

高阶收敛的加速最小二乘渐进迭代逼近

曹一菲, 彭兴璇*

辽宁师范大学数学学院, 辽宁 大连

收稿日期: 2026年5月16日; 录用日期: 2026年6月7日; 发布日期: 2026年6月16日

摘要

最小二乘渐进迭代逼近(progressive iterative approximation for least square fitting, LSPIA)是一种应用于数据拟合问题的迭代算法。针对经典LSPIA算法具有收敛速度较慢的问题, 本文引入一种求解非奇异矩阵逆的具有8阶收敛性的迭代算法, 与经典LSPIA算法相结合, 提出了一类加速LSPIA算法。首先对数据点进行参数化; 然后利用加速LSPIA算法构造拟合曲线序列, 并证明该序列的极限曲线收敛至给定数据点的最小二乘拟合结果。数值实例进一步表明, 相较于经典LSPIA算法, 本文方法能显著提升收敛速度, 在相同终止误差条件下, 可减少迭代次数和运行时间。

关键词

渐进迭代逼近, 最小二乘拟合, 近似逆, B样条

Accelerated Least Squares Progressive-Iterative Approximation of Higher Order of Convergence

Yifei Cao, Xingxuan Peng*

School of Mathematics, Liaoning Normal University, Dalian Liaoning

Received: May 16, 2026; accepted: June 7, 2026; published: June 16, 2026

Abstract

The progressive iterative approximation for least square fitting (LSPIA) is an iterative algorithm, which is commonly applied to data fitting problems. To address the issue of slow convergence in the classical LSPIA method, this paper integrates an iterative algorithm for computing the inverse of non-singular matrices, which possesses eighth-order convergence. Combined with the classical LSPIA

*通讯作者。

approach, a class of accelerated LSPIA algorithms is proposed. Firstly, the data points are parameterized. Then, an accelerated LSPIA algorithm is employed to generate a sequence of fitting curves. This sequence converges to the least squares fitting result for the given data points. Experimental results demonstrate that the proposed algorithm achieves a significantly accelerated convergence rate compared to the classical LSPIA method, and it can reduce the number of iterations and running time under the same termination error condition.

Keywords

Progressive-Iterative Approximation, Least Squares Fitting, Approximate Inverse, B-Spline

Copyright © 2026 by author(s) and Hans Publishers Inc.

This work is licensed under the Creative Commons Attribution International License (CC BY 4.0).

<http://creativecommons.org/licenses/by/4.0/>



Open Access

1. 引言

数据拟合是一种经典的几何设计方法, 在计算机辅助设计与计算机图形学等领域均有广泛应用。传统的拟合方法通常要求解大规模线性方程组, 计算成本高且缺乏灵活性。近年来, 渐进迭代逼近 (progressive iterative approximation, PIA) 由于其在数据拟合中所展现出的显著几何直观性而备受关注。

PIA 方法最早可追溯至齐东旭等人[1]的工作, 他们在研究均匀三次 B 样条曲线拟合的迭代特性时, 首次提出了一种盈亏修正算法。de Boor [2]也从理论上严格证明了该算法的收敛性。Lin 等[3]在此基础上进一步拓展, 深入探讨了非均匀三次 B 样条曲线与曲面的情况, 并成功证明了其同样具备盈亏修正特性。随后, Lin 等[4]进一步指出该性质适用于所有标准全正基生成的混合曲线曲面, 并正式将此类方法命名为 PIA。该方法的核心思想在于通过迭代计算求解最优控制多边形, 使该多边形生成的曲线对应点与给定数据点的平方距离和最小。

经典 PIA 方法要求控制点与数据点在数量上严格相等, 这一约束条件导致其在处理大规模数据时收敛效率低下。考虑到这个问题, Deng 等[5]提出了一种最小二乘渐进迭代逼近 (progressive iterative approximation for least square fitting, LSPIA) 算法, 该算法能够高效稳定地拟合大规模数据点, 并且在给定数据点处, 其生成的曲线曲面序列严格收敛到最小二乘拟合结果。此后, 许多学者对 LSPIA 方法及其应用进行研究。为了局部调整部分控制点, Wang [6]提出了一种扩展的 LSPIA 方法, 王曾珍等[7]进一步提出了一种带互异权值的 LSPIA 算法。周雅情等[8]提出了一种基于关键点的 LSPIA 算法, 该方法通过识别局部曲率最大点与极端曲率点等特征点来驱动拟合过程, 能更准确地揭示曲线的几何特征。Hamza 等[9]提出了基于高斯-赛德尔迭代法的 GS-LSPIA 算法。蒋旖旋等[10]提出了基于共轭梯度法的 CG-LSPIA 算法。Wu 和 Liu [11]根据切比雪夫多项式的根自适应选择步长, 提出了一种异步 LSPIA 方法; 在文[12]中, 还基于 Polyak 和 Nesterov 动量技术, 提出了 LSPIA 的两种改进形式。针对几何配点法中的拟合问题, Jiang 和 Lin [13]提出了一种等几何最小二乘渐进迭代逼近法。Hu 等[14]将计算 Moore-Penrose 广义逆的迭代方法与经典 LSPIA 方法相结合, 构建了一类用于曲线拟合的迭代方法, 具有更快的收敛速度。

在经典 LSPIA 迭代算法中, 计算差向量的权值始终为固定值, 随着迭代次数增加, 迭代矩阵谱半径下降速度较慢, 引入逆矩阵可以使得迭代矩阵谱半径迅速下降并趋于 0。在文[15]中提出了具有 8 阶收敛性的矩阵求逆迭代算法, 本文将它与 LSPIA 迭代格式相结合, 提出了一类具有更快收敛速度的加速 LSPIA 算法, 并严格证明了该算法的收敛性, 同时证明了该算法的收敛阶数至少达到 8 阶。最后利用数值实例

对 LSPIA 算法和本文所提方法进行误差对比分析。数值实例表明, 本文所提方法的收敛速度显著提升, 验证了本文方法的可行性。

2. 预备知识

在经典 LSPIA 算法的改进研究中, 高效求解矩阵逆是一个关键问题。本节介绍一种求解矩阵逆的 8 阶收敛的迭代算法[15], 用于快速近似非奇异矩阵的逆。相较于已有的同类方法[16]-[18], 该方法在达到 8 阶收敛时仅需 7 次矩阵乘法, 其信息效率指数 $HIEI = \rho/k$ (ρ 为收敛阶, k 为矩阵乘法次数) 约为 1.143, 高于上述对比方法。这意味着在相同计算量下, 该迭代格式能提供更高精度的逆近似。同时, 该方法具有优异的数值稳定性, 对中等病态矩阵仍能稳定迭代, 不会出现发散或震荡, 适用于大规模稀疏矩阵的逆近似问题。

给定一个可逆矩阵 A , 想要构造一个迭代序列 $\{V_n\}$ 使其收敛于 A^{-1} , 则有迭代格式

$$V_{n+1} = -\frac{1}{4}V_n W_n (-4I + K_n W_n), n = 0, 1, 2, \dots, \quad (1)$$

其中

$$\begin{cases} W_n = 9I + K_n (-16I + K_n (14I + K_n (-6I + K_n))) \\ K_n = AV_n \end{cases}$$

(1)式定义了一种矩阵求逆的 8 阶迭代方法, 当满足引理 1 时迭代格式收敛。

引理 1 [15] 设 $A = [a_{ij}]_{m \times m}$ 是一个非奇异矩阵, 如果 V_0 满足

$$\|I - AV_0\| < 1,$$

则迭代格式(1)收敛到 A^{-1} , 收敛阶为 8 阶。

3. 加速的 LSPIA 算法

针对一组给定的有序数据点列 $\{Q_j | j = 0, 1, \dots, m\}$, 先进行参数化, 为每个点赋予一个参数值 $t_j, j = 0, 1, \dots, m$, 满足 $0 = t_0 < t_1 < \dots < t_m = 1$ 。选取 $P^0 = \{P_i^0, i = 0, 1, \dots, n\} (m \geq n)$ 作为初始控制点列, 从而构建一条初始混合曲线

$$C^0(t) = \sum_{i=0}^n B_i(t) P_i^0, t \in [t_0, t_m],$$

其中 $\{B_i(t), i = 0, 1, \dots, n\}$ 是一组 B 样条基, 其配置矩阵为

$$B = \begin{bmatrix} B_0(t_0) & B_1(t_0) & \dots & B_n(t_0) \\ B_0(t_1) & B_1(t_1) & \dots & B_n(t_1) \\ \vdots & \vdots & \ddots & \vdots \\ B_0(t_m) & B_1(t_m) & \dots & B_n(t_m) \end{bmatrix}. \quad (2)$$

在第 1 次迭代时, 首先对每一个数据点 Q_j 分别构造相应的差向量

$$\delta_j^0 = Q_j - C^0(t_j), j = 0, 1, \dots, m,$$

其中 $P^0 = [P_0^0, P_1^0, \dots, P_n^0]^T$ 为初始控制点, $Q = [Q_0, Q_1, \dots, Q_m]^T$ 为给定数据点。

假定

$$Z^0 = \omega I,$$

其中 I 为 $n+1$ 阶单位矩阵, 权因子 ω 满足

$$0 < \omega < 2/\lambda_0, \quad (3)$$

其中 λ_0 是矩阵 $B^T B$ 的最大特征值。

根据预备知识中的公式(1), 定义矩阵多项式

$$\begin{cases} g(X, Y) = 9I + XY(-16I + XY(14I + XY(-6I + XY))) \\ f(X, Y, \varphi) = -\frac{1}{4}X\varphi(-4I + XY\varphi) \end{cases}, \quad (4)$$

其中 X, Y 都为 $n+1$ 阶方阵。

令 $Z^1 = f(Z^0, B^T B, g(Z^0, B^T B))$, 进而构造与控制点 P^0 对应的调整向量

$$\Delta^0 = Z^1 B^T \delta^0,$$

其中 $\delta^0 = [\delta_0^0, \delta_1^0, \dots, \delta_n^0]$, B 是配置矩阵(2)。则得到新的控制点 $P^1 = P^0 + \Delta^0$ 和新的迭代曲线

$$C^1(t) = \sum_{i=0}^n B_i(t) P_i^1, t \in [t_0, t_m].$$

现假设迭代 k 次后生成的曲线为 $C^k(t)$, 其控制点为 $P^k = \{P_i^k\}_{i=0}^n$ 。根据以下迭代格式

$$\begin{cases} \delta^k = Q - BP^k \\ Z^{k+1} = f(Z^k, B^T B, g(Z^k, B^T B)) \\ \Delta^k = Z^{k+1} B^T \delta^k \\ P^{k+1} = P^k + \Delta^k \end{cases}, \quad (5)$$

第 $k+1$ 次迭代后生成的曲线可表示为

$$C^{k+1}(t) = \sum_{i=0}^n B_i(t) P_i^{k+1}, t \in [t_0, t_m],$$

重复以上过程, 可以得到一系列拟合曲线 $\{C^k(t), k=0, 1, \dots\}$ 。

4. 算法的收敛性分析

本节将证明上述加速 LSPIA 算法是收敛的。

定理 1 当权因子 ω 满足公式(3)时, 由迭代格式(5)生成的曲线序列 $\{C^k(t), k=0, 1, \dots\}$ 是收敛的。并且极限曲线 $P^\infty(t)$ 将严格收敛于初始数据点的最小二乘拟合结果, 且该方法的拟合误差呈超线性快速下降。

证明 根据迭代格式(5), 有

$$P^{k+1} = P^k + Z^{k+1} B^T (Q - BP^k), \quad (6)$$

且 Z^k 满足

$$I - Z^{k+1} B^T B = \frac{1}{4} \left[(I - Z^k B^T B)^8 + 2(I - Z^k B^T B)^9 + (I - Z^k B^T B)^{10} \right], \quad (7)$$

综合式(6) (7), 可以得到

$$\begin{aligned}
& P^{k+1} - (B^T B)^{-1} B^T Q \\
&= P^k + Z^{k+1} (B^T Q - B^T B P^k) - (B^T B)^{-1} B^T Q \\
&= (I - Z^{k+1} B^T B) \left[P^k - (B^T B)^{-1} B^T Q \right] \\
&= (I - Z^{k+1} B^T B) (I - Z^k B^T B) \left[P^{k-1} - (B^T B)^{-1} B^T Q \right] \\
&= \dots \\
&= \prod_{i=1}^{k+1} (I - Z^i B^T B) \left[P^0 - (B^T B)^{-1} B^T Q \right] \\
&= \prod_{i=0}^k \frac{1}{4} \left[(I - Z^i B^T B)^8 + 2(I - Z^i B^T B)^9 + (I - Z^i B^T B)^{10} \right] \left[P^0 - (B^T B)^{-1} B^T Q \right],
\end{aligned}$$

假设 $D_k = \prod_{i=0}^k \frac{1}{4} \left[(I - Z^i B^T B)^8 + 2(I - Z^i B^T B)^9 + (I - Z^i B^T B)^{10} \right]$, 上式可改写为

$$P^{k+1} - (B^T B)^{-1} B^T Q = D_k \left[P^0 - (B^T B)^{-1} B^T Q \right]. \quad (8)$$

假设 I 为 $n+1$ 阶单位矩阵, $D = I - Z^0 B^T B = I - \omega B^T B$. 令 $\{\lambda_i(D), i=0, 1, \dots, n\}$ 是矩阵 D 按非升序排列的所有特征值, 则 $\lambda_i(D) = 1 - \omega \lambda_i$. 由(3)式, 可以得到 $0 < \omega \lambda_i < 2\lambda_i / \lambda_0 \leq 2$, 故 $-1 < 1 - \omega \lambda_i \leq 1, i=0, 1, \dots, n$, 于是 $-1 < \lambda_i(D) \leq 1, i=0, 1, \dots, n$. 由于 $B^T B$ 是实对称矩阵, 因此 D 是实对称矩阵, 从而矩阵 D 的谱范数 $\|D\| = \rho(D) = \|I - Z^0 B^T B\| < 1$.

令 $\|D_k\|$ 是 D_k 的谱范数, 对于 D_k , 则有

$$\begin{aligned}
\|D_k\| &= \left\| \prod_{i=0}^k \frac{1}{4} \left[(I - Z^i B^T B)^8 + 2(I - Z^i B^T B)^9 + (I - Z^i B^T B)^{10} \right] \right\| \\
&\leq \prod_{i=0}^k \frac{1}{4} \left\| (I - Z^i B^T B)^8 + 2(I - Z^i B^T B)^9 + (I - Z^i B^T B)^{10} \right\| \\
&\leq \prod_{i=0}^k \frac{1}{4} \left[\|I - Z^i B^T B\|^8 + 2\|I - Z^i B^T B\|^9 + \|I - Z^i B^T B\|^{10} \right] \\
&\leq \prod_{i=0}^k \|I - Z^i B^T B\|^8 \\
&= \|I - Z^0 B^T B\|^8 \|I - Z^1 B^T B\|^8 \dots \|I - Z^k B^T B\|^8 \\
&\leq \|I - Z^0 B^T B\|^8 \|I - Z^0 B^T B\|^{8^2} \dots \|I - Z^0 B^T B\|^{8^{k+1}} \\
&= \|I - Z^0 B^T B\|^{8^{(k+1)-1}},
\end{aligned}$$

由于 $\|I - Z^0 B^T B\| < 1$, 所以 $k \rightarrow \infty$ 时, $\|D_k\| \rightarrow 0$, 则

$$\lim_{k \rightarrow \infty} D_k = 0_{n+1},$$

其中 0_{n+1} 为 $n+1$ 阶零矩阵.

根据式(8), 可知当 $k \rightarrow \infty$ 时, 有

$$P^\infty = (B^T B)^{-1} B^T Q,$$

等价于 $(B^T B) P^\infty = B^T Q$. 这就意味着迭代格式(5)生成的曲线序列 $\{C^k(t), k=0, 1, \dots\}$ 是收敛的, 且其极限

曲线即为给定初始数据点的最小二乘拟合结果。

令 $\varepsilon_{k+1} = (B^T B)^{-1} - Z^{k+1}$, 则 $\varepsilon_{k+1} B^T B = I - Z^{k+1} B^T B$, 这意味着

$$\varepsilon_{k+1} B^T B = \frac{1}{4} \left[(\varepsilon_k B^T B)^8 + 2(\varepsilon_k B^T B)^9 + (\varepsilon_k B^T B)^{10} \right],$$

因此, 可以得出

$$\varepsilon_{k+1} = \frac{1}{4} \left[\varepsilon_k (\varepsilon_k B^T B)^7 + 2\varepsilon_k (\varepsilon_k B^T B)^8 + \varepsilon_k (\varepsilon_k B^T B)^9 \right],$$

令 $\|\varepsilon_{k+1}\|$ 是 ε_k 的谱范数, 可以得到

$$\|\varepsilon_{k+1}\| \leq \left(\frac{1}{4} \left[\|B^T B\|^7 + 2\|B^T B\|^8 \|\varepsilon_k\| + \|B^T B\|^9 \|\varepsilon_k\|^2 \right] \right) \|\varepsilon_k\|^8,$$

由此证明, 矩阵迭代格式 $Z^{k+1} = f(Z^k, B^T B, g(Z^k, B^T B))$ 具有 8 阶的收敛性, 因此算法拟合误差在实际计算中则呈现更快的超线性下降, 于是定理得证。

权值满足式(3)时的加速 LSPIA 迭代方法是收敛的。定理 2 中给出了最优权值, 以实现本节中的快速收敛速度。

定理 2 当权因子 $\omega_{ib} = \frac{2}{\lambda_0 + \lambda_n}$ 时, 加速的 LSPIA 算法收敛速度最快, 并且此时迭代矩阵 D 的谱半径

$$\rho(D) = \frac{\lambda_0 - \lambda_n}{\lambda_0 + \lambda_n}.$$

证明 类似于文献[5]中定理 3.1 的证明, 可以得到结果。

5. 数值实例

在本节中, 我们将以三次 B 样条基函数为例, 通过数值实例来验证上述迭代算法的收敛性和效率。

5.1. 权值选取

虽然定理 2 中给出了加速 LSPIA 算法在理论上提供最快收敛速度的权因子 $\omega_{ib} = \frac{2}{\lambda_0 + \lambda_n}$, 但在实践中, 它需要大量的计算来计算最大和最小的特征值。为了避免特征值的计算, 我们利用文献[5]中的方法来定义权因子的值。

对于(2)式定义的矩阵 B , 令 $B^T B = \{b_{ij}\}_{i,j=0}^{m,n}$, 其中 $b_{ij} = \sum_{k=0}^m B_i(t_k) B_j(t_k)$ 。结合 $\sum_{i=0}^n B_i(t_k) = 1$, 则有

$$\sum_{j=0}^n b_{ij} = \sum_{j=0}^n \left[\sum_{k=0}^m B_i(t_k) B_j(t_k) \right] = \sum_{k=0}^m B_i(t_k) \left(\sum_{j=0}^n B_j(t_k) \right) = \sum_{k=0}^m B_i(t_k) = c_i.$$

这意味着 c_i 是 $B^T B$ 的第 i 行元素之和。因此 $\lambda_0 \leq \max_i \{c_i\} = C (i=0, 1, \dots, n)$ 。并且有 $\frac{2}{C} < \frac{2}{\lambda_0}$ 。

于是我们用如下的权因子 ω 进行实际应用:

$$\omega = \frac{2}{C}.$$

记 $D_k = \sum_{i=0}^n \|P_i^k - P_i^\infty\|, (k=0, 1, 2, \dots, \bar{k}-1)$, 其中 \bar{k} 为最终迭代次数。图 1 中绘制了三个例子(见 5.3 节)

在 $\omega = \frac{2}{C}$ 和理论最优权重 $\omega_{ib} = \frac{2}{\lambda_0 + \lambda_n}$ 下的 D_k 。图 1 表明, 采用 $\omega = \frac{2}{C}$ 和 $\omega_{ib} = \frac{2}{\lambda_0 + \lambda_n}$ 的两种迭代方法的收敛速度是相近的。

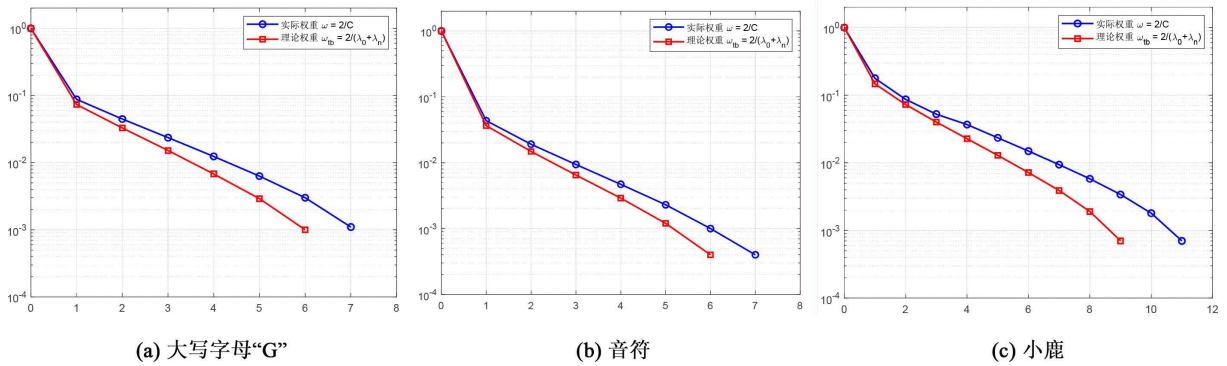


Figure 1. D_k for practical weight and theoretical weight (scaled such that $D_0 = 1$)
 图 1. 实际权重和理论权重下的 D_k (缩放使得 $D_0 = 1$)

5.2. 方法准备

定义第 k 次迭代的拟合误差为 $E_k = \sum_{j=0}^m \left\| Q_j - \sum_{i=0}^n B_i(t_j) P_i^k \right\|^2$, 设 E_0 为初始混合曲线的拟合误差, $\varepsilon = 1 \times 10^{-7}$ 为拟合精度。当 $|E_k - E_{k+1}| \leq \varepsilon$ 时迭代停止, 并记最后一个 E_k 为 E_∞ 。数值实验中, 给定一个数据点列 $\{Q_i | i = 0, 1, \dots, m\}$, 采用累加弦长法[19]为每个数据点分配一个参数 $t_i, i = 0, 1, \dots, m$ 。即

$$\begin{cases} t_0 = 0, t_m = 1 \\ t_i = t_{i-1} + \frac{\|q_i - q_{i-1}\|}{L}, i = 1, 2, \dots, m-1 \end{cases}$$

其中 $L = \sum_{i=0}^m \|q_i - q_{i-1}\|$ 为总弦长。

定义节点向量 $\{0, 0, 0, 0, u_4, u_5, \dots, u_n, 1, 1, 1, 1\}$, 其中

$$u_{j+3} = (1 - \alpha)t_{i-1} + \alpha t_i, j = 1, 2, \dots, n-3, i = [jd], \alpha = jd - i, d = \frac{m+1}{n-2}$$

虽然加速 LSPIA 方法可以从任意初始控制点开始, 但适当选择初始控制点可以使 LSPIA 快速收敛。选择初始控制点 $P^0 = \{P_i^0, i = 0, 1, \dots, n\}$ 为

$$\begin{cases} P_0^0 = Q_0 \\ P_n^0 = Q_m \\ P_i^0 = Q_{f(i)}, i = 1, 2, \dots, n-1 \end{cases}$$

其中, $f(i) = \left\lfloor \frac{(m+1)i}{n} \right\rfloor$ 。

本文实例中, 蓝色点表示给定的数据点, 绿色圆圈表示选取的初始控制点, 红色曲线表示拟合曲线。

5.3. 方法实现

例 1 使用具有 97 个控制点的三次 B 样条曲线来拟合从大写字母 “G” 上采样得到的 269 个点。

例 2 使用具有 30 个控制点的三次 B 样条曲线来拟合从音符上采样得到的 305 个点。

例 3 使用具有 176 个控制点的三次 B 样条曲线来拟合从鹿形上采样得到的 550 个点。

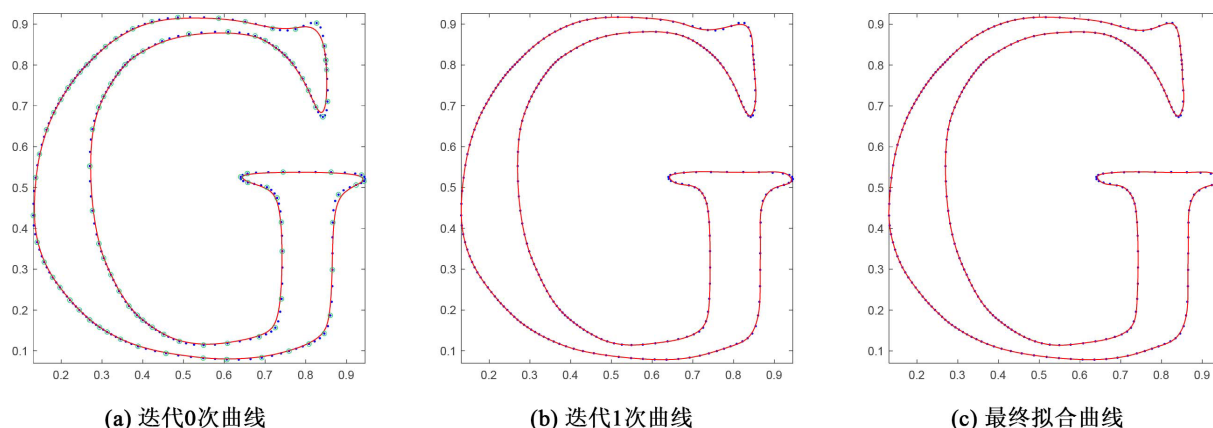


Figure 2. Curve fitting of G-shaped data points is carried out by the accelerated LSPIA method

图 2. 利用加速 LSPIA 方法对 G 形数据点进行曲线拟合

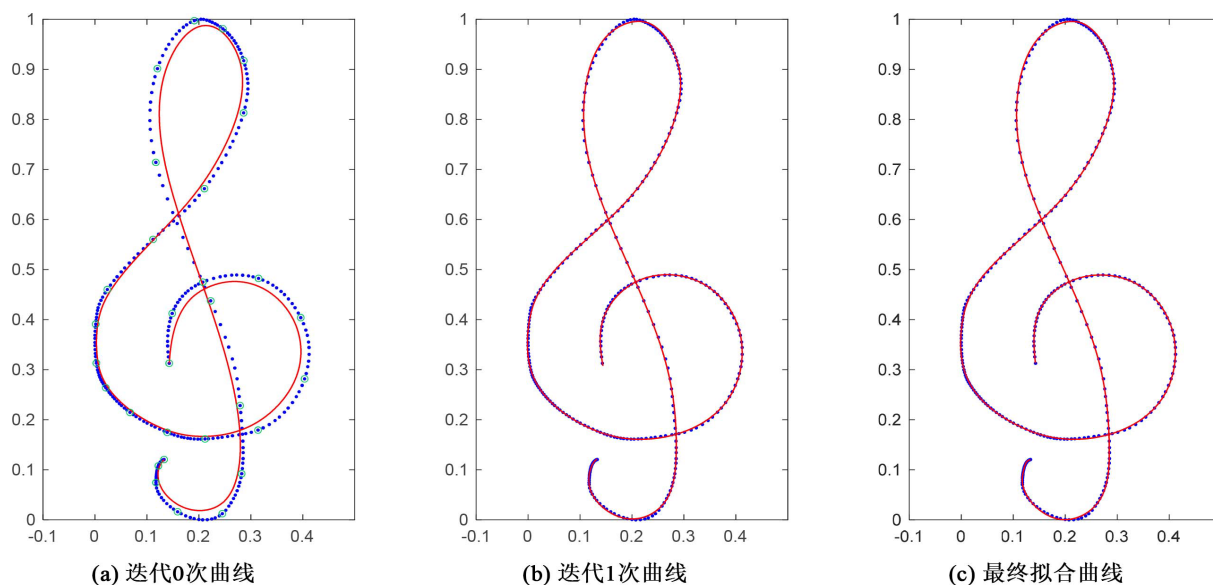


Figure 3. Curve fitting of note data points is carried out by the accelerated LSPIA method

图 3. 利用加速 LSPIA 方法对音符数据点进行曲线拟合

图 2(a)~图 4(a)描绘了从大写字母“G”、音符和鹿形上采样得到的数据点、选取的初始控制点以及它们的初始拟合曲线。图 2(b)~图 4(b)和图 2(c)~图 4(c)分别是用加速 LSPIA 方法进行迭代 1 次和最终拟合的曲线。可以看到它们能有效地拟合原始数据点。

表 1 列出了上述三个模型利用经典 LSPIA 算法和本文方法在迭代 $k = 0, 1, \dots, 4$ 步后的曲线拟合误差 E_k 。表 2 展示了原始数据点和控制点的个数、不同算法迭代终止时的拟合误差 E_∞ 、迭代次数和运行时间。结合表 1 和表 2 可以看出, 在达到相同拟合误差的情况下, 对于每一组数据点, 本文方法的拟合误差的衰减速度相比经典 LSPIA 算法都快得多, 且在达到迭代终止条件时的迭代次数和运行时间都小于经典 LSPIA 算法。其根本原因是随着迭代次数的增加, 本文方法对应的迭代矩阵的谱半径呈指数级衰减且

趋近于 0, 因此收敛速度显著快于经典 LSPIA 算法。

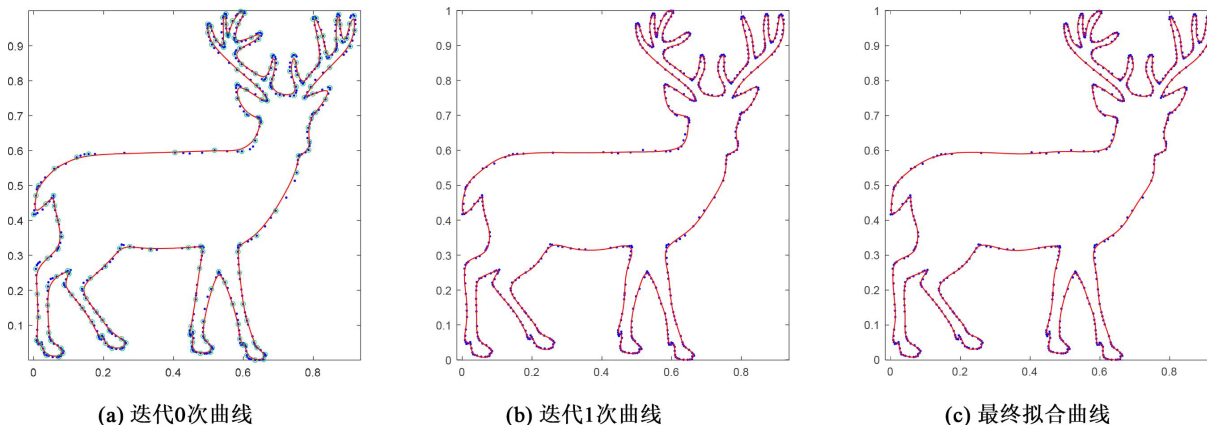


Figure 4. Curve fitting of deer-shaped data points is carried out by the accelerated LSPIA method
图 4. 利用加速 LSPIA 方法对鹿形数据点进行曲线拟合

Table 1. Iteration errors obtained by different algorithms for Examples 1 to 3

表 1. 不同算法拟合例 1~例 3 得到的迭代误差

| 模型 | 算法 | 迭代误差 | | | | |
|----------|-------|----------|----------|----------|----------|----------|
| | | E_0 | E_1 | E_2 | E_3 | E_4 |
| 大写字母 “G” | LSPIA | 0.256321 | 0.028713 | 0.004906 | 0.001506 | 0.000825 |
| | 本文方法 | 0.256321 | 0.000358 | 0.000246 | 0.000220 | 0.000214 |
| 音符 | LSPIA | 1.023522 | 0.098999 | 0.020787 | 0.006286 | 0.002720 |
| | 本文方法 | 1.023522 | 0.000878 | 0.000718 | 0.000694 | 0.000688 |
| 小鹿 | LSPIA | 0.361085 | 0.047267 | 0.015804 | 0.008796 | 0.006418 |
| | 本文方法 | 0.361085 | 0.003564 | 0.002727 | 0.002562 | 0.002512 |

Table 2. Experimental data obtained by different algorithms for Examples 1 to 3

表 2. 不同算法拟合例 1~例 3 得到的实验数据

| 模型 | 控制点/数据点 | 算法 | E_∞ | 迭代次数 | 运行时间/s |
|----------|---------|-------|------------|------|----------|
| 大写字母 “G” | 97/269 | LSPIA | 0.000211 | 47 | 0.010088 |
| | | 本文方法 | 0.000211 | 8 | 0.004354 |
| 音符 | 30/305 | LSPIA | 0.000687 | 40 | 0.009084 |
| | | 本文方法 | 0.000686 | 8 | 0.004409 |
| 小鹿 | 176/550 | LSPIA | 0.002482 | 73 | 0.016445 |
| | | 本文方法 | 0.002480 | 12 | 0.005408 |

6. 结语

基于 LSPIA 算法和矩阵求逆的具有 8 阶收敛性的迭代算法, 本文提出了一种曲线拟合的加速 LSPIA 算法, 证明了其收敛性, 并且显著提升了收敛速度。在未来的工作中, 我们将考虑利用本文方法解决曲

面拟合的问题。

基金项目

辽宁师范大学科学研究项目(20250097)。

参考文献

- [1] 齐东旭, 田自贤, 张玉心, 等. 曲线拟合的数值磨光方法[J]. 数学学报, 1975, 18(3): 173-184.
- [2] de Boor, C. (1979) How Does Agee's Smoothing Method Work? Army Research Office.
- [3] Lin, H.W., Wang, G.J. and Dong, C.S. (2004) Constructing Iterative Nonuniform B-Spline Curve and Surface to Fit Data Points. *Science in China Series: Information Sciences*, **47**, 315-331. <https://doi.org/10.1360/02yf0529>
- [4] Lin, H.W., Bao, H.J. and Wang, G.J. (2005) Totally Positive Bases and Progressive Iteration Approximation. *Computers & Mathematics with Applications*, **50**, 575-586. <https://doi.org/10.1016/j.camwa.2005.01.023>
- [5] Deng, C.Y. and Lin, H.W. (2014) Progressive and Iterative Approximation for Least Squares B-Spline Curve and Surface Fitting. *Computer-Aided Design*, **47**, 32-44. <https://doi.org/10.1016/j.cad.2013.08.012>
- [6] Wang, H.D. (2022) On Extended Progressive and Iterative Approximation for Least Squares Fitting. *The Visual Computer*, **38**, 591-602. <https://doi.org/10.1007/s00371-020-02036-8>
- [7] 王曾珍, 刘华勇. 带互异权值的 B 样条曲线的最小二乘渐进迭代逼近[J]. 小型微型计算机系统, 2023, 44(4): 845-849.
- [8] 周雅情, 张莉, 王积荣, 等. 关键点选取的最小二乘渐进迭代逼近[J]. 中国图象图形学报, 2020, 25(1): 148-157.
- [9] Hamza, Y.F., 蒋旖旎, 蔺宏伟. Gauss-Seidel 最小二乘渐进迭代逼近[J]. 计算机辅助设计与图形学学报, 2021, 33(1): 1-10.
- [10] 蒋旖旎, 蔺宏伟. 混合曲线曲面的 CG-LSPIA 拟合算法[J]. 中国科学: 信息科学, 2022, 52(7): 1251-1271.
- [11] Wu, N.C. and Liu, C.Z. (2024) Asynchronous Progressive Iterative Approximation Method for Least Squares Fitting. *Computer Aided Geometric Design*, **111**, Article 102295. <https://doi.org/10.1016/j.cagd.2024.102295>
- [12] Liu, C.Z., Wu, N.C., Li, J., et al. (2024) Two Novel Iterative Approaches for Improved LSPIA Convergence. *Computer Aided Geometric Design*, **111**, Article 102312. <https://doi.org/10.1016/j.cagd.2024.102312>
- [13] Jiang, Y.N. and Lin, H.W. (2023) IG-LSPIA: Least Squares Progressive Iterative Approximation for Isogeometric Collocation Method. *Mathematics*, **11**, Article 898. <https://doi.org/10.3390/math11040898>
- [14] Hu, Q.Q., Wang, Z.F., Yao, Z.M., et al. (2023) A Family of Hybrid Iterative Approximation Methods for Fitting Blending Curves. *The Visual Computer*, **40**, 4287-4301. <https://doi.org/10.1007/s00371-023-03082-8>
- [15] Soleymani, F. (2014) A Fast Convergent Iterative Solver for Approximate Inverse of Matrices. *Numerical Linear Algebra with Applications*, **21**, 439-452. <https://doi.org/10.1002/nla.1890>
- [16] Schulz, G. (1933) Iterative Berechnung der reziproken Matrix. *Zeitschrift für Angewandte Mathematik und Mechanik*, **13**, 57-59. <https://doi.org/10.1002/zamm.19330130111>
- [17] Li, W.G. and Li, Z. (2010) A Family of Iterative Methods for Computing the Approximate Inverse of a Square Matrix and Inner Inverse of a Non-Square Matrix. *Applied Mathematics and Computation*, **215**, 3433-3442. <https://doi.org/10.1016/j.amc.2009.10.038>
- [18] Li, H.B., Huang, T.Z., Zhang, Y., Liu, X. and Gu, T. (2011) Chebyshev-Type Methods and Preconditioning Techniques. *Applied Mathematics and Computation*, **218**, 260-270. <https://doi.org/10.1016/j.amc.2011.05.036>
- [19] Piegl, L. and Tiller, W. (1997) *The NURBS Book*. 2nd Edition, Springer-Verlag.