Electronic Research Archive

https://www.aimspress.com/journal/era

ERA, 33(11): 6558–6576.
DOI: 10.3934/era.2025290
Received: 20 August 2025
Revised: 09 October 2025
Accepted: 28 October 2025
Published: 04 November 2025

Research article

Preconditioned Landweber iteration for nonlinear least-squares conical surface fitting

Shangzuo Xie, Gangrong Qu*[*] and Liyi Feng

School of Mathematics and Statistics, Beijing Jiaotong University, Beijing 100044, China

* **Correspondence:** Email: grqu@bjtu.edu.cn.

Abstract: To address the slow convergence and sensitivity to initial values of the traditional Levenberg-Marquardt (LM) algorithm in 3D point cloud conical surface fitting-issues caused by ill-conditioned Hessian matrices—this paper developed a preconditioned Landweber iterative framework for nonlinear least-squares optimization. A residual model based on point-to-cone geometric distances was constructed, the analytical Jacobian was derived, and a spectral preconditioning strategy was introduced. Theoretical analysis showed that the preconditioner effectively reduced the condition number of the normal matrix, thereby improving numerical stability and accelerating practical convergence. A relaxation-factor rule was further derived to balance stability and efficiency. Experiments on synthetic and industrial datasets demonstrated that, compared with the Levenberg-Marquard(LM) algorithm, the proposed method reduced iteration counts by 51.9%, decreased root mean square error (RMSE) by 98%, and shortened computation time by 73%. It also maintained robust performance under high-noise and non-uniform sampling conditions, providing an efficient and reliable tool for accurate conical surface fitting in industrial inspection and reverse engineering.

Keywords: preconditioned Landweber iteration; nonlinear least-squares optimization; conical surface fitting; spectral preconditioning; convergence analysis

1. Introduction

In modern engineering and geoscience, precise geometric reconstruction and model fitting have become critical tools for high-accuracy data processing. Applications ranging from mechanical component design, quality inspection, and digital twins to geological resource assessment and environmental simulation often involve target objects composed of standard quadric surfaces, such as planes, cylinders, spheres, tori, and conical surfaces [1–3]. These geometric primitives are ubiquitous in industrial and geological engineering, where their accurate parameter measurement directly impacts product assembly precision, service life, and subsurface resource utilization. Particularly in geological and environmental modeling—including mining, stratigraphic analysis, and hazard risk assessment—the accurate reconstruction of conical structures in rock masses or soil bodies not only reveals subsurface configurations but also provides key insights for resource evaluation and risk prediction [4].

Traditionally, engineers and geoscientists relied on manual tools (e.g., rulers, calipers, micrometers) for surface measurements. However, these methods struggle to achieve the required precision when handling large-scale, high-density point cloud data or complex terrains. With the rapid advancement of laser scanning, remote sensing, and 3D sensing technologies, high-precision dense point clouds are now widely used in geological information systems (GIS) and environmental monitoring [5]. Extracting accurate geometric parameters from such massive, often noisy point cloud datasets has emerged as a pressing challenge.

Among geometric fitting methods, nonlinear least-squares optimization algorithms—particularly the Levenberg-Marquardt (LM) and Gauss-Newton (GN) methods—are commonly employed for their theoretical foundation and practical effectiveness [3, 6]. While these methods achieve high fitting accuracy under favorable conditions, they face significant limitations in practical applications. The LM algorithm, despite its robustness through damping, is notoriously sensitive to ill-conditioned Hessian matrices arising from parameter scaling discrepancies and noisy observations [7, 8]. This ill-conditioning leads to numerical instability during matrix inversion, manifesting as oscillatory convergence or even divergence. Furthermore, the performance of LM is heavily constrained by initial parameter selection, with poor initialization often trapping the optimization in suboptimal local minima [9]. For high-dimensional geometric models like conical surfaces, these issues are exacerbated due to the complex interaction between vertex coordinates, axial directions, and angular parameters. Although Random sample consensus(RANSAC)-based methods offer robustness to outliers, they face efficiency limitations due to the combinatorial explosion of sampling iterations in high-dimensional parameter spaces [10].

To address these challenges, this study proposes a novel preconditioned Landweber iteration framework specifically tailored for nonlinear conical surface fitting. While the Landweber method has proven effective in linear inverse problems such as computed tomography reconstruction [11] and bundle adjustment [12], its application to geometric surface fitting remains largely unexplored. The proposed method extends this classical iteration from linear to nonlinear problems by constructing the objective function based on rigorous point-to-cone geometric distances and deriving the complete analytical Jacobian matrix. A spectrally-designed diagonal preconditioning matrix is introduced to adaptively scale gradient steps, effectively improving the Hessian condition number and significantly enhancing convergence speed and numerical stability.

Comprehensive experiments on synthetic and industrial scan datasets demonstrate that the proposed method outperforms traditional Levenberg-Marquardt algorithms in iteration count, computation time, memory usage, and fitting accuracy. Moreover, the method maintains stable convergence under significant noise and non-uniform point cloud distributions, highlighting its robustness and engineering applicability.

The main contributions of this paper are threefold:

- Novel Optimization Framework: We develop the first preconditioned Landweber iterative scheme for nonlinear least-squares conical surface fitting, extending its application from linear inverse problems to complex geometric optimization tasks.

- Spectral Preconditioning Strategy: We design an innovative diagonal preconditioner through geometric analysis of the Hessian matrix structure, with theoretical proof demonstrating its effectiveness in reducing the condition number of the normal equation system.
- Optimal Relaxation trategy: We derive a theoretically-grounded relaxation factor selection rule that automatically balances convergence stability and efficiency, eliminating the need for manual parameter tuning required in traditional LM approaches.

In summary, the proposed algorithm offers a systematic mathematical framework and efficient numerical implementation for quadric surface fitting in point cloud data. The combination of theoretical proof and numerical validation demonstrates significant advantages in robustness, convergence efficiency, and noise adaptability, fulfilling high-precision requirements in industrial manufacturing and mechanical inspection, while providing a mathematically rigorous and extensible tool for 3D data processing in GIS and environmental modeling.

2. Related work

Conical surface fitting has extensive applications in engineering, geology, computer vision, and related fields. This mathematical method aims to fit a set of data points to a conical surface model, with parameters typically including the cone's apex, axial direction, and half-apex angle. The objective is to determine optimal parameters that minimize the distance between data points and the fitted conical surface.

Rusu et al. proposed semantic 3D object models and demonstrated how to automatically obtain these models from dense 3D ranging data [5]. Schnabel et al. developed an automated algorithm for conical point cloud detection based on random sampling theory and plane detection. This method derives initial values from preliminary data and iteratively optimizes geometric parameters, but suffers from insufficient accuracy in initial value estimation and prolonged iteration time [10]. Lukács introduced a least-squares fitting method for conical surfaces requiring parameter transformation for computation [1, 13]. Shakarji proposed an orthogonal distance regression-based algorithm with stringent initial value requirements and poor robustness during Levenberg-Marquardt optimization [3]. Sun and Tian constructed an objective function using conical coordinates, normal vectors, and geometric features to transform nonlinear fitting into linear least-squares problems [14, 15]. Gong iteratively derived conical parameters directly from point clouds via least-squares theory, albeit with complex parametric modeling [16]. Deng proposed a novel method for optimizing axial direction, half-apex angle, and apex parameters using Levenberg-Marquardt optimization [9]. Karambakhsh et al. employed three compact Convolutional Neural Networks(CNN) models combined with geometric features (surface normals, curvature) and dual neural networks for 3D object recognition [17]. Jiang et al. implemented parametric decomposition for surface fitting to reduce the likelihood of local minima during iterative optimization [8]. Meghashyam et al. significantly reduced computational time through Graphics Processing Unit(GPU)-accelerated operations [18]. Mulchrone et al. developed a Mathematica application for estimating best-fit cones from orientation data [4]. Shakarji et al. provided practical algorithms and heuristic methods for industrial applications of geometric fitting [19]. Marshall et al. integrated least-squares fitting of quadric surfaces into segmentation frameworks [2].

The Landweber method has been applied to computed tomography (CT) reconstruction [11], light field restoration [20], and bundle adjustment in 3D reconstruction [12]. However, to our knowledge, it has not been applied to conical surface fitting. We therefore propose a novel Landweber-based conical surface fitting algorithm, representing the first implementation of this approach in the field.

Existing conical fitting methods face three key trade-offs: 1) speed versus geometric distortion in algebraic fitting, 2) precision versus numerical instability in geometric optimization, and 3) adaptability versus computational cost in data-driven approaches. This work innovatively introduces preconditioned Landweber iteration to geometric fitting. By designing diagonal preconditioners through geometric analysis of Hessian matrix structures, the proposed method theoretically overcomes the convergence limitations of traditional LM algorithms. Compared to classical approaches, our framework demonstrates significant improvements in noise immunity , computational complexity, and reduced sensitivity to initial values.

3. Problem modeling

3.1. Conic parameterization and geometric residuals

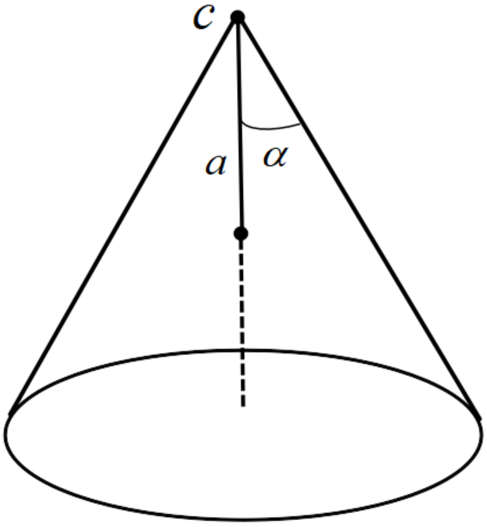


Figure 1. Schematic of the conical model.

The conical surface model describes a rotational surface formed by a cone, whose mathematical and geometric properties hold significant practical implications. Conical surface fitting constitutes a mathematical modeling technique designed to fit a given set of data points to a conical surface model. In conical fitting problems, clear parametric modeling and well-defined optimization objectives form the foundation of algorithm design. As illustrated in Figure 1, a cone is uniquely determined by three parameters: the vertex $c \in R^3$, the axial unit vector $a \in R^3(\|a\| = 1)$, and the half-apex angle $\alpha \in (0, \frac{\pi}{2})$. Based on the geometric definition of a conical surface, the ratio of a point's distance to the axis and its distance to the vertex equals $\sin\alpha$. Consequently, the conical surface comprises all points p satisfying the following condition:

$$\frac{\|(p - c) \times a\|}{\|p - c\|} = \sin\alpha, \tag{3.1}$$

where × represents the cross product operation between vectors.

Geometric residuals should be defined to reflect the distance difference between the point and the conical surface. For any point p_i, the distance from the axis $\|(p_i - c) \times a\|$ should be the same as the theoretical value $\|p_i - c \sin\alpha$. The normalized residual formula is defined as:

$$d_i = \frac{\|(p_i - c) \times a\| - \|p_i - c\| \sin\alpha}{\sqrt{1 + \sin^2\alpha}}, \tag{3.2}$$

by normalizing the denominator factor, the influence of different α on the residual dimension is eliminated to ensure the comparability of residuals under different parameters.

The optimization goal is to minimize the sum of squares of residuals at all points:

$$E(\theta) = \frac{1}{2}\sum_{i=1}^{N} d_i^2, \quad \theta = (c, a, \alpha) \tag{3.3}$$

where the unit vector constraint $\|a\| = 1$ and the half-vertex range constraint $\alpha \in (0, \frac{\pi}{2})$ must be satisfied.

$E(\theta)$ measures the overall deviation between the set of points $\{p_i\}_{i=1}^N$ and the conical surface model, and it is a typical residual sum-of-squares function. Since each residual d_i is a nonlinear function of the parameters c, a, and α, the function $E(\theta)$ exhibits a complex nonlinear dependence on these parameters. In particular, the directional nature of the axis vector a and the unit vector constraint make the optimization space a curved nonlinear manifold, while the trigonometric relation of the half-apex angle α further increases the nonlinear sensitivity of the residuals. Therefore, the conical surface fitting problem is a typical nonlinear least-squares problem and needs to be solved by iterative optimization algorithms, commonly using the Levenberg-Marquardt method. During the solution process, the parameters θ are iteratively updated to reduce the residual sum-of-squares $E(\theta)$, approaching the optimal fitting solution.

3.2. The bottleneck of LM algorithm

LM algorithm iteratively updates parameters by the damped Gauss-Newton method, and its core formula is as follows:

$$\Delta\theta = -(J^T J + \mu I)^{-1} J^T r \tag{3.4}$$

where J is the Jacobian matrix of residuals against parameters, r is the residuals vector, and μ is the damping factor.

The LM algorithm exhibits critical limitations in practical implementations. First, significant parameter scaling discrepancies (e.g., differing units between vertex coordinates and angular parameters) create substantial variations in the L_2-norms of the Jacobian matrix J column vectors, resulting in a high condition number for $J^T J$. This induces numerical instability during matrix inversion, manifesting as oscillatory step sizes or divergence. Second, the algorithm demonstrates heightened sensitivity to noise and initial conditions: minor data noise becomes amplified by ill-conditioned matrices, distorting optimization trajectories, while perturbed residual surfaces exacerbate local minima entrapment. Third, geometric constraints—specifically the unit vector requirement ($\|a\| = 1$) and half-apex angle domain ($\alpha \in (0, \pi/2)$)—necessitate specialized handling. Conventional constraint-management techniques like parameter projection or variable substitution may compromise optimization path continuity, potentially degrading convergence rates or trapping iterations in local minima.

3.3. Mathematical derivation of Jacobian matrix

In the conical fitting problem, each element of the Jacobian matrix J corresponds to the partial derivative of the residual d_i with respect to the parameter θ. Let $v = p_i - c$, then the residual can be written as:

$$d_i = \frac{||v \times a|| - ||v|| \sin\alpha}{\sqrt{1+\sin^2\alpha}}. \tag{3.5}$$

The first derivative is:

$$\frac{\partial ||v \times a||}{\partial c} = \frac{(v \times a)^T}{||v \times a||} \cdot \frac{\partial (v \times a)}{\partial c} = -\frac{(v \times a)^T (I \times a)}{||v \times a||} \tag{3.6}$$

where $I \times a$ is the cross product of the Jacobian matrix with dimension 3×3.

The second derivative is:

$$\frac{\partial (||v|| \sin\alpha)}{\partial c} = -\sin\alpha \cdot \frac{v^T}{||v||}. \tag{3.7}$$

So we get:

$$\frac{\partial d_i}{\partial c} = \frac{1}{\sqrt{1+\sin^2\alpha}} \left(-\frac{(v \times a)^T (I \times a)}{||v \times a||} + -\sin\alpha \cdot \frac{v^T}{||v||}\right). \tag{3.8}$$

Since a is a unit vector, we need to deal with the constraint $||a|| = 1$, using local parameterization or directly taking the derivative of a and projecting:

$$\frac{\partial d_i}{\partial a} = \frac{1}{\sqrt{1+\sin^2\alpha}} \cdot \frac{(v \times a)^T (v \times \cdot)}{||v \times a||}. \tag{3.9}$$

Taking the derivative of α directly yields:

$$\frac{\partial d_i}{\partial \alpha} = \frac{-||v|| \cos\alpha \cdot \sqrt{1+\sin^2\alpha} - (||v \times a|| - ||v|| \sin\alpha) \cdot \frac{\sin\alpha \cos\alpha}{\sqrt{1+\sin^2\alpha}}}{1+\sin^2\alpha}. \tag{3.10}$$

For each point p_i, a row of the Jacobian matrix contains:

$$J_i = [\frac{\partial d_i}{\partial c}, \frac{\partial d_i}{\partial a}, \frac{\partial d_i}{\partial \alpha}]. \tag{3.11}$$

4. Preconditioned Landweber iterative algorithm

For the nonlinear least-squares problem, the proposed algorithm linearizes the residual function at the current parameter estimate $\theta^{(k)}$ in each iteration, transforming the original problem into a linear least-squares subproblem, which is then solved using a preconditioned Landweber iterative framework.

Many mathematical applications can be approximated as a linear system:

$$Ax = b. \tag{4.1}$$

Among them, the $b = [b^1, b^2, \cdots, b^n]^T$ is observation data, $x = [x^1, x^2, \cdots, x^n]^T$ is real solutions, and T is said vector and matrix transpose. Here, the field K can be the real number R or the complex number C, and the dimension of the system matrix A is $M \times N$.

For the system of linear equations (4.1), an algebraic reconstruction method has been proposed to solve iteratively. One widely used approach is the weighted Landweber iteration [7,21–23] :

$$x^{(n+1)} = x^{(n)} + \lambda_n V^{-1} A^T W(b - Ax^{(n)}). \tag{4.2}$$

For $n = 0, 1, \cdots, n$, parameter λ_n is known as the relaxation factor and, $x^{(0)}$is the initial value. Here, A^* represents the conjugate transpose of A, and V and W are two positive definite matrices with dimensions of $N \times N$ and $M \times M$, respectively. If the $y^{(n)} = V^{\frac{1}{2}} x^{(n)}, D = W^{\frac{1}{2}} A V^{-\frac{1}{2}}, \overline{b} = W^{\frac{1}{2}} b$, Landweber iteration (4.2) can be converted to the unweighted Landweber iteration [7,24,25]:

$$x^{(n+1)} = x^{(n)} + \lambda_n A^T (b - Ax^{(n)}). \tag{4.3}$$

The Landweber iteration is used to solve the least squares solution of (4.1). The least square functional is:

$$L(x) = \frac{1}{2} \|b - Ax\|^2, \tag{4.4}$$

where $\| \cdot \|$ represents the Euclidean norm. We solve this by minimizing the Eq (4.4). The gradient of L [6] can be written as:

$$\nabla L(x) = -A^T (b - Ax). \tag{4.5}$$

Therefore, all minima of L satisfy the following normal equation:

$$A^T A x = A^T b. \tag{4.6}$$

The minimizer of the least-squares functional L corresponds to all solutions satisfying Eq (4.6). This equation is always solvable and possesses a unique minimum 2-norm solution x^+, which equals $A^+ b$, where A^+ denotes the Moore-Penrose pseudoinverse of matrix A [26]. The Landweber iteration scheme (4.2) constitutes a method derived from the least-squares functional framework. It has been proven that under convergence conditions, the Landweber iteration converges to the solution $x^+ + \mathcal{P}_{N(A)}(x^{(0)})$ of the normal equations (4.6), where $\mathcal{P}_{N(A)}(x^{(0)})$ represents the orthogonal projection of the initial value $x^{(0)}$ onto the null space of A.

4.1. Landweber weighted method

We propose a modified weighted Landweber iteration algorithm based on Eq (4.2), in which weighting matrices V and W are introduced to reduce the condition number cond($A^T A$). This modification aims to accelerate the convergence and reduce the iteration error.

Let $\{\mu_k\}_{k=1}^m$ be the set of distinct positive eigenvalues of the matrix $A^T A$, arranged in descending order as $\mu_1 > \mu_2 > \cdots > \mu_m > 0$. Each eigenvalue μ_k has algebraic multiplicity p_k, where $1 \leq k \leq m$. For each $1 \leq j \leq p_k$, denote by $u_{j,k}$ the orthonormal eigenvectors associated with μ_k. According to the standard eigenvalue-eigenvector relation, we have: $A^T A\, u_{j,k} = \mu_k u_{j,k}$ [24].

In addition, if $A^T A$ admits zero as an eigenvalue with multiplicity q, let $\{v_j\}_{j=1}^q$ denote a set of orthonormal eigenvectors corresponding to this zero eigenvalue, satisfying $A^T A\, v_j = \theta$, where θ is the zero vector. Based on this eigendecomposition, the spectral norm (i.e., the operator 2-norm) of matrix A is given by $\|A\|_2 = \mu_1$.

Let $U_k = (u_{1,k}, \dots, u_{p_k,k})$ for $k = 1, \dots, m$, and define $V = (v_1, \dots, v_q)$. Construct the matrix $U = (U_1, \dots, U_m, V)$. Then, the following diagonalization holds:

$$U^T A^T A U = \operatorname{diag}(\mu_1 I_{p_1}, \dots, \mu_m I_{p_m}, O_q). \tag{4.7}$$

Here, I_k denotes the $k \times k$ identity matrix, and O_q represents the $q \times q$ zero matrix. The set of eigenvectors $\{u_{j,k} : 1 \leq j \leq p_k,\ 1 \leq k \leq m\} \cup \{v_j : 1 \leq j \leq q\}$ forms a complete orthonormal basis for the Hilbert space $\mathbb{K}^N$.

Moreover, the collection $\{u_{j,k}\}$ spans an orthonormal basis for the orthogonal complement $\mathcal{N}(A)^\perp$, while the vectors $\{v_j\}$ form an orthonormal basis for the null space $\mathcal{N}(A)$ of matrix A.

Given that the eigenvalues $\mu_1 > \mu_2 > \cdots > \mu_m > 0$ are strictly decreasing, we refer to Eq (4.7) and utilize the orthogonal matrix U as the similarity transformation to construct a diagonalized matrix whose eigenvalues are arranged in ascending order. By applying the same matrix U, if $\omega > \mu_1/2$, we can derive the following:

$$U^T(2\omega I - A^T A)U = \operatorname{diag}((2\omega - \mu_1) I_{p_1}, \cdots, (2\omega - \mu_m) I_{p_m}, O_q). \tag{4.8}$$

In this case, the values $2\omega - \mu_1 < 2\omega - \mu_2 < \cdots < 2\omega - \mu_m$ form a strictly increasing sequence. We observe that choosing $\omega = (\mu_1 + \mu_m)/2$ satisfies the condition $\omega > \mu_1/2$. Therefore, the following result can be obtained:

$$U^T((\mu_1 + \mu_m)I - A^T A)U = \operatorname{diag}((\mu_1 + \mu_m - \mu_1) I_{p_1}, \cdots, (\mu_1 + \mu_m - \mu_m) I_{p_m}, O_q). \tag{4.9}$$

Here, $\mu_1 + \mu_m - \mu_1 < \mu_1 + \mu_m - \mu_2 < \cdots < \mu_1 + \mu_m - \mu_m$. Therefore, we propose to set the weighting matrix as $V^{-1} = (\mu_1 + \mu_m)I - A^T A$ and $W = I$. The motivation behind this choice is that the matrix $((\mu_1 + \mu_m)I - A^T A)$, when used as a preconditioner, effectively compresses the spectrum of $A^T A$ into the interval $[\mu_1 \mu_m, ((\mu_1 + \mu_m)/2)^2]$. Under this choice, the weighted Landweber iteration in Eq (4.2) can be rewritten as:

$$x^{(n+1)} = x^{(n)} + \lambda_n \left((\mu_1 + \mu_m) I - A^T A \right) A^T \left(b - A x^{(n)} \right). \tag{4.10}$$

The matrix $(\mu_1 + \mu_m)I - A^T A$ is symmetric and positive definite. Accordingly, the corresponding iteration matrix can be expressed as:

$$\begin{aligned} G(\lambda_n) =& U \operatorname{diag}((1 - \lambda_n(\mu_1 + \mu_m - \mu_1)\mu_1) I_{p_1}, (1 - \lambda_n(\mu_1 + \mu_m - \mu_2)\mu_2) I_{p_2} \\ & , \cdots, (1 - \lambda_n(\mu_1 + \mu_m - \mu_m)\mu_m) I_{p_m}, O_q) U^T. \end{aligned} \tag{4.11}$$

Definition 1. The condition number of the matrix $A^T A$ is defined as

$$\operatorname{cond}(A^T A) = \frac{\mu_1}{\mu_m}, \tag{4.12}$$

where μ_1 and μ_m denote the largest and smallest positive eigenvalues of $A^T A$, respectively.

Theorem 1. *The condition number of the matrix $(\mu_1 + \mu_m - A^T A)A^T A$ is smaller than that of $A^T A$, i.e.,*

$$\operatorname{cond}[((\mu_1 + \mu_m)I - A^T A)A^T A] < \operatorname{cond}(A^T A). \tag{4.13}$$

Proof. By applying the same similarity transformation matrix U as in Eq (4.7), we obtain:

$$U^T(((\mu_1+\mu_m)I-A^TA)A^TA)U = \operatorname{diag}((((\mu_1+\mu_m)-\mu_1)\mu_1)I_{p_1}, \\ (((\mu_1+\mu_m)-\mu_2)\mu_2)I_{p_2},\cdots,(((\mu_1+\mu_m)-\mu_m)\mu_m)I_{p_m},O_q). \tag{4.14}$$

The eigenvalues of the matrix $((\mu_1+\mu_m)I-A^{\top}A)A^TA$ are given by $(\mu_1+\mu_m-\mu_1)\mu_1$, $(\mu_1+\mu_m-\mu_2)\mu_2$, $\ldots$, $(\mu_1+\mu_m-\mu_m)\mu_m$. The algebraic multiplicity of each eigenvalue is denoted by p_k for $1\le k\le m$. We denote the eigenvalues of $(\mu_1+\mu_m-A^TA)A^TA$ by

$$\lambda_k=(\mu_1+\mu_m-\mu_k)\,\mu_k=\mu_k\,(\mu_1+\mu_m-\mu_k)\,,\quad k=1,\ldots,m. \tag{4.15}$$

Therefore, the condition number of $((\mu_1+\mu_m)I-A^{\top}A)A^TA$ is given by:

$$\operatorname{cond}(((\mu_1+\mu_m)I-A^{\top}A)A^TA)=\frac{\max_{1\le k\le m}\lambda_k}{\min_{1\le k\le m}\lambda_k}=\frac{\max_{1\le k\le m}\mu_k\,(\mu_1+\mu_m-\mu_k)}{\min_{1\le k\le m}\mu_k\,(\mu_1+\mu_m-\mu_k)}. \tag{4.16}$$

Let $\phi(\omega)=\omega\,(\mu_1+\mu_m-\omega)$, For $\mu_m\le\omega\le\mu_1$, the equivalent form of the function $\phi(\omega)$ can be expressed as:

$$\phi(\omega)=-(\omega-\frac{\mu_1+\mu_m}{2})^2+(\frac{\mu_1+\mu_m}{2})^2. \tag{4.17}$$

Therefore, $\phi(\omega)$ is a quadratic function of the variable ω, whose axis of symmetry is the line $\omega=\frac{\mu_1+\mu_m}{2}$. It follows directly that:

$$\arg\max_{\mu_m\le\omega\le\mu_1}\phi(\omega)=\frac{\mu_1+\mu_m}{2},\ \max_{\mu_m\le\omega\le\mu_1}\phi(\omega)=(\frac{\mu_1+\mu_m}{2})^2. \tag{4.18}$$

For $\mu_m\le\omega\le\mu_1$, the points $\omega=\mu_1$ and $\omega=\mu_m$ are equidistant from the axis of symmetry of the quadratic function. Without loss of generality, we take $\omega=\mu_1$ as the farthest point from the axis, and, thus, we obtain:

$$\arg\min_{\mu_m\le\omega\le\mu_1}\phi(\omega)=\mu_1,\ \min_{\mu_m\le\omega\le\mu_1}\phi(\omega)=\mu_1\mu_m. \tag{4.19}$$

So we can obtain:

$$\frac{\max_{\mu_m\le\omega\le\mu_1}\phi(\omega)}{\min_{\mu_m\le\omega\le\mu_1}\phi(\omega)}=\frac{(\frac{\mu_1+\mu_m}{2})^2}{\mu_1\mu_m}. \tag{4.20}$$

Therefore,

$$\operatorname{cond}\Big(\big((\mu_1+\mu_{\mathrm{m}})\,\mathrm{I}-\mathrm{A}^{\mathrm{T}}\,\mathrm{A}\big)\,\mathrm{A}^{\mathrm{T}}\,\mathrm{A}\Big)=\frac{(\mu_{\mathrm{i}})^2}{\mu_1\mu_{\mathrm{m}}}, \tag{4.21}$$

where,

$$\mu_i=\arg\min_{1\le k\le m}\left|\frac{\mu_1+\mu_m}{2}-\mu_k\right|. \tag{4.22}$$

Thus, it can be obtained:

$$\operatorname{cond}\Big(\big((\mu_1+\mu_{\mathrm{m}})\,\mathrm{I}-\mathrm{A}^{\mathrm{T}}\,\mathrm{A}\big)\,\mathrm{A}^{\mathrm{T}}\,\mathrm{A}\Big)=\frac{(\mu_{\mathrm{i}})^2}{\mu_1\mu_{\mathrm{m}}}<\frac{(\mu_1)^2}{\mu_1\mu_{\mathrm{m}}}=\frac{\mu_1}{\mu_{\mathrm{m}}}=\operatorname{cond}\big(\mathrm{A}^{\mathrm{T}}\,\mathrm{A}\big). \tag{4.23}$$

The theorem has been proved.

According to the conclusion of Theorem 1, the proposed weighting matrix $V^{-1}=((\mu_1+\mu_m)I-A^TA)$ can effectively reduce the condition number of A^TA.

4.2. Relaxation strategy

In the Landweber iteration framework, the choice of the relaxation parameter λ_n is a decisive factor that simultaneously influences the numerical accuracy of the solution and the overall rate of convergence. Previous studies have established rigorous convergence proofs for this method [7, 27]. However, conventional analyses typically focus on bounding λ_n within theoretical convergence intervals, without fully exploiting its potential for accelerating iterative performance.

In this work, we introduce a novel formulation of the iteration matrix associated with the Landweber algorithm, enabling a more explicit characterization of its spectral properties. By systematically analyzing the relationship between the spectral radius and the convergence behavior, we derive a new parameter selection methodology that integrates two complementary strategies: minimum spectral radius relaxation, which minimizes the spectral radius to ensure optimal stability, and accelerated convergence, which leverages this minimization to enhance iterative speed. This dual-strategy approach not only generalizes the theoretical convergence guarantees but also provides practical guidelines for parameter tuning in large-scale computational settings.

Theorem 2. *Let the positive eigenvalues of $A^T A$ be arranged in descending order as*

$$\mu_1 \geq \mu_2 \geq \cdots \geq \mu_m > 0. \tag{4.24}$$

Suppose there exists a unique index h such that

$$\max_{1\leq k\leq m} (\mu_1 + \mu_m - \mu_k)\mu_k = (\mu_1 + \mu_m - \mu_h)\mu_h. \tag{4.25}$$

Consider the iteration matrix

$$\begin{aligned} G(\lambda_n) =&U \operatorname{diag}((1 - \lambda_n(\mu_1 + \mu_m - \mu_1)\mu_1)I_{p_1}, \\ &\cdots, (1 - \lambda_n(\mu_1 + \mu_m - \mu_m)\mu_m)I_{p_m}, O_q)U^T, \end{aligned} \tag{4.26}$$

whose spectral radius is given by

$$\rho(G(\lambda_n)) = \max_{1\leq k\leq m} |1 - \lambda_n(\mu_1 + \mu_m - \mu_k)\mu_k| . \tag{4.27}$$

Then, the relaxation parameter minimizing the spectral radius is

$$\lambda_n = \frac{2}{(\mu_1 + \mu_m - \mu_h)\mu_h + \mu_1\mu_m}, \tag{4.28}$$

and the corresponding minimal spectral radius is

$$\rho_{\min} = \frac{(\mu_1 + \mu_m - \mu_h)\mu_h - \mu_1\mu_m}{(\mu_1 + \mu_m - \mu_h)\mu_h + \mu_1\mu_m}. \tag{4.29}$$

Proof. Since $A^T A$ is orthogonally diagonalizable, we have

$$A^T A = U diag(\mu_1 I_{p_1}, \ldots, \mu_m I_{p_m}, 0_q)U^T, \tag{4.30}$$

where U is orthogonal and p_k is the multiplicity of μ_k. So,

$$\begin{aligned} ((\mu_1 + \mu_m)I - A^T A)A^T A =&U \operatorname{diag}((\mu_1 + \mu_m - \mu_1)\mu_1 I_{p_1}, \\ &\ldots, (\mu_1 + \mu_m - \mu_m)\mu_m I_{p_m}, 0_q)U^T. \end{aligned} \tag{4.31}$$

The iteration matrix can be written as

$$\begin{aligned} G(\lambda_n) =& U\,\mathrm{diag}((1-\lambda_n(\mu_1+\mu_m-\mu_1)\mu_1)I_{p_1},\\ & \cdots,(1-\lambda_n(\mu_1+\mu_m-\mu_m)\mu_m)I_{p_m},O_q)U^T, \end{aligned} \tag{4.32}$$

whose eigenvalues are $1-\lambda_n(\mu_1+\mu_m-\mu_k)\mu_k$. Therefore, the spectral radius is

$$\rho(G(\lambda_n)) = \max_{1\le k\le m} |1-\lambda_n(1-\lambda_n(\mu_1+\mu_m-\mu_k)\mu_k)|. \tag{4.33}$$

Consider the function

$$f(\mu_k) = (\mu_1+\mu_m-\mu_k)\mu_k, \tag{4.34}$$

which is a concave quadratic function on the interval $[\mu_m,\mu_1]$. Its value at the endpoint is:

$$f(\mu_1) = (\mu_1+\mu_m-\mu_1)\mu_1 = \mu_1\mu_m \tag{4.35}$$

$$f(\mu_m) = (\mu_1+\mu_m-\mu_m)\mu_m = \mu_1\mu_m \tag{4.36}$$

and it achieves its unique maximum at $\mu=\mu_h$, that is,

$$f(\mu_h) = (\mu_1+\mu_m-\mu_h)\mu_h = \max_{\mu_k\in[\mu_m,\mu_1]} f(\mu_k). \tag{4.37}$$

So,

$$\min_k(\mu_1+\mu_m-\mu_k)\mu_k = \mu_1\mu_m, \quad \max_k(\mu_1+\mu_m-\mu_k)\mu_k = (\mu_1+\mu_m-\mu_h)\mu_h. \tag{4.38}$$

To minimize the spectral radius

$$\rho(G(\lambda_n)) = \max\{|1-\lambda_n\mu_1\mu_m|, |1-\lambda_n(\mu_1+\mu_m-\mu_h)\mu_h|\}, \tag{4.39}$$

the optimal λ_n must satisfy the balance condition

$$|1-\lambda_n\mu_1\mu_m| = |1-\lambda_n(\mu_1+\mu_m-\mu_h)\mu_h|. \tag{4.40}$$

Since $0<\mu_1\mu_m<(\mu_1+\mu_m-\mu_h)\mu_h$, the equality holds if, and only if,

$$1-\lambda_n\mu_1\mu_m = -(1-\lambda_n(\mu_1+\mu_m-\mu_h)\mu_h). \tag{4.41}$$

Solving this yields

$$\lambda_n = \frac{2}{(\mu_1+\mu_m-\mu_h)\mu_h+\mu_1\mu_m}. \tag{4.42}$$

Substituting λ_n back, the minimal spectral radius is

$$\rho_{\min} = |1-\lambda_n\mu_1\mu_m| = |1-\lambda_n(\mu_1+\mu_m-\mu_h)\mu_h| = \frac{(\mu_1+\mu_m-\mu_h)\mu_h-\mu_1\mu_m}{(\mu_1+\mu_m-\mu_h)\mu_h+\mu_1\mu_m}. \tag{4.43}$$

The theorem has been proved.

Theorem 3. *Then, the necessary and sufficient condition for convergence of the iteration is*

$$0 < \lambda_n < \frac{2}{(\mu_1 + \mu_m - \mu_h)\mu_h}, \tag{4.44}$$

and the accelerated convergence relaxation parameter λ_n should lie in the interval

$$\frac{1}{(\mu_1 + \mu_m - \mu_h)\mu_h} < \lambda_n < \frac{2}{(\mu_1 + \mu_m - \mu_h)\mu_h}. \tag{4.45}$$

Proof. The convergence of the iteration matrix $G(\lambda_n)$ requires its spectral radius to be less than one, that is,

$$\rho(G(\lambda_n)) = \max_{1\le k\le m} |1 - \lambda_n(\mu_1 + \mu_m - \mu_k)\mu_k| < 1. \tag{4.46}$$

For each $k = 1, 2, \ldots, m$, the inequality

$$|1 - \lambda_n(\mu_1 + \mu_m - \mu_k)\mu_k| < 1 \tag{4.47}$$

is equivalent to

$$-1 < 1 - \lambda_n(\mu_1 + \mu_m - \mu_k)\mu_k < 1, \tag{4.48}$$

which simplifies to

$$0 < \lambda_n(\mu_1 + \mu_m - \mu_k)\mu_k < 2. \tag{4.49}$$

Since $\mu_k > 0$ and $(\mu_1 + \mu_m - \mu_k)\mu_k > 0$ for all k, it follows that

$$0 < \lambda_n < \frac{2}{(\mu_1 + \mu_m - \mu_k)\mu_k}, \quad \forall k. \tag{4.50}$$

Hence, the overall convergence condition is

$$\begin{aligned} 0 < \lambda_n &< \min_{1\le k\le m} \frac{2}{(\mu_1 + \mu_m - \mu_k)\mu_k} \\ &= \frac{2}{\max_{1\le k\le m}(\mu_1 + \mu_m - \mu_k)\mu_k} \\ &= \frac{2}{(\mu_1 + \mu_m - \mu_h)\mu_h}. \end{aligned} \tag{4.51}$$

Next, analyze the behavior of the spectral radius in different ranges of λ_n. When

$$0 < \lambda_n^{(1)} < \frac{1}{(\mu_1 + \mu_m - \mu_h)\mu_h}, \tag{4.52}$$

its symmetric value with respect to the point $\lambda_n = \frac{1}{(\mu_1+\mu_m-\mu_h)\mu_h}$ is given by

$$\lambda_n^{(2)} = \frac{2}{(\mu_1 + \mu_m - \mu_h)\mu_h} - \lambda_n^{(1)}, \tag{4.53}$$

which satisfies

$$\frac{1}{(\mu_1 + \mu_m - \mu_h)\mu_h} < \lambda_n^{(2)} < \frac{2}{(\mu_1 + \mu_m - \mu_h)\mu_h}. \tag{4.54}$$

We have

$$1 - \lambda_n^{(1)}(\mu_1 + \mu_m - \mu_h)\mu_h > 0, \tag{4.55}$$

and, thus, the spectral radius becomes

$$\rho(G(\lambda_n^{(1)})) = \max_k \left(1 - \lambda_n^{(1)}(\mu_1 + \mu_m - \mu_k)\mu_k\right) = 1 - \lambda_n^{(1)} \min_k(\mu_1 + \mu_m - \mu_k)\mu_k. \tag{4.56}$$

Since

$$\min_k(\mu_1 + \mu_m - \mu_k)\mu_k = \mu_1\mu_m, \tag{4.57}$$

it follows that

$$\rho(G(\lambda_n^{(1)})) = 1 - \lambda_n^{(1)}\mu_1\mu_m. \tag{4.58}$$

When

$$\frac{1}{(\mu_1 + \mu_m - \mu_h)\mu_h} \leq \lambda_n^{(2)} < \frac{2}{(\mu_1 + \mu_m - \mu_h)\mu_h}, \tag{4.59}$$

if $\lambda_n^{(2)} < \frac{2}{(\mu_1+\mu_m-\mu_h)\mu_h+\mu_1\mu_m}$,then we have

$$\rho\left(G\left(\lambda_n^{(2)}\right)\right) = 1 - \lambda_n^{(2)}\mu_1\mu_m < 1 - \lambda_n^{(1)}\mu_1\mu_m = \rho\left(G\left(\lambda_n^{(1)}\right)\right), \tag{4.60}$$

if $\lambda_n^{(2)} > \frac{2}{(\mu_1+\mu_m-\mu_h)\mu_h+\mu_1\mu_m}$, we can obtain

$$\begin{aligned} \rho\left(G\left(\lambda_n^{(2)}\right)\right) &= \lambda_n^{(2)}((\mu_1 + \mu_m - \mu_h)\mu_h) - 1 \\ &= 1 - \lambda_n^{(1)}(((\mu_1 + \mu_m - \mu_h)\mu_h)) \\ &< 1 - \lambda_n^{(1)}\mu_1\mu_m = \rho\left(G\left(\lambda_n^{(1)}\right)\right). \end{aligned} \tag{4.61}$$

In summary, for $\lambda_n^{(1)} \in \left(0, \frac{1}{(\mu_1+\mu_m-\mu_h)\mu_h}\right)$ and $\lambda_n^{(2)} = \frac{2}{(\mu_1+\mu_m-\mu_h)\mu_h} - \lambda_n^{(1)} \in \left(\frac{1}{(\mu_1+\mu_m-\mu_h)\mu_h}, \frac{2}{(\mu_1+\mu_m-\mu_h)\mu_h}\right)$, the following inequality holds:

$$\rho(G(\lambda_n^{(1)})) > \rho(G(\lambda_n^{(2)})). \tag{4.62}$$

This completes the proof.

4.3. Algorithm flow

Based on the preceding analysis of the preconditioned Landweber iteration algorithm for conical surface fitting, the proposed method effectively addresses the numerical instability and ill-conditioning issues inherent to traditional LM algorithms in cone fitting applications. Algorithm 1 presents the complete computational procedure, in which the maximum and minimum singular values, σ_1 and σ_m, are obtained by performing a singular value decomposition (SVD) of $J^{(k)}$. The corresponding spectral parameters are then set as $\mu_1 = \sigma_1^2$ and $\mu_m = \sigma_m^2$.

Algorithm 1 Preconditioned Landweber iteration for conical surface fitting

```
**Input:** Point cloud $\{p_i\}_{i=1}^{N}$, initial parameters $\theta^{(0)} = (c^{(0)}, a^{(0)}, \alpha^{(0)})$, maximum iterations $K_{\max}$, tolerance $\epsilon$
**Output:** Optimal parameters $\theta^* = (c^*, a^*, \alpha^*)$
**for** $k = 0, 1, \ldots, K_{\max}$ **do**
    Compute residual vector $r^{(k)} \in \mathbb{R}^N$:
$r_i^{(k)} = \frac{\|(p_i - c^{(k)}) \times a^{(k)}\| - \|p_i - c^{(k)}\| \sin \alpha^{(k)}}{\sqrt{1 + \sin^2 \alpha^{(k)}}}, \quad \forall i$
    Construct Jacobian matrix $J^{(k)} \in \mathbb{R}^{N \times 7}$:
$\frac{\partial r_i}{\partial c}, \frac{\partial r_i}{\partial a}, \frac{\partial r_i}{\partial \alpha}$
    Build preconditioning matrix $V^{-1} = (\mu_1 + \mu_m)I - J^{(k)T} J^{(k)}, W = I$
    Calculate step size:
$\lambda_k = \frac{2}{(\mu_1 + \mu_m - \mu_h)\mu_h + \mu_1 \mu_m}$
    Update parameters:
$\theta^{(k+1)} = \theta^{(k)} + \lambda_k((\mu_1 + \mu_m)I - J^{(k)T} J^{(k)}) J^{(k)T} (r^{(k)} - J^{(k)} \theta^{(k)})$
    Apply constraints:
  • Normalize $a^{(k+1)} \leftarrow a^{(k+1)} / \|a^{(k+1)}\|_2$
  • Project $\alpha^{(k+1)}$ to $(0, \pi/2)$
    **if** $\|\theta^{(k+1)} - \theta^{(k)}\|_2 < \epsilon$ **then**
        **break**
    **end if**
**end forreturn** $\theta^* = \theta^{(k+1)}$
```

4.4. Complexity analysis

In the conical fitting problem, the Jacobian matrix $J \in \mathbb{R}^{m \times p}$ consists of the partial derivatives of residuals with respect to the parameters, where m is the number of data points and p denotes the parameter dimension. As derived above, each row of J corresponds to a single data point and includes partial derivatives with respect to the cone vertex position $c \in \mathbb{R}^3$, direction vector $a \in \mathbb{R}^3$, and the opening angle α. Since each row computation involves only vector differences, cross products, and basic algebraic operations, the complexity for calculating the Jacobian for one point is constant, leading to an overall complexity of $O(m)$ for the entire Jacobian matrix.

The classical LM algorithm requires explicit construction and computation of the matrix $J^T J$ in each iteration, with a computational complexity of approximately $O(mp^2)$, followed by matrix decomposition for solving the linear system, incurring an additional $O(p^3)$ cost. Although the parameter dimension p is relatively small (typically around 7), the computational burden for forming $J^T J$ grows linearly with the data size m, and the memory requirements for matrix factorization limit its scalability for large-scale problems.

In contrast, the proposed preconditioned Landweber iterative algorithm exploits the sparsity of the Jacobian matrix and avoids expensive matrix factorizations. Each iteration mainly consists of several sparse matrix-vector multiplications, resulting in a computational complexity of approximately $O(mp)$, where p denotes the average number of nonzero elements per row. This complexity is significantly lower than that of the LM algorithm, especially for large-scale sparse datasets. Moreover, the preconditioned Landweber method demands substantially less memory, features a simpler structure, and is well suited for parallel and distributed implementation, making it particularly suitable for resource-constrained environments and massive data processing tasks.

In summary, although the LM algorithm achieves faster local convergence when the parameter dimension is small, its computational and memory costs increase rapidly with data size, limiting its applicability to large-scale conical fitting problems. The preconditioned Landweber iterative algorithm significantly reduces computational complexity through efficient sparse matrix operations, enhancing scalability and practical usability. It thus offers an effective and efficient solution for large-scale three-dimensional conical surface fitting.

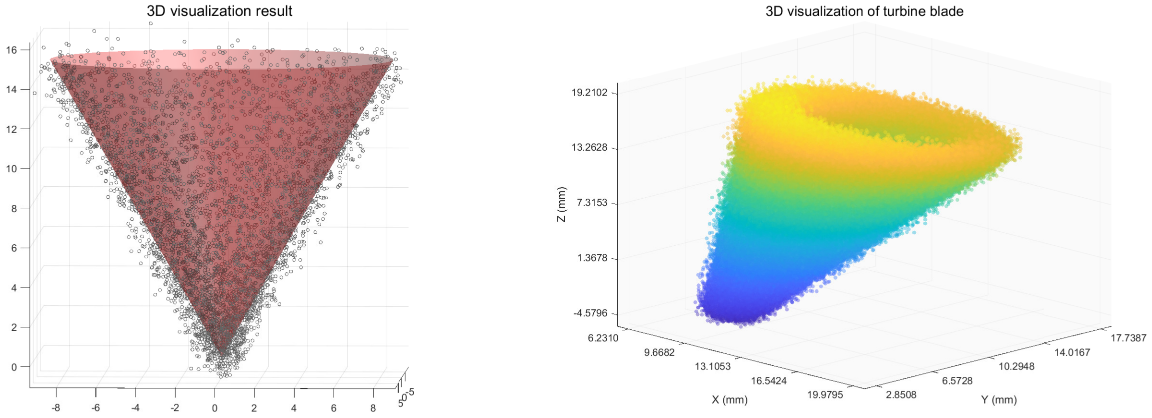
5. Experimental evaluation

5.1. Experimental setup

Experiments were conducted on a Lenovo Y7000 laptop (Intel i7-9750H, 16GB RAM) using MATLAB R2021a. The test datasets included synthetic data and industrial scan data. The synthetic data consisted of 500,000 point cloud points generated with vertex position $c = [0, 0, 0]$ mm, axial direction $a = [0, 0, 1]$, and half-apex angle $\alpha = 30°$, contaminated with Gaussian noise ($\sigma = 0.5$ mm) and 30% outliers. The industrial scan data comprised turbine blade point clouds with·520,000 points, containing surface waviness noise and non-uniform sampling. Comparative algorithms included the classical LM algorithm and the proposed preconditioned Landweber optimization method (Landweber), evaluated through three metrics: accuracy, efficiency, and robustness.

5.2. Result analysis

Figure 2 presents the visualization results of conical surface fitting on both the synthetic and industrial turbine-blade datasets, showing that the preconditioned Landweber method delivers stable and accurate fitting even in the presence of measurement noise. The quantitative comparisons in Table 1 (synthetic data) and Table 2 (industrial data) reveal that the proposed approach consistently outperforms the conventional LM algorithm in terms of Root Mean Square Error(RMSE), maximum error, computation time, iteration count, vertex error, axial error, and angular error.



(a) synthesized data

(b) Turbine blade data

Figure 2. 3D fitting visualization of synthetic and industrial data.

Table 1. Comparison of quantitative indicators on synthetic data.

Method	RMSE	Max error	Time	Iterations	Vertex error	Axis error	Angle error
LM	0.97298	6.7368	0.28052	54	4.6592	89.897	59
Landweber	**0.017756**	**0.039024**	**0.10456**	**26**	**2.0493**	**45.931**	**32.454**

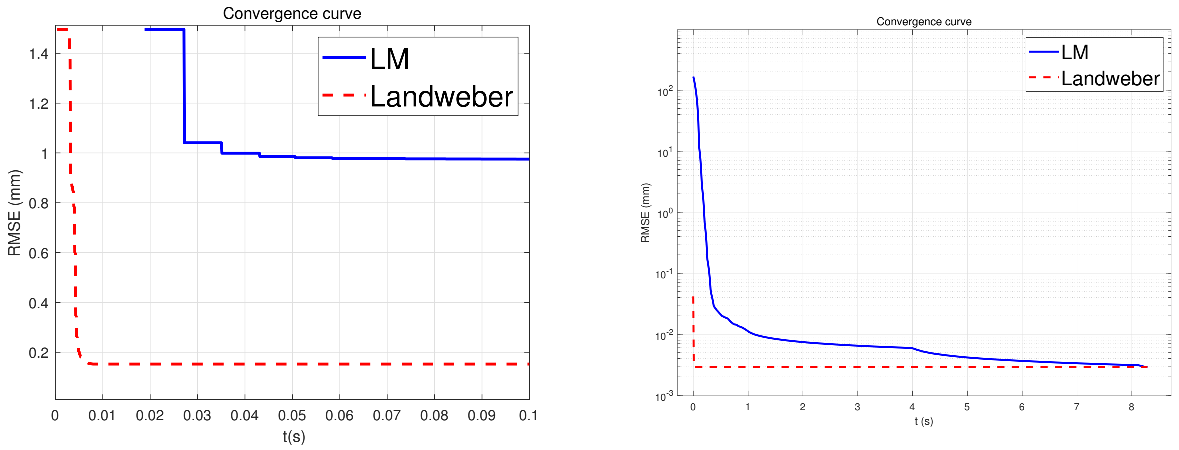
Table 2. Comparison of quantitative indicators on industrial data.

Method	RMSE	Max error	Time	Iterations	Vertex error	Axis error	Angle error
LM	8.883×10^{-4}	9.532×10^{-3}	15.951	35	12.9	156.39	74.898
Landweber	$\mathbf{5.255 \times 10^{-4}}$	$\mathbf{2.226 \times 10^{-3}}$	**4.256**	**26**	**7.443**	**101.5**	**74**

For the synthetic dataset, RMSE is reduced by 98%, the number of iterations decreases by 51.9%, and computation time is shortened by 62.7%. For the industrial dataset, RMSE decreases by 41%, the number of iterations is reduced by 25%, and computation time is shortened by 73%. These improvements not only indicate higher numerical accuracy but also reflect a substantial gain in computational efficiency, which is critical for large-scale 3D reconstruction tasks.

The RMSE–time curve in Figure 3 exhibits a steep initial drop followed by rapid stabilization, indicating that the preconditioned Landweber method suppresses noise effects more effectively and reaches a near-optimal solution in fewer iterations. Similarly, the RMSE–iteration curve in Figure 4 shows a consistently faster error decay rate compared with LM, with the gap between the two methods widening as iterations proceed. This trend demonstrates that the proposed method benefits from a more favorable convergence trajectory, which leads to lower residual errors in both synthetic and industrial scenarios.

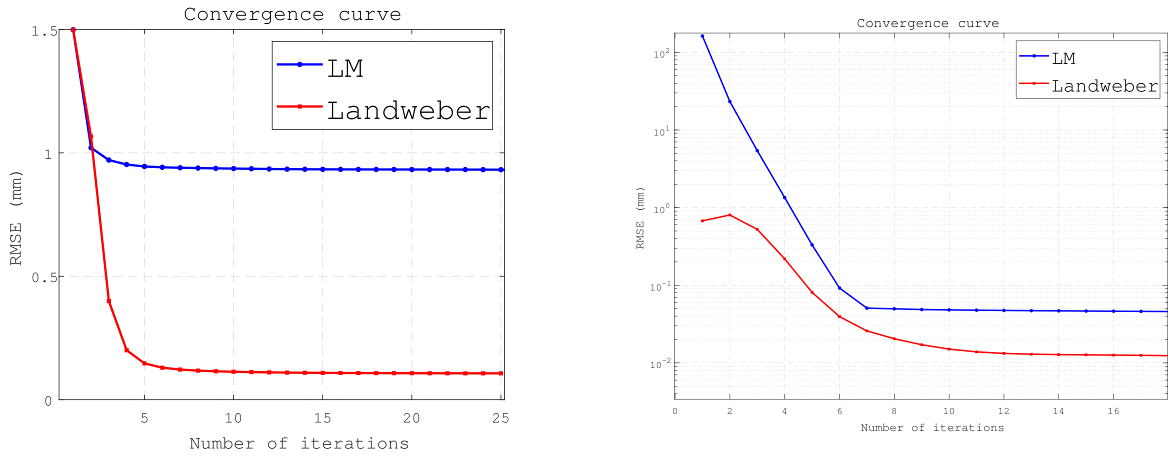
Overall, these results confirm that the preconditioned Landweber method offers high accuracy, fast convergence, and robust performance across diverse datasets, providing a reliable and efficient solution for complex 3D point cloud processing.



(a) synthesized data

(b) Turbine blade data

Figure 3. A graph of RMSE changes over time for synthetic and industrial data.



(a) synthesized data

(b) Turbine blade data

Figure 4. A chart showing the RMSE of synthetic data and industrial data varying with the number of iterations.

6. Conclusions

This work developed a preconditioned Landweber iterative scheme for nonlinear least-squares fitting of conical surfaces in 3D point cloud data. By introducing a geometric distance-based residual formulation, deriving explicit Jacobian expressions, and designing a spectral preconditioning strategy, the method effectively resolves numerical instabilities and convergence stagnation induced by ill-conditioned Hessian matrices in conventional LM optimization. Theoretical analysis established that the preconditioned iteration admits a provable linear convergence rate under an optimally chosen relaxation factor, highlighting its mathematical rigor as well as computational efficiency.

Beyond its immediate contribution to conical surface fitting, this study also provides insights into the design of preconditioned iterative solvers for nonlinear least-squares problems. The spectral compression mechanism ensures favorable conditioning of the normal matrix, which may be generalized to other classes of nonlinear inverse problems. In particular, the framework reveals a promising connection between iterative regularization theory and nonlinear geometric model fitting, bridging methodologies from numerical analysis and computational geometry.

Comprehensive experiments on synthetic and industrial datasets verified the effectiveness of the proposed approach in terms of convergence speed, accuracy, and robustness to noise. The results not only confirm its superiority over LM-based methods but also demonstrate the practical relevance of iterative preconditioning for real-world 3D reconstruction tasks.

Future work will extend the proposed framework in several directions: (i) generalization to broader families of quadric and higher-order algebraic surfaces, (ii) large-scale point cloud optimization under distributed or parallel computing settings, (iii) hybridization with adaptive regularization techniques to further enhance stability and convergence guarantees, and (iv) exploration of synergies with deep learning approaches, such as using neural networks to provide high-quality initial parameter estimates or embedding our optimization method as a differentiable layer within end-to-end learning pipelines. These directions aim to consolidate the role of preconditioned Landweber iteration as a versatile and mathematically grounded approach within the broader field of computational and applied mathematics.

Use of AI tools declaration

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

Conflict of interest

The authors declare there is no conflict of interest.

References

1. G. Lukács, A. D. Marshall, R. R. Martin, Geometric least-squares fitting of spheres, cylinders, cones and tori, *RECCAD, Deliverable Document*, 1997.
2. D. Marshall, G. Lukacs, Ralph Martin, Robust segmentation of primitives from range data in the presence of geometric degeneracy, *IEEE Trans. Pattern Anal. Mach. Intell.*, **23** (2001), 304–314. https://doi.org/10.1109/34.910883
3. C. M. Shakarji, Least-squares fitting algorithms of the nist algorithm testing system, *J. Res. Natl. Inst. Stand. Technol.*, **103** (1998), 633–641. https://doi.org/10.6028/jres.103.043
4. K. F. Mulchrone, D. Pastor-Galan, G. Gutierrez-Alonso, Mathematica code for least-squares cone fitting and equal-area stereonet representation, *Comput. Geosci.*, **54** (2013), 203–210. https://doi.org/10.1016/j.cageo.2013.01.005
5. R. B. Rusu, Semantic 3d object maps for everyday manipulation in human living environments, *KI Kunstliche Intell.*, **24** (2010), 345–348. https://doi.org/10.1007/s13218-010-0059-6
6. R. Courant, *Differential and Integral Calculus, Volume II*, John Wiley & Sons, 2011.
7. M. Jiang, G. Wang, Convergence studies on iterative algorithms for image reconstruction, *IEEE Trans. Med. Imaging*, **22** (2003), 569–579. https://doi.org/10.1109/TMI.2003.812253
8. X. Jiang, D. Cheng, A novel parameter decomposition approach to faithful fitting of quadric surfaces, *Pattern Recognit.*, (2005), 168–175. https://doi.org/10.1007/11550518_21
9. J. Deng, G. Liu, L. Yue, A novel method of cone fitting based on 3d point cloud, *Appl. Mech. Mater.*, **722** (2015), 321–326. https://doi.org/10.4028/www.scientific.net/AMM.722.32
10. R. Schnabel, R. Wahl, R. Klein, Efficient ransac for point-cloud shape detection, *Comput. Graphics Forum*, **26** (2007), 214–226. https://doi.org/10.1111/j.1467-8659.2007.01016.x
11. G. Han, G. Qu, Q. Wang, Weighting algorithm and relaxation strategies of the landweber method for image reconstruction, *Math. Probl. Eng.*, (2018), 5674647. https://doi.org/10.1155/2018/5674647
12. S. Xie, G. Qu, W. Li, A preconditioned landweber iteration-based bundle adjustment for large-scale 3d reconstruction, *Commun. Nonlinear Sci. Numer. Simul.*, **130** (2024), 107770. https://doi.org/10.1016/j.cnsns.2023.107770
13. G. Lukács, R. Martin, D. Marshall, Faithful least-squares fitting of spheres, cylinders, cones and tori for reliable segmentation, in *Computer Vision — ECCV'98*, (1998), 671–686. https://doi.org/10.1007/BFb0055697

14. H. Tian, S. Guo, Research on algorithm of quadric surface fitting in reverse modeling for mechanical parts, *J. Southwest Jiaotong Univ.*, **42** (2007), 553–557. Available from: https://xnjdxb.swjtu.edu.cn/en/article/id/11091.

15. C. Sun, B. Zhu, W. Wang, Optimizing conical reconstruction of linear point clouds, *J. Comput.-Aided Des. Comput. Graphics*, **22** (2010), 1324–1330. https://doi.org/10.3724/SP.J.1089.2010.10979

16. X. Gong, J. Wang, Fitting of circular conical surface, *Geotech. Invest. Surv.*, **38** (2010), 79–82.

17. A. Karambakhsh, B. Sheng, P. Li, H. Li, J. Kim, Y. Jung, et al., Sparsevoxnet: 3D object recognition with sparsely aggregation of 3-d dense blocks, *IEEE Trans. Neural Networks Learn. Syst.*, **35** (2022), 532–546. https://doi.org/10.1109/TNNLS.2022.3175775

18. M. P. M. Ram, T. R. Kurfess, T. M. Tucker, Least-squares fitting of analytic primitives on a GPU, *J. Manuf. Syst.*, **27** (2008), 130–135. https://doi.org/10.1016/j.jmsy.2008.07.004

19. C. M. Shakarji, V. Srinivasan, On algorithms and heuristics for constrained least-squares fitting of circles and spheres to support standards, *J. Comput. Inf. Sci. Eng.*, **19** (2019), 031012. https://doi.org/10.1115/1.4043226

20. S. Gao, G. Qu, Filter-based landweber iterative method for reconstructing the light field, *IEEE Access*, **8** (2020), 138340–138349. https://doi.org/10.1109/ACCESS.2020.3012535

21. Y. Censor, T. Elfving, Block-iterative algorithms with diagonally scaled oblique projections for the linear feasibility problem, *SIAM J. Matrix Anal. Appl.*, **24** (2002), 40–58. https://doi.org/10.1137/S089547980138705X

22. D. J. Parker, Introduction to inverse problems in imaging, *Comput. Phys. Commun.*, **120** (1999), 269–270. https://doi.org/10.1016/S0010-4655(99)00257-X

23. G. Qu, C. Wang, R. Kang, Reconstruction of band-limited signals from uniform random samples, *Adv. Math.*, **41** (2012).

24. G. Qu, M. Jiang, Y. Yang, Convergence results of landweber iterations for linear systems, *Acta Math. Appl. Sin.*, **30** (2014), 111–118. https://doi.org/10.1007/s10255-013-0299-y

25. G. Qu, M. Jiang, Landweber scheme for compact operator equation in hilbert space and its applications, *Commun. Numer. Methods Eng.*, **25** (2009), 771–786. https://doi.org/10.1002/cnm.1196

26. A. Kirsch, *An Introduction to the Mathematical Theory of Inverse Problems*, Springer, 2021.

27. G. Qu, C. Wang, M. Jiang, Necessary and sufficient convergence conditions for algebraic image reconstruction algorithms, *IEEE Trans. Image Process.*, **18** (2008), 435–440. https://doi.org/10.1109/TIP.2008.2008076

AIMS Press