 1 lists the quantitative reconstruction results for representative cases.

As shown in Table 1, the proposed method achieves stable accuracy under different missing patterns. For the hole-type missing region, the missing data are locally concentrated, leading to relatively larger reconstruction errors inside the missing area. In contrast, the strip-type missing pattern preserves stronger directional continuity, resulting in a smaller missing-region RMSE even with a higher missing rate. Moreover, the global RMSE values indicate that the proposed method can effectively preserve the overall surface geometry in the presence of missing data.

Table 1. Quantitative comparison of surface reconstruction under different missing patterns.

Missing pattern	Missing rate	Global RMSE	Missing RMSE	Missing MAE	Missing max error
Hole region	1.35%	1.27×10^{-2}	1.09×10^{-1}	7.56×10^{-2}	3.26×10^{-1}
Strip region	6.63%	1.21×10^{-2}	4.71×10^{-2}	3.20×10^{-2}	3.71×10^{-1}

Figures 1 and 2 show that the reconstructed surfaces remain visually smooth and geometrically consistent with the original surfaces under both hole-type and strip-type missing patterns.

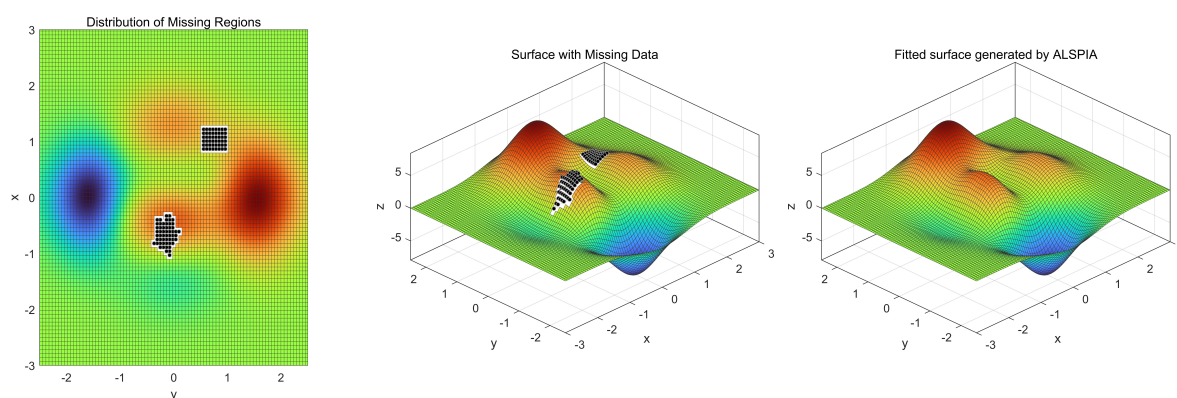


Figure 1. Surface reconstruction with mixed missing-data regions.

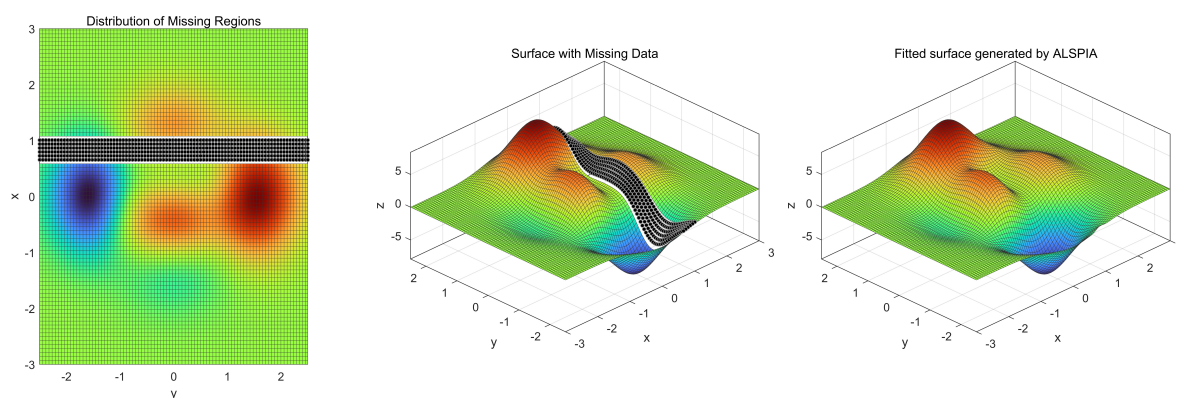


Figure 2. Surface reconstruction with strip-shaped missing-data regions.

The corresponding error and Gaussian curvature distributions in Figures 3 and 4 further demonstrate that the proposed method preserves the main geometric features of the original surface while maintaining relatively small reconstruction errors.

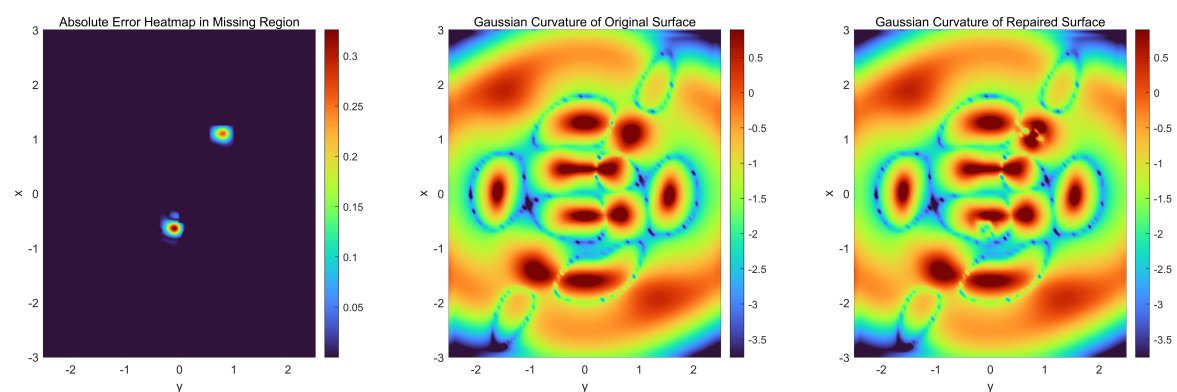


Figure 3. Absolute error and Gaussian curvature comparison for the hole-type missing pattern.

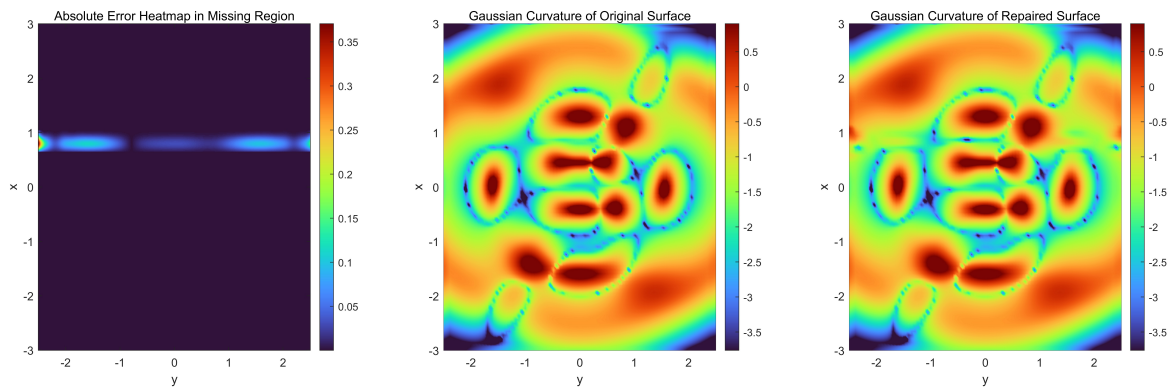


Figure 4. Absolute error and Gaussian curvature comparison for the strip-type missing pattern.

To evaluate the performance of the proposed method under disorganized missing-data patterns, we additionally consider randomly scattered missing entries for Example 4.1. For a prescribed missing rate ρ , each data entry is independently designated as missing with probability ρ . Accordingly, the mask matrix is generated according to

$$w_{ij} = \begin{cases} 0, & \text{with probability } \rho, \\ 1, & \text{with probability } 1 - \rho. \end{cases}$$

To reduce the influence of randomness, each experiment is independently repeated 20 times, and the mean and standard deviation of the corresponding error measures and CPU times are reported in Table 2. As the missing rate increases from 5% to 30%, the global RMSE increases from 1.40×10^{-4} to 7.89×10^{-4} , and the missing-region RMSE increases from 6.29×10^{-4} to 1.44×10^{-3} . Similarly, the missing-region MAE increases from 3.30×10^{-4} to 7.41×10^{-4} . These results are expected because fewer observed data points are available to constrain the control points as the missing rate increases. Nevertheless, even at a missing rate of 30%, both the global and missing-region reconstruction errors remain relatively small, demonstrating that the proposed method can effectively handle randomly and discontinuously distributed missing entries.

Table 2. Reconstruction results under randomly distributed missing entries.

Missing rate	Global RMSE	Missing RMSE	Missing MAE	CPU (s)
5%	$1.40 \times 10^{-4} \pm 1.38 \times 10^{-5}$	$6.29 \times 10^{-4} \pm 6.16 \times 10^{-5}$	$3.30 \times 10^{-4} \pm 2.81 \times 10^{-5}$	0.49
10%	$2.28 \times 10^{-4} \pm 2.44 \times 10^{-5}$	$7.20 \times 10^{-4} \pm 7.66 \times 10^{-5}$	$3.81 \times 10^{-4} \pm 2.80 \times 10^{-5}$	0.64
20%	$4.63 \times 10^{-4} \pm 4.07 \times 10^{-5}$	$1.03 \times 10^{-3} \pm 9.08 \times 10^{-5}$	$5.87 \times 10^{-4} \pm 5.83 \times 10^{-5}$	0.81
30%	$7.89 \times 10^{-4} \pm 7.73 \times 10^{-5}$	$1.44 \times 10^{-3} \pm 1.40 \times 10^{-4}$	$7.41 \times 10^{-4} \pm 8.22 \times 10^{-5}$	0.94

The relatively small standard deviations listed in Table 2 further indicate that the reconstruction performance is stable with respect to different random realizations of the missing-data mask. This stability can be attributed to the scattered distribution of the remaining observations over the parameter domain. Unlike a large connected missing region, randomly scattered missing entries generally preserve observed data within the local supports of most B-spline basis functions, thereby providing

sufficient constraints for the corresponding control points. In addition, the average CPU time increases moderately from 0.49 s to 0.94 s as the missing rate increases from 5% to 30%, indicating that the method maintains reasonable computational efficiency.

Figure 5 presents a representative reconstruction result with a random missing rate of 20%. Although the missing entries are scattered throughout the parameter domain, the absolute errors remain small, and the fitted surface successfully preserves the main geometric features of the original surface. This observation is consistent with Table 2.

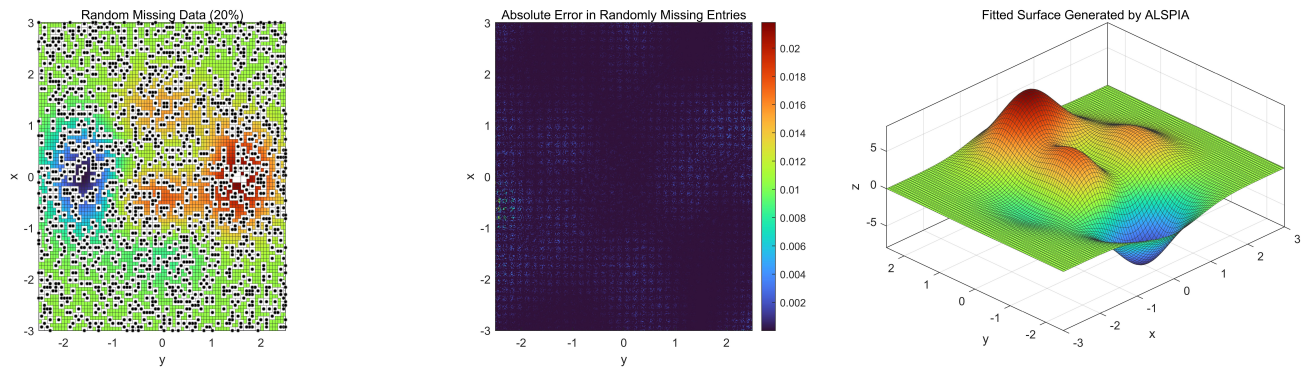


Figure 5. Reconstruction results for the synthetic surface with a random missing rate of 20%.

4.3.2. Influence of the number of control points

The number of control points determines both the approximation capability of the tensor-product B-spline surface and the computational cost of the proposed method. To investigate the influence of the tensor-product B-spline representation, we test the proposed method under different numbers of control points. Table 3 reports the corresponding reconstruction errors and computational costs.

Table 3. Reconstruction performance under different numbers of control points.

$(n_1 + 1, n_2 + 1)$	Global RMSE	Global MAE	Missing RMSE	Missing MAE	Missing Max	CPU (s)
(21, 21)	7.98×10^{-4}	6.79×10^{-5}	6.88×10^{-3}	5.05×10^{-3}	1.87×10^{-2}	0.27
(31, 31)	1.30×10^{-2}	9.92×10^{-4}	1.12×10^{-1}	7.37×10^{-2}	3.61×10^{-1}	0.67
(41, 36)	1.27×10^{-2}	1.02×10^{-3}	1.09×10^{-1}	7.56×10^{-2}	3.26×10^{-1}	0.75
(51, 46)	6.53×10^{-2}	4.82×10^{-3}	5.63×10^{-1}	3.58×10^{-1}	1.98	1.03
(61, 56)	8.62×10^{-2}	6.24×10^{-3}	7.43×10^{-1}	4.63×10^{-1}	2.62	2.08

As shown in Table 3, increasing the number of control points does not necessarily improve the reconstruction accuracy. For example, for the 21×21 control mesh, the proposed method achieves a global RMSE of 7.98×10^{-4} and a missing-region RMSE of 6.88×10^{-3} , with a CPU time of 0.27 seconds. When the control mesh is increased to 61×56 , the corresponding errors increase to 8.62×10^{-2} and 7.43×10^{-1} , and the CPU time increases to 2.08 seconds. This result indicates that an excessively dense control mesh may increase the computational cost without improving reconstruction accuracy. As the control mesh becomes denser, the supports of the B-spline basis functions become smaller. Some basis functions may then have insufficient overlap with the observed data surrounding the missing regions, causing the associated control points to become weakly constrained. This may

deteriorate the conditioning of the weighted least-squares system and produce local oscillations. Therefore, a moderate control-mesh resolution provides a more favorable balance among geometric flexibility, reconstruction accuracy, numerical stability, and computational cost.

Figures 6 and 7 visually illustrate the reconstructed surfaces and their corresponding absolute error distributions for different control mesh sizes. For moderate control mesh resolutions, the reconstructed surfaces accurately capture the main geometric features well and maintain smooth transitions across missing regions. This is consistent with the quantitative results reported in Table 3.

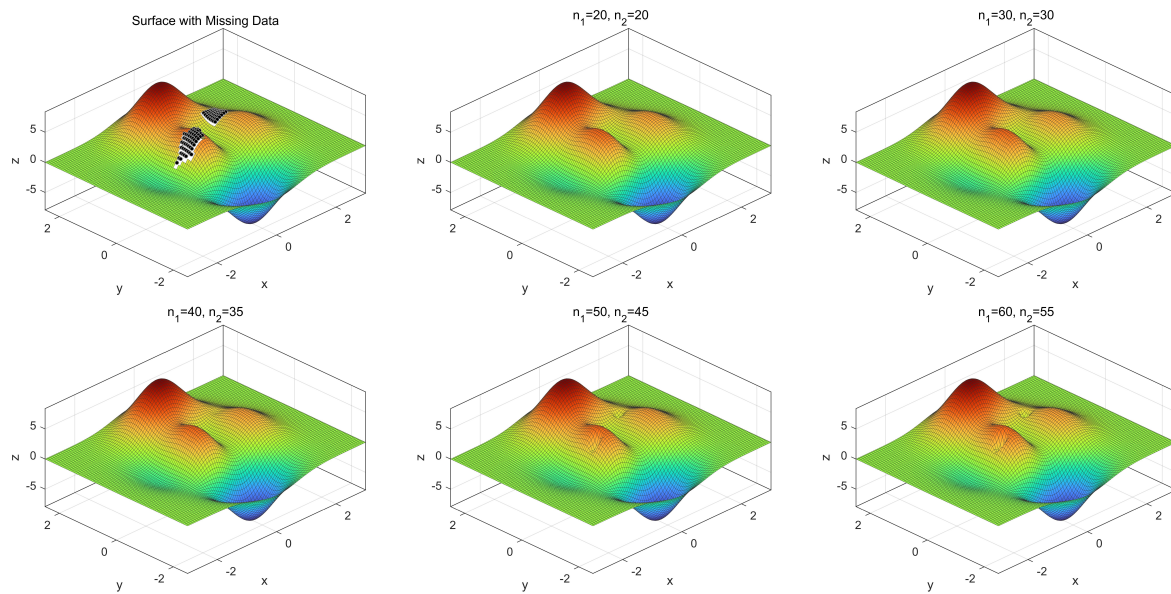


Figure 6. Surface reconstruction under different numbers of control points for Example 4.1.

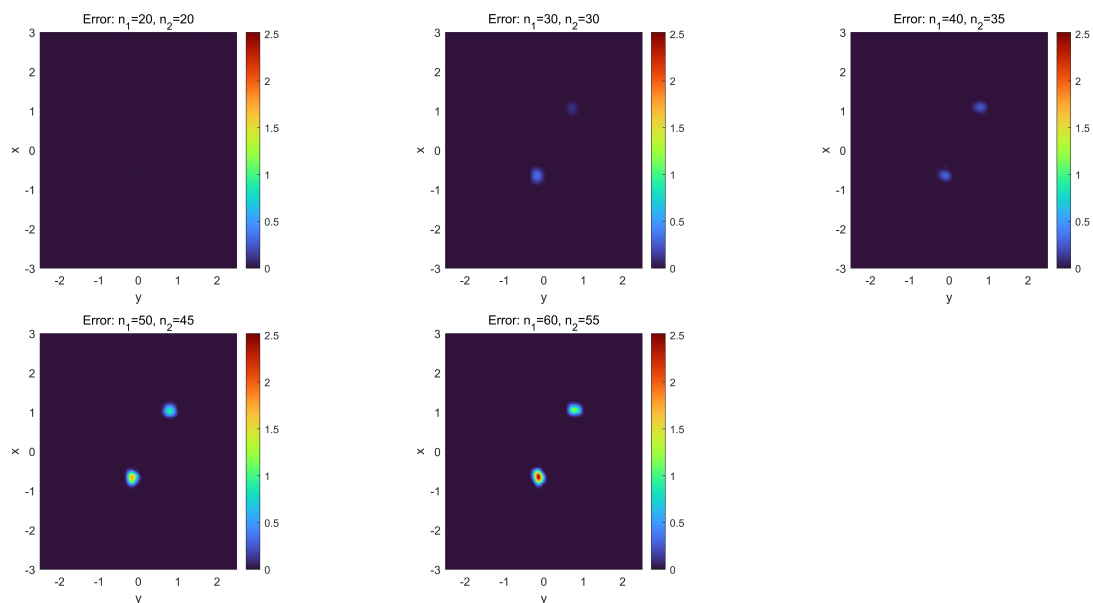


Figure 7. Absolute error under different numbers of control points for Example 4.1.

4.3.3. More numerical results on real and parametric surface datasets

Furthermore, we test the proposed method on the real sampled dataset in Example 4.2 and the parametric surface sampled dataset in Example 4.3. Figures 8–11 present the corresponding reconstruction results and absolute error distributions.

As shown in Figures 8 and 10, the proposed ALSPIA framework can effectively recover the missing surface regions while preserving the overall geometric structure of the original surfaces. The reconstructed surfaces remain smooth near the missing areas without introducing obvious visual artifacts.

Moreover, the absolute error distributions in Figures 9 and 11 indicate that the reconstruction errors are mainly concentrated around the missing regions, while the remaining regions maintain relatively small errors. These results further demonstrate the robustness and effectiveness of the proposed tensor-product ALSPIA method for different types of surface datasets.

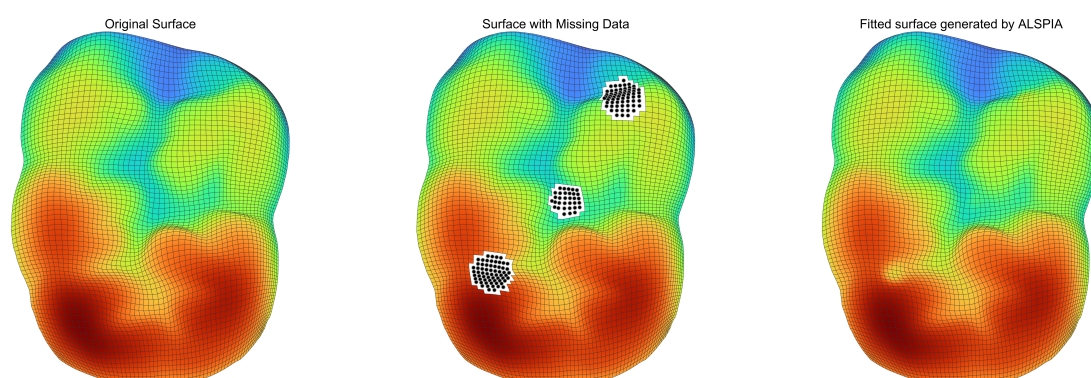


Figure 8. Surface reconstruction for Example 4.2.

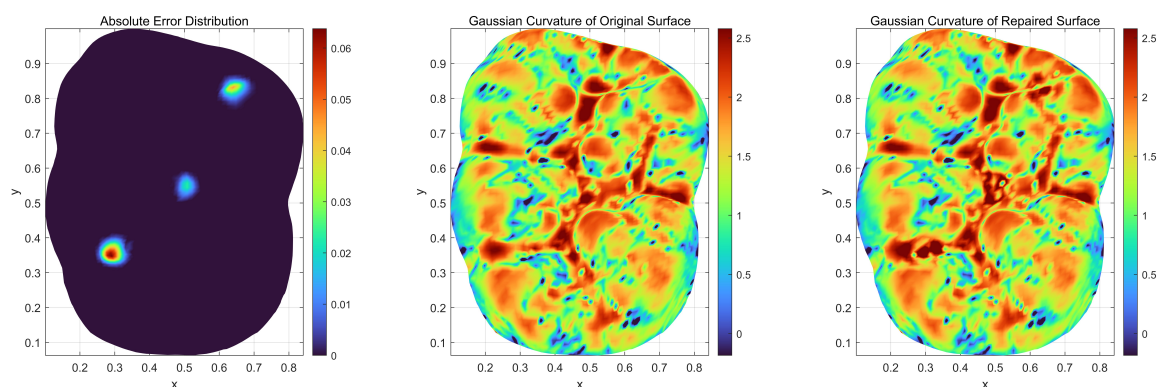


Figure 9. Absolute error and Gaussian curvature comparison for Example 4.2.

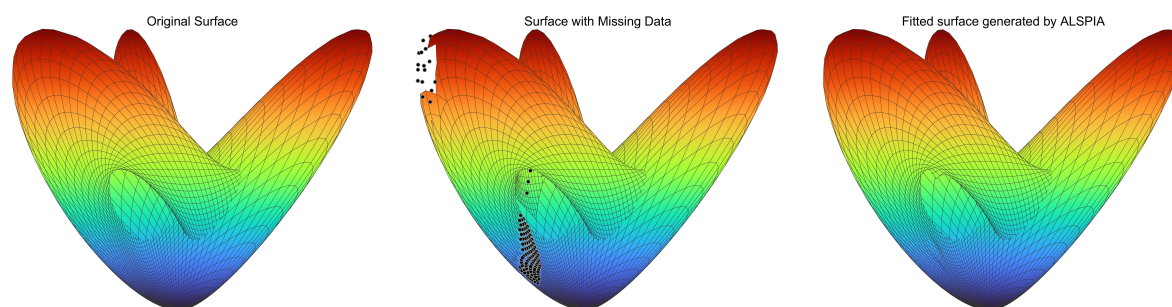


Figure 10. Surface reconstruction for Example 4.3.

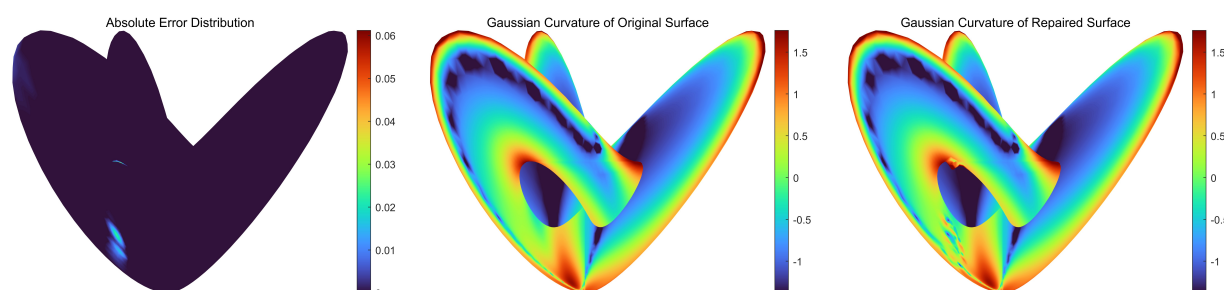


Figure 11. Absolute error and Gaussian curvature comparison for Example 4.3.

5. Conclusions

This paper has developed a tensor-structured formulation and implementation of ALSPIA for the fitting and reconstruction of incomplete tensor-product B-spline surface data. The binary mask represents the availability of the gridded observations and ensures that only the observed entries contribute to the least-squares objective.

The proposed matrix iteration is algebraically equivalent to applying ALSPIA to the vectorized least-squares system containing only the observed entries. Its main advantage is that the residual correction is evaluated directly through matrix operations involving the two one-dimensional collocation matrices and the binary mask. Consequently, the selection matrix, the full Kronecker-product matrix, and the row-restricted coefficient matrix do not need to be explicitly constructed.

The convergence properties of the tensor-structured iteration have been analyzed for both singular and nonsingular weighted normal matrices. Numerical experiments demonstrate that the method can reconstruct different types of missing surface regions while preserving the principal geometric characteristics of the original data.

Future work will focus on adaptive local refinement, regularized reconstruction near large missing regions, and applications to image and point-cloud data.

Author contributions

Xianglan Ge: Software, writing—original draft; Lijuan Hu: Conceptualization, methodology, writing—review & editing, supervision. All authors have read and approved the final version of the manuscript for publication.

Use of Generative-AI tools declaration

The authors declare they have not used Artificial Intelligence (AI) tools in the creation of this article.

Acknowledgments

This work was partially supported by the Natural Science Foundation of Hunan Province under grant number 2025JJ70357 and the College Student Innovation and Entrepreneurship Training Program Project, Hunan University of Humanities, Science and Technology.

Conflict of interest

The authors declare that they have no conflict of interest.

References

1. G. Farin, *Curves and surfaces for CAGD: A practical guide*, 5 Eds., Morgan Kaufmann, 2002. <https://doi.org/10.1016/B978-1-55860-737-8.X5000-5>
2. M. Unser, Splines: A perfect fit for signal and image processing, *IEEE Signal Process. Mag.*, **16** (1999), 22–38. <https://doi.org/10.1109/79.799930>
3. D. Mokriš, B. Jüttler, Using low-rank approximations of gridded data for spline surface fitting, *J. Comput. Appl. Math.*, **438** (2024), 115519. <https://doi.org/10.1016/j.cam.2023.115519>
4. S. Merchel, B. Jüttler, D. Mokriš, M. Pan, Fast formation of matrices for least-squares fitting by tensor-product spline surfaces, *Comput. Aided Des.*, **150** (2022), 103307. <https://doi.org/10.1016/j.cad.2022.103307>
5. H. Lin, T. Maekawa, C. Deng, Survey on geometric iterative methods and their applications, *Comput. Aided Des.*, **95** (2018), 40–51. <https://doi.org/10.1016/j.cad.2017.10.002>
6. C. Deng, H. Lin, Progressive and iterative approximation for least-squares B-spline curve and surface fitting, *Comput. Aided Des.*, **47** (2014), 32–44. <https://doi.org/10.1016/j.cad.2013.08.012>
7. Q. Chang, W. Ma, C. Deng, Constrained least square progressive and iterative approximation (CLSPIA) for B-spline curve and surface fitting, *Vis. Comput.*, **40** (2024), 4427–4439. <https://doi.org/10.1007/s00371-023-03090-8>
8. N. Wu, C. Liu, Asynchronous progressive iterative approximation method for least squares fitting, *Comput. Aided Geom. Des.*, **111** (2024), 102295. <https://doi.org/10.1016/j.cagd.2024.102295>

9. L. Lan, Y. Ji, M. Y. Wang, C. G. Zhu, Full-LSPIA: A least-squares progressive-iterative approximation method with optimization of weights and knots for NURBS curves and surfaces, *Comput. Aided Des.*, **169** (2024), 103673. <https://doi.org/10.1016/j.cad.2023.103673>
10. N. Wu, C. Liu, Randomized progressive iterative approximation for B-spline curve and surface fittings, *Appl. Math. Comput.*, **473** (2024), 128669. <https://doi.org/10.1016/j.amc.2024.128669>
11. N. -C. Wu, H. -H. Cao, C. Liu, Momentum-accelerated randomized geometric iterative methods for curve and surface approximation, *Comput. Aided Des.*, **192** (2026), 104011. <https://doi.org/10.1016/j.cad.2025.104011>
12. Z. Yao, Q. Hu, Accelerated local progressive-iterative approximation methods for curve and surface fitting, *Vis. Comput.*, **41** (2025), 5979–5993. <https://doi.org/10.1007/s00371-024-03764-x>
13. C. Liu, N. -C. Wu, J. Li, Efficient two-dimensional randomized progressive iterative approximation for large-scale B-spline fitting, *CSIAM Trans. Appl. Math.*, **7** (2026), 556–575. <https://doi.org/10.4208/csiam-am.SO-2025-0062>
14. C. Liu, N. -C. Wu, J. Li, L. Hu, Two novel iterative approaches for improved LSPIA convergence, *Comput. Aided Geom. Des.*, **111** (2024), 102312. <https://doi.org/10.1016/j.cagd.2024.102312>
15. J. Liu, P. Musialski, P. Wonka, J. Ye, Tensor completion for estimating missing values in visual data, *IEEE Trans. Pattern Anal. Mach. Intell.*, **35** (2013), 208–220. <https://doi.org/10.1109/TPAMI.2012.39>
16. S. Gandy, B. Recht, I. Yamada, Tensor completion and low- n -rank tensor recovery via convex optimization, *Inverse Probl.*, **27** (2011), 025010.
17. D. Kressner, M. Steinlechner, B. Vandereycken, Low-rank tensor completion by Riemannian optimization, *Bit Numer. Math.*, **54** (2014), 447–468. <https://doi.org/10.1007/s10543-013-0455-z>
18. H. Lin, Q. Cao, X. Zhang, The convergence of least-squares progressive iterative approximation for singular least-squares fitting system, *J. Syst. Sci. Complex.*, **31** (2018), 1618–1632. <https://doi.org/10.1007/s11424-018-7443-y>
19. S. Li, H. Xu, C. Deng, Data-weighted least square progressive and iterative approximation and related B-spline curve fitting, *J. Comput.-Aided Des. Comput. Graph.*, **31** (2019), 1574–1580. <https://dx.doi.org/10.3724/SP.J.1089.2019.17585>
20. V. Simoncini, Computational methods for linear matrix equations, *SIAM Rev.*, **58** (2016), 377–441. <https://doi.org/10.1137/130912839>



AIMS Press

©2026 the Author(s), licensee AIMS Press. This is an open access article distributed under the terms of the Creative Commons Attribution License (<https://creativecommons.org/licenses/by/4.0>)