

# 基于 Newton 迭代法的最小二乘 渐进迭代逼近<sup>\*1)</sup>

兰 林 朱春钢<sup>2)</sup>

(大连理工大学数学科学学院, 大连 116024)

## 摘 要

最小二乘渐进迭代逼近 (LSPIA) 是一种有效的大规模数据拟合方法. 针对 LSPIA 的加速问题, 基于 Newton 迭代法, 本文提出曲线曲面的两类最小二乘渐进迭代逼近格式. 首先构造一个以控制顶点为变量的多元函数, 其 Hessian 矩阵为正定矩阵, 多元函数存在极小值, 且其极小值所对应的控制顶点与 LSPIA 的收敛结果一致. 对多元函数极小值问题, 采用 Newton 迭代法进行求解. 然后对 Newton 迭代格式中的 Hessian 矩阵和调整向量分别采用奇异值分解法和共轭梯度法求解, 从而给出两种 LSPIA 迭代格式, 分别记为 NLSPIA 和 INLSPIA. 最后给出两种迭代格式收敛性的理论证明. 数值实例验证了文中方法的有效性和可行性, 也表明了相同条件下, NLSPIA 与 INLSPIA 的收敛速度和计算时间都优于经典 LSPIA.

**关键词:** 渐进迭代逼近; 最小二乘; Newton 迭代法; 奇异值分解; 共轭梯度法.

**MR (2010) 主题分类:** 65D17.

## LEAST SQUARE PROGRESSIVE ITERATIVE APPROXIMATION BASED ON NEWTON ITERATIVE METHOD

Lan Lin Zhu Chungang

(School of Mathematical Sciences, Dalian University of Technology, Dalian 116024, China)

## Abstract

The least square progressive iterative approximation (LSPIA) is an efficient method for large-scale data fitting. To accelerate the convergence rate of LSPIA, we propose two kinds of LSPIA for curves and surfaces based on Newton iterative method. Firstly, we construct a multivariate function with control points as variables, whose Hessian matrix is a positive definite matrix. The control points corresponding to the minimal value of the multivariate function are consistent with the convergence results of LSPIA. Meanwhile, the Newton iterative method is applied to solve the minimal value of the multivariate function. Then we obtain two LSPIA iterative schemes by using singular value decomposition and the conjugate gradient method respectively to deal with the Hessian matrix and adjustment vectors, which are named NLSPIA and INLSPIA respectively. Finally, we also give the theoretical proof of the convergence of the proposed schemes. Numerical examples verify the

<sup>\*</sup> 2021 年 8 月 24 日收到.

<sup>1)</sup> 基金项目: 国家自然科学基金 (12071057, 11671068) 资助.

<sup>2)</sup> 通讯作者: 朱春钢, Email: cgzhu@dlut.edu.cn

effectiveness and feasibility of the proposed methods, and also show that the convergence rate and the calculation time of NLSPIA and INLSPIA are both better than those of the classical LSPIA under the same conditions.

**Keywords:** progressive iterative approximation; least square; Newton iterative method; singular value decomposition; conjugate gradient method.

**2010 Mathematics Subject Classification:** 65D17.

## 1. 引言

在计算机辅助几何设计 (Computer Aided Geometric Design, 简称 CAGD) 中, 采用参数曲线和曲面对一组数据点的拟合和逼近是一个基础而又十分重要的研究问题. 渐进迭代逼近 (Progressive Iterative Approximation, 简称 PIA) 是近年来出现的一种十分有效的数据拟合方法, 它从初始曲线或曲面开始, 通过不断迭代调整控制顶点, 生成一组混合曲线或曲面序列, 其极限插值或逼近给定的数据点集. 因为其明显的几何意义、稳定的收敛性、算法简单以及易于编程等优势, 它一出现就受到学者们的关注和研究<sup>[6-10]</sup>.

PIA 迭代法的产生可以追溯到 1975 年齐东旭等<sup>[1]</sup>所提出的均匀 3 次 B 样条的盈亏修正算法. 1979 年, de Boor<sup>[2]</sup>也提出相似的算法并给出了算法收敛性的相关证明. 2004 年, Lin 等<sup>[3]</sup>将盈亏修正算法推广到了非均匀 3 次 B 样条曲线和曲面上, 并证明了其收敛性. 2005 年, Lin 等<sup>[4]</sup>更进一步地证明了任意标准全正 (Normalized Totally Positive, 简称 NTP) 基生成的混合曲线和曲面都具有盈亏修正性质, 称其为渐进迭代逼近性质, 命名为 PIA. 2006 年, 史利民和王仁宏<sup>[5]</sup>证明了 NURBS 曲线和曲面同样也具有渐进迭代逼近性质. 2007 年, Delgado 等<sup>[6]</sup>对比了不同基函数下 PIA 方法的收敛速度, 并证明了 B 样条基函数具有最快的收敛速度. 2012 年, 陈杰等<sup>[7]</sup>将 PIA 推广至了非标准全正基, 拓宽了 PIA 的适用范围.

近年来, 为了提升 PIA 的收敛速度, 不断有学者对 PIA 算法进行改进. Lu<sup>[8]</sup>对迭代过程中的调整向量赋予权值, 提出带权的 PIA(WPIA), 并给出了理论最优权值的取法. 基于此, 陈杰<sup>[7]</sup>将 WPIA 进行了推广, 对不同控制顶点的调整向量赋予不同的权值, 使其具有不同的收敛速度. 对于带权 PIA 的研究, 张莉等<sup>[9]</sup>提出一种带互异权值的 PIA, 增加了实际操作中的灵活性. 此外, 刘晓艳和邓重阳<sup>[10]</sup>提出 Jacobi-PIA 法, 将 Jacobi 迭代法的思想巧妙地融入了 PIA 当中.

对于经典的 PIA, 要求控制顶点数与待拟合的数据点数保持相等, 这一要求限制了 PIA 在大规模数据拟合中的应用. 为了解决这个问题, Lin 和 Zhang<sup>[11]</sup>提出了一种 PIA 的推广形式 (Extended Progressive Iterative Approximation, 简称 EPIA), 采用该法可以进行并行计算且允许控制顶点数小于数据点数. 类似于 EPIA, Deng 和 Lin<sup>[12]</sup>提出了一种最小二乘渐进迭代逼近 (Least Square Progressive Iterative Approximation, 简称 LSPIA), 它能够高效稳定地拟合大规模数据点, 具有保形性, 其迭代生成的曲线和曲面序列的极限与给定数据点列的最小二乘逼近结果一致. 更多关于 PIA 和 LSPIA 方法的内容详见文献 [13,14].

相比于 PIA 中的各种收敛加速方法, 对 LSPIA 的收敛速度加速的研究相对较少. 本文从一个以控制顶点为变量的多元函数出发, 采用具有平方收敛阶的 Newton 迭代法求解多元函数的极小值, 其取得极小值时的控制顶点生成的混合曲线和曲面为待拟合数据点列的最小二乘逼近结果, 即与 LSPIA 的收敛结果一致. 基于 Newton 迭代格式, 运用直接分解法和间接逼近法处理多元函数的 Hessian 矩阵, 提出两种 LSPIA 迭代格式, 分别记为 NLSPIA 和 INLSPIA,

并证明了这两种格式的收敛性. 直接分解法对 Hessian 矩阵进行奇异值分解, 规避了迭代格式中逆矩阵求解的问题. 间接逼近法则是运用共轭梯度法求取每一步迭代中调整向量的近似解, 类似于文献 [17] 中提出的 PPIA 算法. 与 PPIA 相比, 本文需要处理的矩阵为 Hessian 矩阵, 本文对 Hessian 矩阵的预处理矩阵的可逆性进行了理论证明, 规避了文献 [17] 提出的 PPIA 中预处理矩阵的可逆性未通过理论证明, 而是由大量数值实验得到的缺陷. 此外, 又因本文方法得到的 Hessian 矩阵为正定矩阵, 而需要求解的方程组的系数矩阵就为 Hessian 矩阵, 可直接采用共轭梯度法进行求解, 避免了 PPIA 中需要先构造正定矩阵方程组的问题. 最后, 数值实验结果表明, 在相同条件下, 本文提出的方法在收敛速度与计算时间上都优于经典 LSPIA 算法.

## 2. 预备知识

本节介绍 PIA 方法和 LSPIA 方法的相关基础知识.

**定义 2.1** [20]. 一个  $n \times m$  阶矩阵  $X$  被称为完全正矩阵当且仅当  $X$  的所有子式都是非负的.

**定义 2.2** [4]. 定义在参数域  $\Xi \subseteq \mathbb{R}$  上的基函数  $\{B_0(t), B_1(t), \dots, B_n(t)\}$  被称为标准全正 (NTP) 基当且仅当基函数满足下列性质:

- (1) 非负性,  $\forall t \in \Xi$ , 有  $B_i(t) \geq 0, i = 0, 1, \dots, n$ ;
- (2) 单位分解性,  $\sum_{i=0}^n B_i(t) = 1$ ;
- (3) 基函数在参数域中任意递增序列下的配置矩阵为完全正矩阵.

给定待拟合的数据点列  $\{Q_j\}_{j=1}^m$ , 并赋予每一个点  $Q_j$  参数值  $t_j$ , 满足

$$0 = t_0 < t_1 < \dots < t_m = 1.$$

选取  $\{B_0(t), B_1(t), \dots, B_n(t)\}$  作为定义在区间  $[0, 1]$  上的 NTP 基, 则基函数  $\{B_i(t)\}_{i=0}^n$  在给定的递增参数序列  $0 = t_0 < t_1 < \dots < t_m = 1$  上的配置矩阵为

$$C_{m \times n} = \begin{bmatrix} B_0(t_0) & B_1(t_0) & \dots & B_n(t_0) \\ B_0(t_1) & B_1(t_1) & \dots & B_n(t_1) \\ \vdots & \vdots & \ddots & \vdots \\ B_0(t_m) & B_1(t_m) & \dots & B_n(t_m) \end{bmatrix}. \quad (2.1)$$

从点列  $\{Q_j\}_{j=0}^m$  中选取初始控制点列  $\{P_i^0\}_{i=0}^n$ , 其中  $m \geq n$ , 便可得到初始混合曲线为

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

对 PIA 方法, 要求数据点数和控制点数相等, 也即  $m = n$ , 这导致 PIA 方法不宜于拟合大规模的数据点列; 对 LSPIA 方法, 数据点数大于控制点数, 即  $m > n$ , LSPIA 能够高效地应用于大规模数据拟合中并且能显著地减少计算量. 无论是 PIA 还是 LSPIA, 都是通过如下的迭代格式不断调整控制顶点, 从而生成混合曲线序列去插值或逼近目标曲线

$$P_i^{k+1} = P_i^k + \Delta_i^k, i = 0, 1, 2, \dots, n; k \geq 0,$$

相应的每一步迭代生成的混合曲线为

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

其中  $k$  为迭代次数,  $\Delta_i^k$  为调整向量, PIA 和 LSPIA 的区别就在于调整向量的求取,  $\Delta_i^k$  可具体表示为:

$$\Delta_i^k = \begin{cases} Q_i - P^k(t_i), & \text{PIA,} \\ \mu \sum_{j=0}^m B_i(t_j)(Q_j - P^k(t_j)), & \text{LSPIA,} \end{cases} \quad i = 0, 1, \dots, n.$$

因为 PIA 要求数据点数和控制点数相同, 所以每一步的调整向量直接求取控制点列的差向量. LSPIA 中, 数据点数大于控制点数, 于是先求取数据点列的差向量

$$\delta_j^k = Q_j - P^k(t_j), \quad j = 0, 1, 2, \dots, m,$$

再对每一个控制顶点所对应的数据点进行加权平均求得调整向量. 计算表达式中的  $\mu$  是一个常数, 且满足  $0 < \mu < \frac{2}{\lambda_0}$ ,  $\lambda_0$  为矩阵  $C^T C$  的最大特征值, 矩阵  $C$  为配置矩阵. 此外, 记  $\{P^k(t)\}_{k=0}^\infty$  为迭代生成的曲线序列, PIA 中,  $\{P^k(t)\}_{k=0}^\infty$  的极限曲线插值于给定的数据点列<sup>[4]</sup>; 而 LSPIA 中,  $\{P^k(t)\}_{k=0}^\infty$  的极限曲线为给定数据点列的最小二乘逼近结果<sup>[12]</sup>.

关于曲面的 PIA 与 LSPIA 的知识与曲线类似, 其具体内容可详见文献 [4] 和 [12], 我们在此不再赘述.

### 3. 两类新型的曲线 LSPIA 迭代格式

在本节中, 基于 Newton 迭代法, 构造两类新型的曲线 LSPIA 迭代格式并探究其收敛性. 对于本节涉及的所有定理, 我们均在附录 A 中给出详细的理论证明.

#### 3.1. 迭代格式的构造

无论是 PIA 还是 LSPIA, 其核心思想都是最小化混合曲线  $P(t) = \sum_{i=0}^n B_i(t) P_i$  和数据点列  $\{Q_j\}_{j=0}^m$  之间的距离, 即通过迭代使得  $\sum_{j=0}^m |Q_j - \sum_{i=0}^n B_i(t_j) P_i|$  收敛至其最小值. 本文便从数据点列和混合曲线之间的距离出发, 构造以控制点列为变量的  $n+1$  元函数

$$f(P_0, P_1, \dots, P_n) = \sum_{i=0}^m (Q_i - P(t_i))^2 = \sum_{i=0}^m \left( Q_i - \sum_{j=0}^n B_j(t_i) P_j \right)^2. \quad (3.1)$$

对于标准全正基函数  $\{B_i(t)\}_{i=0}^n$ , 有如下的性质.

**定理 3.1.** 标准全正基函数  $\{B_i(t)\}_{i=0}^n$  在参数值  $\{t_j\}_{j=0}^m$  上的配置矩阵  $C$  是列满秩矩阵, 其中  $n \leq m$ .

**定理 3.2.** 多元函数  $f(P_0, P_1, \dots, P_n)$  的 Hessian 矩阵满足  $H(f) = 2C^T C$ , 且为正定矩阵, 其中  $C$  为基函数  $\{B_i(t)\}_{i=0}^n$  在参数值  $\{t_j\}_{j=0}^m$  上的配置矩阵.

**定理 3.3.** 多元函数  $f(P_0, P_1, \dots, P_n)$  在临界点处存在极小值.

定理 3.1、定理 3.2 和定理 3.3 的详细证明我们分别在附录 A 的 A.1、A.2 和 A.3 中给出.

记函数  $f$  的一阶导为  $h(f)$ , 由定理 3.3 可知, 求  $f$  的极小值, 等价于求得  $P^\lambda$  使得  $h(f(P^\lambda)) = \mathbf{0}$  成立; 以临界点  $P^\lambda$  为控制顶点所生成的曲线则插值或逼近待拟合的数据点

列. 本文采用具有平方收敛阶的 Newton 迭代法求解  $h(f) = 0$ , 而  $h(f)$  关于各变量的偏导数为  $f$  的 Hessian 矩阵  $H(f)$ , 所以基于求解此方程的 Newton 迭代法的 LSPIA 具体形式如下

$$P^{k+1} = P^k - H(f)^{-1}h(f) = P^k + (C^T C)^{-1} [C^T (Q - CP^k)]; \quad (3.2)$$

其中  $Q = [Q_0, Q_1, \dots, Q_m]^T$  为给定的待拟合数据点列,  $P^k = [P_0^k, P_1^k, \dots, P_n^k]^T$  为第  $k$  次迭代后的控制点列,  $P^{k+1} = [P_0^{k+1}, P_1^{k+1}, \dots, P_n^{k+1}]^T$  为第  $k+1$  次迭代后的控制点列. 因为  $f$  的 Hessian 矩阵  $H(f) = 2C^T C$  为常量矩阵, 在 Newton 迭代法的每一步迭代中不需要重新计算.

对于迭代格式 (3.2) 的收敛性, 从 Newton 迭代法的角度出发, 因为多元函数  $f$  的 Hessian 矩阵  $H(f)$  是正定矩阵, 故迭代格式收敛. 从迭代矩阵的角度出发, 令  $B = C^T C$ , 则迭代格式 (3.2) 等价于

$$P^{k+1} - B^{-1}C^T Q = (I - B^{-1}C^T C)[P^k - B^{-1}C^T Q],$$

迭代矩阵为  $I - B^{-1}C^T C = O$ , 所以其谱半径为 0, 故迭代格式收敛.

对于迭代格式 (3.2), 具体迭代过程如下. 首先从数据点列  $\{Q_j\}_{j=0}^m$  中选取初始控制顶点  $\{P_i^0\}_{i=0}^n$ , 其中  $m \geq n$ , 然后在区间  $t \in [0, 1]$  上构造初始混合曲线  $P^0(t) = \sum_{i=0}^n B_i(t)P_i^0$ . 在第一次迭代中, 先求得数据点列的差向量

$$\delta^0 = [Q_0 - P^0(t_0), Q_1 - P^0(t_1), \dots, Q_m - P^0(t_m)]^T,$$

然后求得控制顶点的差向量为

$$\Delta^0 = B^{-1}C^T \delta^0,$$

于是可得到新的控制点列为

$$P^1 = [P_0^1, P_1^1, \dots, P_n^1]^T = P^0 + \Delta^0;$$

新的混合曲线则为  $P^1(t) = \sum_{i=0}^n B_i(t)P_i^1$ . 类似的, 假设已经得到  $k$  次迭代后生成的混合曲线  $P^k(t)$ , 为了得到第  $k+1$  次混合曲线, 先计算数据点列的差向量

$$\delta^k = [Q_0 - P^k(t_0), Q_1 - P^k(t_1), \dots, Q_m - P^k(t_m)]^T,$$

然后求得控制顶点差向量  $\Delta^k = B^{-1}C^T \delta^k$  和第  $k+1$  次迭代后的控制顶点  $P^{k+1} = P^k + \Delta^k$ , 最后得到第  $k+1$  迭代生成的混合曲线

$$P^{k+1}(t) = \sum_{i=0}^n B_i(t)P_i^{k+1}.$$

当  $m > n$  时, 即数据点数大于控制点数时, 由式 (3.2) 可得其迭代极限结果为  $P^\infty = (C^T C)^{-1}C^T Q$ , 等价于

$$C^T C P^\infty = C^T Q,$$

表示极限曲线为数据点列  $\{Q_j\}_{j=0}^m$  的最小二乘逼近结果. 当  $m = n$  时,  $C^T C$  和  $C$  都是  $n+1$  阶方阵, 此时, 迭代格式 (3.2) 可化简为

$$P^{k+1} = P^k + C^{-1}(Q - CP^k),$$

与文献 [16] 和 [17] 中所提出的预处理 PIA(PPIA) 的迭代格式相似. 此时迭代极限结果为  $P^\infty = C^{-1}Q$ , 等价于

$$P^\infty(t) = CP^\infty = Q,$$

表示极限曲线插值于数据点列.

在下一小节中, 我们将通过对迭代格式 (3.2) 中的 Hessian 矩阵的近似计算来构造出两类新型的曲线 LSPIA 迭代格式.

### 3.2. 基于 Hessian 矩阵近似计算的两类新型 LSPIA 迭代格式

虽然从理论上讲, 迭代格式 (3.2) 的迭代矩阵为零矩阵, 具有非常快的收敛速度, 但需要面对求解矩阵  $C^TC$  的逆矩阵的问题. 为了避免直接求取逆矩阵, 本文采用直接分解法和间接逼近法来近似求解矩阵  $C^TC$ , 进而构造两类新型 LSPIA 迭代格式.

#### 3.2.1. 直接分解法

$C^TC$  是正定矩阵, 具有许多良好的性质, 从其本身出发, 考虑矩阵的奇异值分解. 对于矩阵  $B = C^TC$ , 设其奇异值分解如下

$$B = U\Sigma V^T = U \begin{bmatrix} \sigma_0 & & & \\ & \sigma_1 & & \\ & & \ddots & \\ & & & \sigma_n \end{bmatrix} V^T,$$

其中矩阵  $U$  和  $V$  为正交矩阵,  $\Sigma$  为对角矩阵. 令

$$W = \begin{bmatrix} \frac{1}{\sigma_0} & & & \\ & \frac{1}{\sigma_1} & & \\ & & \ddots & \\ & & & \frac{1}{\sigma_n} \end{bmatrix},$$

于是可以很轻松地得到  $B^{-1} = VWU^T$ . 此时 (3.2) 的迭代格式转化为

$$P^{k+1} = P^k + VWU^T C^T (Q - CP^k). \quad (3.3)$$

将其记为 NLSPIA (Newton Least Square Progressive Iterative Approximation). 关于 NLSPIA 的曲线拟合算法具体过程如算法 3.1 所示.

#### 算法 3.1. NLSPIA 的曲线拟合算法

输入: 数据点列  $Q = [Q_0, Q_1, \dots, Q_m]$  和拟合误差精度  $E > 0$ .

输出: 达到拟合误差精度的混合曲线  $P^k(t)$ .

步 1. 对数据点进行参数化;

步 2. 计算配置矩阵  $C$  和矩阵  $B = C^TC$ ;

步 3. 对矩阵  $B$  进行奇异值分解  $B = U\Sigma V^T$ ;

步 4. 选取初始控制点列  $\{P_i^0\}_{i=0}^n$ ;

步 5. 令  $k = 0$ ;

步 6. 计算数据点列的差向量  $\delta^k = [Q_0 - P^k(t_0), Q_1 - P^k(t_1), \dots, Q_m - P^k(t_m)]^T$ ;

步 7. 计算调整向量  $\Delta^k = VWU^T C^T \delta^k$ ;

步 8. 计算第  $k+1$  次迭代后的控制点列  $P^{k+1} = P^k + \Delta^k$  和第  $k+1$  次迭代后的混合曲线  $P^{k+1}(t) = \sum_{i=0}^n B_i(t)P_i^{k+1}$ ;

步 9. 计算拟合误差  $E_k$ , 若  $|E_k - E_{k-1}| < E$ , 则停止迭代, 否则令  $k = k+1$ , 转步 6.

对于 NLSPIA 算法的收敛性问题, 有如下的定理 3.4 成立, 且其详细的理论证明我们在附录 A 的 A.4 中给出.

**定理 3.4.** NLSPIA 迭代格式 (3.3) 是收敛的, 且收敛结果为待拟合数据点列的最小二乘逼近结果.

更进一步分析, 对 Hessian 矩阵进行奇异值分解后, NLSPIA 的时间复杂度主要分为奇异值分解和最小二乘渐进迭代逼近两个部分, 且矩阵  $C^T C$  的奇异值分解可以预先计算好, 在进行最小二乘渐进迭代逼近的时候可以直接调用.

### 3.2.2. 间接逼近法

不同于直接分解法中从矩阵本身出发, 间接逼近法将重点放在迭代格式 (3.2) 的整个调整向量上

$$(C^T C)^{-1} [C^T (Q - CP^k)], \quad (3.4)$$

式 (3.4) 等价于求解如下  $X$

$$C^T C X = C^T (Q - CP^k). \quad (3.5)$$

又当矩阵  $C^T C$  越逼近对角矩阵, 式 (3.5) 的求解就越简单. 所以先将  $B = C^T C$  进行拆分<sup>[17]</sup>  $B = B_q + R$ , 其中矩阵  $R$  为

$$R = \begin{bmatrix} 0 & \cdots & 0 & b_{0,q+1} & \cdots & b_{0n} \\ 0 & \ddots & \ddots & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & \ddots & b_{n-q-1,n} \\ b_{q+1,0} & \ddots & \ddots & \ddots & \ddots & 0 \\ \vdots & \ddots & \ddots & \ddots & \ddots & \vdots \\ b_{n,0} & \cdots & b_{n,n-q-1} & 0 & \cdots & 0 \end{bmatrix},$$

$$b_{i,j} = \sum_{k=0}^m B_i(t_k) B_j(t_k), i, j \in \Omega_q,$$

$$\Omega_q = \{i, j \mid |i - j| > q, 0 \leq i, j \leq n, i, j \in \mathbb{Z}\}.$$

所以  $B_q = B - R$  是一个  $2q+1$  ( $0 \leq q \leq n, q \in \mathbb{Z}$ ) 对角矩阵, 即  $B_q$  的元素为  $C^T C$  中  $2q+1$  对角线上的元素,  $R$  则为  $C^T C$  中  $2q+1$  对角线外的元素, 且  $B_q$  和  $R$  都为对称矩阵. 再设定一个对角矩阵为

$$A = \text{diag}(a_{00}, a_{11}, \dots, a_{nn}),$$

记  $R = (r_{ij})_{(n+1)}$ , 使矩阵  $A$  的对角元素满足:

$$a_{ii} = \sum_{j=0}^{n+1} r_{ij}, \quad i = 0, 1, \dots, n; \quad (3.6)$$

也即满足  $Ae = Re$ , 其中  $e = (1, 1, \dots, 1)^T$ . 最后构造  $2q + 1$  对角矩阵

$$N = B_q + A. \quad (3.7)$$

当  $q = 0$  时,  $B_q$  为  $B$  的对角线元素构成的对角矩阵, 根据式 (3.7), 矩阵  $N$  也为对角矩阵, 且  $N$  的第  $i$  个对角元素为矩阵  $B$  的第  $i$  行的行和; 当  $q = n$  时,  $R$  为零矩阵, 此时  $N = B$ . 所以随着  $q$  不断地增大, 矩阵  $N$  会越来越逼近  $B$ .

文献 [17] 提出的预处理 PIA(PPIA), 构造配置矩阵  $C_{n \times n}$  的预处理矩阵  $P_C = C_q + A_C$ , 但预处理矩阵  $P_C$  的可逆性未通过理论证明, 而是通过大量数值实验得出. 而对于本文所提出的方法, 需要处理的矩阵  $C^T C$  为正定矩阵, 有如下性质.

**定理 3.5.** 矩阵  $B = C^T C$  按式 (3.7) 得到的逼近矩阵  $N$  是非奇异的且也为正定矩阵.

定理 3.5 的详细理论证明我们在附录 A 的 A.5 中给出.

因为 PPIA [17] 中标准全正基函数在参数值对应的配置矩阵  $C_{n \times n}$  虽然为完全正矩阵, 但并不能保证其正定性与可逆性, 所以文献 [17] 中, 配置矩阵  $C_{n \times n}$  的预处理矩阵  $P_C$  的可逆性的理论证明仍是一个值得研究的问题.

用  $B$  的逼近矩阵来代替矩阵  $B$ , 则迭代格式 (3.2) 转化为

$$P^{k+1} = P^k + N^{-1}C^T(Q - CP^k). \quad (3.8)$$

间接逼近法的重心在整个调整向量上, 根据 (3.8), 在迭代中的第  $k$  步需要计算  $N^{-1}C^T(Q - CP^k)$ , 等价于求解如下  $X^k$ ,

$$NX^k = C^T(Q - CP^k). \quad (3.9)$$

随着矩阵  $C^T C$  的阶数不断增加, 计算复杂度也不断增加. 对于 (3.9), 其实并不要求得其精确解, 只需要快速地得到其近似解然后代入 (3.8) 中, 通过 (3.8) 的迭代得到新的控制顶点. 类似于文献 [17], 本文采用共轭梯度 (CG) 法对 (3.9) 进行迭代求解. 而因为逼近矩阵  $N$  为正定矩阵, 则避免了文献 [17] 中需要先构造正定矩阵方程组的问题. 将 CG 法求解 (3.9) 的迭代过程称作内迭代, 将 (3.9) 的近似解代入 (3.8) 进行最小二乘渐进迭代逼近的过程称作外迭代.

为了区分间接逼近法和直接分解法的迭代格式, 将采用共轭梯度法求解调整向量的近似解然后进行最小二乘渐进迭代逼近的迭代算法记为 INLSPIA (Inexactly NLSPIA).

记  $\tilde{X}^{(l,k)}$  为第  $k$  次外迭代中, 内迭代  $l$  次得到的  $X^k$  的近似解, 内迭代的迭代终止条件为

$$\frac{\|C^T(Q - C\tilde{P}^k) - N\tilde{X}^{(l,k)}\|}{\|C^T(Q - C\tilde{P}^k)\|} < \varepsilon, \quad (3.10)$$

其中  $\varepsilon$  为设定好的迭代误差,  $\tilde{P}^k$  为 INLSPIA 第  $k$  次迭后得到的控制点列. 通过内迭代得到式 (3.9) 的近似解, 则相应的第  $k + 1$  次外迭代 (3.8) 变为

$$\tilde{P}^{k+1} = \tilde{P}^k + \tilde{X}^{(l,k)}. \quad (3.11)$$

由迭代格式 (3.8) 可知, INLSPIA 的迭代矩阵为  $D = I - N^{-1}C^T C$ . 而随着  $q$  不断增大, 矩阵  $N$  越来越逼近矩阵  $C^T C$ , 则迭代矩阵谱半径  $\rho(D)$  不断减小, INLSPIA 的收敛速度不断加快. 不过每一步内迭代  $NX^k = C^T(Q - C\tilde{P}^k)$  求解复杂度会不断增加. 为了平衡收敛速度



和计算复杂度<sup>[17]</sup>, 一般我们选取  $q = \lfloor \frac{n+1}{2} \rfloor$ ,  $n+1$  为矩阵  $C^T C$  的阶数. INLSPIA 的曲线拟合算法具体过程如算法 3.2 所示.

### 算法 3.2. INLSPIA 的曲线拟合算法

输入: 数据点列  $Q = [Q_0, Q_1, \dots, Q_m]$ , 拟合误差精度  $E > 0$ , 内迭代 CG 法的迭代终止误差  $\varepsilon > 0$  和内迭代容许最大迭代次数  $L$ .

输出: 达到拟合误差精度的混合曲线  $\tilde{P}^k(t)$ .

步 1. 对数据点进行参数化;

步 2. 计算配置矩阵  $C$  和矩阵  $B = C^T C$ ;

步 3. 计算逼近矩阵  $N = B_q + A$ ;

步 4. 选取初始控制点列  $\{P_i^0\}_{i=0}^n$ ;

步 5. 运用 CG 法求解初始调整向量的近似解  $NX^0 = C^T(Q - CP^0)$ , 当内迭代误差  $\varepsilon_l < \varepsilon$  或  $l$  大于内迭代的容许最大迭代次数  $L$  时, 返回  $X^0$  的近似解  $\tilde{X}^{(l,0)}$ ;

步 6. 计算第一次迭代后的控制点列  $\tilde{P}^1 = P^0 + \tilde{X}^{(l,0)}$ ;

步 7. 令  $k = 1$ ;

步 8. 令  $l = 0$ , 运用 CG 法求第  $k$  次调整向量的近似解  $NX^k = C^T(Q - C\tilde{P}^k)$ , 当内迭代误差  $\varepsilon_l < \varepsilon$  或  $l$  大于内迭代容许最大迭代次数  $L$  时, 返回  $X^k$  的近似解  $\tilde{X}^{(l,k)}$ ;

步 9. 计算第  $k+1$  次迭代后的控制点列  $\tilde{P}^{k+1} = \tilde{P}^k + X^{(l,k)}$  和第  $k+1$  次迭代后的混合曲线  $\tilde{P}^{k+1}(t) = \sum_{i=0}^n B_i(t)\tilde{P}_i^{k+1}$ ;

步 10. 计算拟合误差  $E_k$ , 若  $|E_k - E_{k-1}| < E$ , 停止迭代, 否则令  $k = k+1$ , 转步 8.

对于 INLSPIA 算法的收敛性问题, 有如下的定理 3.6 成立, 且其详细的理论证明我们在附录 A 的 A.6 中给出.

**定理 3.6.** 设  $P^*$  是  $C^T C X = C^T Q$  的精确解, 则迭代序列  $\{\tilde{P}^k\}_{k=0}^\infty$  收敛于  $P^*$ , 即 INLSPIA 迭代生成的曲线序列的极限曲线为待拟合数据点列的最小二乘逼近结果.

### 3.2.3. 两类迭代格式的再讨论

如上关于  $B = C^T C$  近似的两种方法中, 直接分解法从矩阵本身进行考虑, 其本质为在矩阵信息不丢失或丢失一部分的情况下, 用易于求逆的矩阵  $M$  去近似  $C^T C$ , 其中  $M$  含有  $C^T C$  的部分信息或全部信息. 比如最容易求逆矩阵的对角矩阵, 现令  $D = \text{diag}(d_1, d_2, \dots, d_3)$  为对角矩阵. 再用对角放缩法处理矩阵  $D$ , 对角放缩法指得是对矩阵  $D$  的元素进行放缩得到  $M$ , 使得矩阵  $M$  和矩阵  $B$  的某一指定信息相同. 即  $M = D\beta$ , 其中  $\beta$  可为一个数, 表示  $D$  的每一个对角元的放缩倍数是相同的;  $\beta$  也可为一个向量, 表示  $D$  的每一个对角元的放缩倍数是不同的. 经典 LSPIA 就是对角放缩法的一种特殊情况, 即保证  $D$  和  $B$  的每一行所形成的向量的 1-范数相等. 由  $B = C^T C = (b_{ij})_{n+1}$ , 则可记

$$b_i = [b_{i0}, b_{i1}, \dots, b_{in}], \quad i = 0, 1, \dots, n,$$

$$B = [b_0, b_1, \dots, b_n]^T.$$

则放缩倍数为

$$\beta = [\beta_0, \beta_1, \dots, \beta_n]^T;$$

$$\beta_i = \frac{\|b_i\|_1}{d_i}, \quad i = 0, 1, \dots, n.$$

此时得到的矩阵  $M$  的第  $i$  个对角元素为矩阵  $C^T C$  第  $i$  行的行和; 用  $M^{-1}$  作为  $(C^T C)^{-1}$  的逼近矩阵来替换  $(C^T C)^{-1}$ , 由此得到的迭代格式为经典 LSPIA. 除对角放缩法外, 还有如文献 [16] 中的  $QR$  分解法, 矩阵  $C^T C$  有  $QR$  分解

$$C^T C = PT = P \begin{bmatrix} \alpha_0 & & * \\ & \ddots & \\ & & \alpha_n \end{bmatrix},$$

其中  $P$  为正交矩阵,  $T$  为上三角矩阵. 然后构造矩阵  $\bar{T}$  为

$$\bar{T} = \begin{bmatrix} \frac{1}{\alpha_0} & & * \\ & \ddots & \\ & & \frac{1}{\alpha_n} \end{bmatrix},$$

即将上三角矩阵  $T$  的对角元取倒数, 其余元素不变, 用矩阵  $\bar{T}P^T$  去替换  $(C^T C)^{-1}$ .

而间接逼近法从调整向量  $N^{-1}C^T(Q - CP^k)$  进行考虑, 其本质为每一步迭代中求解如下  $X^k$

$$NX^k = C^T(Q - CP^k).$$

因为矩阵  $N$  是正定矩阵, 除 CG 法外, 还有很多迭代方法可以用于求解正定线性方程组. 需要注意的则是在提高收敛速度的同时尽可能地降低计算复杂度, 所以通过选取合适的  $q$  值来进行平衡.

#### 4. 两类新型的曲面 LSPIA 迭代格式

本节将曲线的两类新型 LSPIA 迭代格式推广至张量积型混合曲面中. 由于与曲线证明类似, 本节中有关理论以文字和公式简单描述形式给出, 省略了严格的定理数学证明与算法描述.

设  $\{Q_{ij}\}_{i=0,j=0}^{m_1,m_2}$  为给定待拟合数据点列, 并赋予每一个点  $Q_{ij}$  一个参数值  $(u_i, v_j)$ , 满足

$$0 = u_0 < u_1 < \cdots < u_{m_1} = 1, \quad 0 = v_0 < v_1 < \cdots < v_{m_2} = 1.$$

又设点列  $\{P_{hl}\}_{h=0,l=0}^{n_1,n_2}$  为控制点列, 构造以控制顶点为变量的  $(n_1 + 1) \cdot (n_2 + 1)$  元函数

$$\begin{aligned} F(P_{00}, \cdots, P_{0n}, P_{10}, \cdots, P_{n_1 n_2}) &= \sum_{i=0}^{m_1} \sum_{j=0}^{m_2} (Q_{ij} - P(u_i, v_j))^2 \\ &= \sum_{i=0}^{m_1} \sum_{j=0}^{m_2} \left( Q_{ij} - \sum_{h=0}^{n_1} \sum_{l=0}^{n_2} B_h(u_i) B_l(v_j) P_{hl} \right)^2. \end{aligned} \quad (4.1)$$

可求得 (4.1) 关于变量  $P_{ab}$  的偏导数为

$$\frac{\partial F}{\partial P_{ab}} = -2 \sum_{i=0}^{m_1} \sum_{j=0}^{m_2} B_a(u_i) B_b(v_j) \left( Q_{ij} - \sum_{h=0}^{n_1} \sum_{l=0}^{n_2} B_h(u_i) B_l(v_j) P_{hl} \right).$$

对其继续关于变量  $P_{cd}$  求导可得

$$\frac{\partial^2 F}{\partial P_{cd} \partial P_{ab}} = 2 \sum_{i=0}^{m_1} \sum_{j=0}^{m_2} B_c(u_i) B_d(v_j) B_a(u_i) B_b(v_j). \quad (4.2)$$

令  $C_1$  为基函数  $\{B_h(u)\}_{h=0}^{n_1}$  在参数值  $\{u_i\}_{i=0}^{m_1}$  上的配置矩阵,  $C_2$  为基函数  $\{B_l(v)\}_{l=0}^{n_2}$  在参数值  $\{v_j\}_{j=0}^{m_2}$  上的配置矩阵. 根据 (4.2), 可以得到多元函数  $F$  的  $(n_1 + 1) \cdot (n_2 + 1)$  阶 Hessian 矩阵为  $H(F) = 2C^T C$ , 其中  $C = C_1 \otimes C_2$ , 运算  $\otimes$  表示 Kronecker 积. 类似于定理 3.2 和定理 3.3 的证明, 可证得多元函数  $F$  的 Hessian 矩阵为正定矩阵, 故  $F$  存在极小值, 可采用 Newton 迭代法求解其极小值. 令数据点列为

$$Q = [Q_{00}, Q_{01}, \dots, Q_{0m_2}, Q_{10}, \dots, Q_{m_1 m_2}]^T,$$

基于 Newton 迭代法, 得到迭代格式

$$P^{k+1} = P^k + H(F)^{-1} h(F) = P^k + (C^T C)^{-1} [C^T (Q - CP^k)]. \quad (4.3)$$

其中  $P^k = [P_{00}^k, P_{01}^k, \dots, P_{0n_2}^k, P_{10}^k, \dots, P_{n_1 n_2}^k]^T$  为第  $k$  次迭代后的控制点列,  $P^{k+1}$  为第  $k+1$  次迭代后的控制点列. 类似于曲线的迭代过程, 记数据点与曲面在对应参数值的差为

$$\{\delta_{ij} = Q_{ij} - P(u_i, v_j)\}_{i=0, j=0}^{m_1, m_2},$$

令  $B = C^T C$ , 假设经过  $k$  次迭代后, 得到混合曲面  $P^k(u, v)$ , 为了得到第  $k+1$  次混合曲面, 先计算数据点列的差向量

$$\delta^k = [\delta_{00}^k, \delta_{01}^k, \dots, \delta_{0m_2}^k, \delta_{10}^k, \dots, \delta_{m_1 m_2}^k]^T.$$

再计算控制顶点的差向量  $\Delta^k = B^{-1} C^T \delta^k$ , 于是得到第  $k+1$  次迭代后的控制点列为  $P^{k+1} = P^k + \Delta^k$ , 则有第  $k+1$  次迭代后的混合曲面为

$$P^{k+1}(u, v) = \sum_{h=0}^{n_1} \sum_{l=0}^{n_2} B_h(u) B_l(v) P_{hl}^{k+1}.$$

同曲线中的情况一样, 当  $m_1 = n_1, m_2 = n_2$  时, 迭代生成曲面序列的极限插值于给定数据点列. 当  $m_1 > n_1, m_2 > n_2$  时, 迭代生成的极限曲面为给定数据点列的最小二乘逼近结果.

对于矩阵  $B = C^T C$  的逆矩阵, 类似于曲线的处理方式, 依旧采用直接分解法和间接逼近法进行处理. 由  $B$  为正定矩阵, 对其进行奇异值分解

$$B = U \Sigma V^T = U \text{diag}(\sigma_{00}, \dots, \sigma_{0n_2}, \sigma_{10}, \dots, \sigma_{n_1 n_2}) V^T.$$

令  $W = \text{diag}(\frac{1}{\sigma_{00}}, \dots, \frac{1}{\sigma_{0n_2}}, \frac{1}{\sigma_{10}}, \dots, \frac{1}{\sigma_{n_1 n_2}})$ , 则曲面拟合的 NLSPIA 迭代格式为

$$P^{k+1} = P^k + VWU^T C^T (Q - CP^k). \quad (4.4)$$

迭代矩阵为  $D = I - VWU^T C^T C$ , 则  $D$  逼近零矩阵, 代表着奇异值分解的误差,  $D$  中所有元素为 0 或接近 0 的误差值  $\tau$ , 则有  $\lim_{k \rightarrow \infty} D^k = O$ , 其中  $O$  为零矩阵, 故曲面的 NLSPIA 迭代格式 (4.4) 收敛, 且收敛结果为待拟合数据点列的最小二乘逼近结果.

在间接逼近法中, 为了降低计算量, 不直接构造矩阵  $B$  的逼近矩阵  $N$ . 因为矩阵  $B$  的阶数为  $(n_1 + 1) \cdot (n_2 + 1)$ , 又有  $B = B_1 \otimes B_2$ , 而  $B_1$  和  $B_2$  的阶数分别为  $n_1 + 1$  和  $n_2 + 1$ . 故先

构造  $B_1 = C_1^T C_1$  的逼近矩阵  $N_1 = B_{q_1} + A_1$  和  $B_2 = C_2^T C_2$  的逼近矩阵  $N_2 = B_{q_2} + A_2$ , 然后通过 Kronecker 积得到  $B = C^T C$  的逼近矩阵为  $N = N_1 \otimes N_2$ . 由定理 3.5 可知,  $N_1$  和  $N_2$  都为正定矩阵, 所以  $N$  也为正定矩阵<sup>[18]</sup>. 此时, 曲面拟合的 INLSPIA 迭代格式则为

$$P^{k+1} = P^k + N^{-1}[C^T(Q - CP^k)], \quad (4.5)$$

每一步迭代中的  $N^{-1}[C^T(Q - CP^k)]$ , 等价求解  $NX^k = C^T(Q - CP^k)$ , 采用 CG 法迭代求取  $X^k$  近似解  $\tilde{X}^{(l,k)}$ , 其中  $l$  为内迭代 CG 法的迭代次数,  $k$  为外迭代最小二乘渐进迭代逼近的迭代次数. 基于此, 第  $k+1$  次迭代后的控制顶点为

$$\tilde{P}^{k+1} = \tilde{P}^k + \tilde{X}^{(l,k)}.$$

设  $P^*$  为  $C^T C X = C^T Q$  的精确解, 则类似定理 3.6 的证明, INLSPIA 生成的迭代序列  $\{\tilde{P}^k\}_{k=0}^\infty$  收敛到  $P^*$ , 即迭代生成的极限曲面为待拟合的数据点列的最小二乘逼近结果. 跟曲线的 INLSPIA 相同, 为了平衡计算复杂度和收敛速度, 对于矩阵  $N_1$  和  $N_2$ , 一般选择  $q_1$  和  $q_2$  分别为  $q_1 = \lfloor \frac{n_1+1}{2} \rfloor$ ,  $q_2 = \lfloor \frac{n_2+1}{2} \rfloor$ , 其中  $n_1+1$  和  $n_2+1$  分别为矩阵  $B_1$  和  $B_2$  的阶数.

采用 NLSPIA 与 INLSPIA 迭代格式的曲面拟合算法同算法 3.1 与算法 3.2 类似, 在此不再赘述.

## 5. 算法实现与实验分析

在本文的数值实验中, 标准全正基函数均选择 B 样条基函数. 本节首先我们将给出 NLSPIA 与 INLSPIA 算法实现中参数的选择与样条函数的节点设定. 随后用一些具体实例展现本文方法的有效性, 并与经典 LSPIA 进行对比分析.

### 5.1. 算法实现

设 B 样条基函数的次数为  $p$ , 根据 B 样条基函数的局部支集性质, 矩阵  $B = C^T C$  为  $2p+1$  对角矩阵

$$B = C^T C = \begin{bmatrix} b_{00} & \cdots & b_{0p} & 0 & \cdots & 0 \\ \vdots & \ddots & \ddots & b_{1,p+1} & \ddots & \vdots \\ b_{p0} & \ddots & \ddots & \ddots & \ddots & 0 \\ 0 & b_{p+1,1} & \ddots & \ddots & \ddots & b_{n-p,n} \\ \vdots & \ddots & \ddots & \ddots & \ddots & \vdots \\ 0 & \cdots & 0 & b_{n,n-p} & \cdots & b_{nn} \end{bmatrix}.$$

对于  $B$  的逼近矩阵  $N = B_q + A$ , 其中矩阵  $B_q$  为  $B$  的  $2q+1$  对角线上的元素形成的矩阵. 所以对于 B 样条基函数, 当  $p \leq q \leq n, q \in \mathbb{Z}$  时, 矩阵  $A$  为零矩阵, 此时逼近矩阵  $N = B_q = B$ . 当  $q > p$  时, 默认都按  $q = p$  处理, 所以可将  $q$  的取值范围缩小到  $0 \leq q \leq p, q \in \mathbb{Z}$ . 此外, 前文提到, 为了平衡收敛速度和计算复杂度, 取  $q = \lfloor \frac{n+1}{2} \rfloor$ , 其中  $n+1$  为矩阵  $C^T C$  的阶数, 也代表着控制顶点的个数和基函数的个数. 一般情况下, B 样条基函数的次数  $p$  都远小于控制顶点个数  $n+1$ , 所以如果  $p \leq \lfloor \frac{n+1}{2} \rfloor$ , 理论上  $q$  可直接取为  $q = p$ ; 此时矩阵  $N = C^T C$ , 收敛

速度达到最大. 而若  $p > \lfloor \frac{n+1}{2} \rfloor$ , 则依旧取  $q = \lfloor \frac{n+1}{2} \rfloor$ . 又因为  $p < n$ , 所以  $\lfloor \frac{p}{2} \rfloor$  逼近  $p$  的程度大于  $\lfloor \frac{n+1}{2} \rfloor$  逼近  $n+1$  的程度, 因此, 在实际应用中, 无论  $p$  是否小于  $\lfloor \frac{n+1}{2} \rfloor$ , 都取  $q = \lfloor \frac{p}{2} \rfloor$ , 在收敛速度不降低的同时, 进一步降低计算复杂度.

同经典 LSPIA 一样<sup>[12]</sup>, 曲线拟合数据点列对应的参数值采用累加弦长参数法求得

$$\begin{aligned} t_0 &= 0, \quad t_m = 1 \\ t_i &= t_{i-1} + \frac{\|Q_i - Q_{i-1}\|}{D} \quad (i = 1, 2, \dots, m-1). \end{aligned}$$

其中  $D = \sum_{i=0}^m \|Q_i - Q_{i-1}\|$  是总弦长. 曲面拟合参数值类似于张量积形式选取. 此外,  $p$  次 B 样条曲线的样条节点选择为

$$\underbrace{\{0, 0, \dots, 0\}}_{p+1}, \bar{t}_{p+1}, \bar{t}_{p+2}, \dots, \bar{t}_n, \underbrace{\{1, 1, \dots, 1\}}_{p+1},$$

其中

$$\begin{aligned} \bar{t}_{j+3} &= (1 - \alpha)t_{i-1} + \alpha t_i, \quad j = p-2, \dots, n-3, \\ i &= [jd], \quad \alpha = jd - i, \quad d = \frac{m+1}{n-2}. \end{aligned}$$

虽然初始控制顶点可以任意选择, 且并不会影响 LSPIA 迭代的收敛性<sup>[12]</sup>, 但为了比较不同方法之间的收敛速度, 初始控制顶点  $\{P_i\}_{i=0}^n$  都依照文献 [12] 中的方法统一确定, 即

$$\begin{aligned} P_i &= Q_{f(i)}, f(i) = \left\lfloor \frac{(m+1)i}{n} \right\rfloor, \\ P_0 &= Q_0, P_n = Q_m, i = 1, 2, \dots, n-1. \end{aligned}$$

对于曲面拟合, 有数据点列  $\{Q_{ij}\}_{i=0, j=0}^{m_1, m_2}$ . 初始控制顶点  $\{P_{hl}\}_{h=0, l=0}^{n_1, n_2}$  选择为

$$\begin{aligned} P_{0,0} &= Q_{0,0}, P_{n_1,0} = Q_{m_1,0}, P_{0,n_2} = Q_{0,m_2}, P_{n_1,n_2} = Q_{m_1,m_2}, \\ P_{i,j} &= Q_{f_1(i)f_2(j)}, i = 1, 2, \dots, n_1-1, j = 1, 2, \dots, n_2-1. \end{aligned}$$

其中  $f_1(i) = \left\lfloor \frac{(m_1+1)i}{n_1} \right\rfloor, f_2(j) = \left\lfloor \frac{(m_2+1)j}{n_2} \right\rfloor$ .

## 5.2. 数值实例

本小节将用一些典型数值实例来验证本文算法的有效性, 并分析相关参数对迭代结果的影响. 在如下实例中, 基函数均选择 3 次 B 样条基函数, 即样条基函数次数  $p = 3$ .

### 5.2.1. INLSPIA 中 $\varepsilon$ 值对迭代结果的影响

**例 1.** 心型线  $\{x(t), y(t)\} = \{(1 - \cos t) \sin t, (1 - \cos t) \cos t\}, t \in [0, 2\pi]$  上取得 101 个数据点, 从中选取 12 个点作为初始控制顶点 (如图 1 所示).

在 INLSPIA 中, 内迭代 CG 法的迭代终止条件为 (3.10)

$$\frac{\|C^T(Q - C\tilde{P}^k) - N\tilde{X}^{(l,k)}\|}{\|C^T(Q - C\tilde{P}^k)\|} < \varepsilon,$$

当  $\varepsilon = 0$  时, 有

$$C^T(Q - C\tilde{P}^k) - N\tilde{X}^{(t,k)} = 0,$$

此时 CG 法求得调整向量的精确解.  $\varepsilon$  越小, 得到的近似解就越精确, 但相应的计算复杂度会增加. 令  $C^T C X = C^T Q$  的精确解为  $P^\infty$ , 则第  $k$  步迭代的迭代误差为:

$$e_k = \max \left\| \tilde{P}^k - P^\infty \right\|_2.$$

取  $q = \lfloor \frac{p}{2} \rfloor$ , 表 1 为在不同  $\varepsilon$  下, 运用 INLSPIA 逼近例 1 心型线的迭代误差. 图 1 为  $\varepsilon = 1 \times 10^{-3}$  时, 运用 INLSPIA 分别迭代 1,5,10 次得到的混合曲线. 记第  $k$  步的拟合误差为

$$E_k = \sum_{j=0}^m \left\| Q_j - \sum_{i=0}^n B_i(t_j) P_i^k \right\|_2^2,$$

令  $E$  为迭代拟合精度, 当满足  $|E_k - E_{k-1}| < E$  时, 迭代停止, 此时得到的迭代混合曲线  $P^k(t)$  便可记为极限曲线, 即  $P^\infty(t) = P^k(t)$ . 表 2 为在不同迭代拟合精度  $E$  下, 取不同  $\varepsilon$  的 INLSPIA 逼近心型线达到拟合精度的运行时间. 从表 1 可以看到, 随着  $\varepsilon$  的减小, 迭代误差  $e_k$  的确不断减小, 不过减小的幅度也越来越小. 主要影响迭代误差的还是外迭代的最小二乘渐进迭代逼近. 但同时从表 2 也可以得到,  $\varepsilon$  越大, 所需运行时间就越少, 也即收敛速度越快, 这是因为内迭代 CG 法所消耗时间变少. 所以不用刻意追求每一步迭代中

$$NX^k = C^T(Q - CP^k)$$

的精确解, 只需能够快速求得其近似解, 再通过外迭代不断迭代逼近至  $P^\infty$ . 综合考虑, 后文的数值实验中, INLSPIA 内迭代 CG 法的迭代终止条件就取为  $\varepsilon = 1 \times 10^{-3}$ .

表 1 例 1 中 INLSPIA 在不同 CG 法迭代终止条件  $\varepsilon$  下的迭代误差  $e_k$

	$\varepsilon = 1 \times 10^{-1}$	$\varepsilon = 1 \times 10^{-4}$	$\varepsilon = 1 \times 10^{-7}$
$k = 1$	3.8961e-01	3.3664e-01	3.3456e-01
$k = 5$	7.9106e-04	6.0880e-04	6.0848e-04
$k = 10$	6.6830e-07	6.6144e-07	6.6099e-07

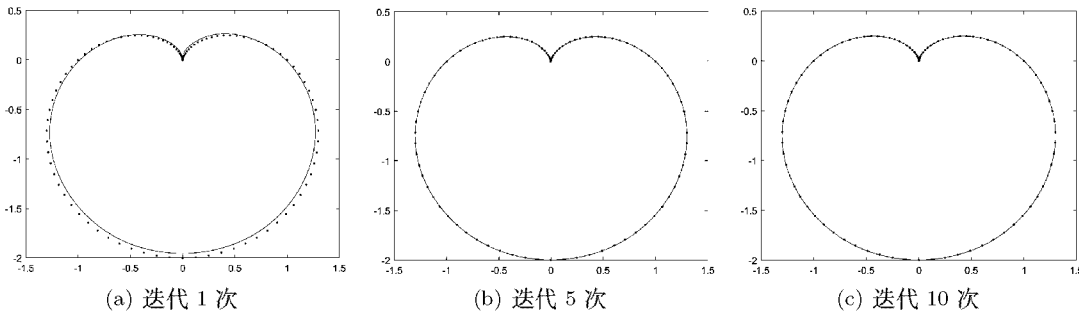


图 1 运用 INLSPIA 逼近例 1 心型线的迭代结果

表 2 例 1 中 INLSPIA 在不同  $\varepsilon$  下达到迭代拟合精度  $E$  的运行时间  $t$

	$\varepsilon = 1 \times 10^{-1}$	$\varepsilon = 1 \times 10^{-4}$	$\varepsilon = 1 \times 10^{-7}$
$E = 1 \times 10^{-4}$	1.4723e-03s	2.8613e-03s	4.6621e-03s
$E = 1 \times 10^{-7}$	8.2971e-03s	9.7784e-03s	1.1923e-02s
$E = 1 \times 10^{-10}$	1.3194e-02s	1.4562e-02s	1.7232e-02s

5.2.2. INLSPIA 中  $q$  值对迭代结果的影响

例 2. 例 2 螺旋线  $\{x(t), y(t)\} = \{\cos \pi t, \sin \pi t, \frac{t}{6}\}, t \in [0, 2\pi]$  上取得 251 个数据点, 从中选取 35 个点作为初始控制顶点 (如图 2 所示).

除 CG 法的迭代终止条件  $\varepsilon$  外, INLSPIA 的收敛速度还被逼近矩阵  $N = B_q + A$  中  $q$  的取值所影响,  $q$  越大, 收敛速度越快,  $q$  越小, 计算复杂度越小. 因为从 251 个数据点中选取了 35 个控制顶点, 所以有  $0 \leq q \leq 34, q \in \mathbb{Z}$ . 又因为基函数为 3 次 B 样条基函数, 由 5.1 节中关于 B 样条基函数中  $q$  的取值的讨论, 当  $3 \leq q \leq 34, q \in \mathbb{Z}$  时, 不同的  $q$  值得到的逼近矩阵  $N$  是相同的, 所以  $q$  的取值范围可缩小到  $0 \leq q \leq 3, q \in \mathbb{Z}$ . 表 3 为在不同迭代拟合精度  $E$  下, 取不同的  $q$ , 运用 INLSPIA 逼近螺旋线达到拟合精度所需的迭代次数和运行时间. 从表 3 可以看到, 随着  $q$  的增长, 迭代所需次数显著减小, 在拟合精度要求高时尤为明显; 不过相应的, 所需的时间也不断增加. 图 2 是在迭代拟合精度  $E = 1 \times 10^{-10}$  时, 取不同  $q$  值的 INLSPIA 得到的迭代结果. 若不做特殊说明, 对于  $p$  次 B 样条基函数, INLSPIA 中的  $q$  就取为  $q = \lfloor \frac{p}{2} \rfloor$ .

表 3 例 2 中 INLSPIA 在不同  $q$  下达到拟合精度  $E$  所需的迭代次数  $k$  和运行时间  $t$

	$E = 1 \times 10^{-04}$		$E = 1 \times 10^{-10}$		$E = 1 \times 10^{-16}$	
	$k$	$t$	$k$	$t$	$k$	$t$
$q = 0$	16	0.1716 s	138	0.2097 s	263	0.2517 s
$q = 1$	4	0.1509 s	9	0.1877 s	15	0.2117 s
$q = 2$	1	0.1434 s	3	0.1664 s	5	0.1942 s

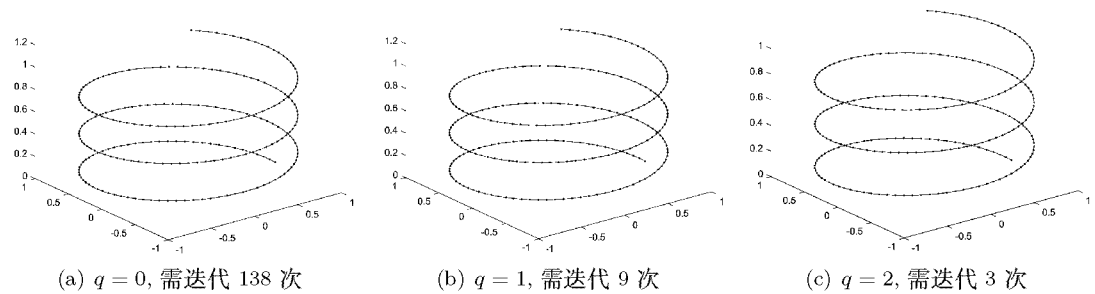


图 2 INLSPIA 取不同  $q$  值逼近例 2 螺旋线的迭代结果

5.2.3. 曲线 LSPIA 与 NLSPIA 的比较

例 3. 四叶线  $\{x(t), y(t)\} = \{\cos 2t \cos t, \cos 2t \sin t\}, t \in [0, 2\pi]$  上取得 301 个数据点, 从中选取 101 个点作为初始控制顶点 (如图 3 所示).

分别运用经典 LSPIA<sup>[12]</sup> 和 NLSPIA 去逼近四叶线, 得到的迭代混合曲线如图 3 所示. 表 4 为在不同迭代次数下, 运用两种方法得到的迭代误差  $e_k = \max \|P^k - P^\infty\|_2$ . 从表 4 可以看到, 相较于经典 LSPIA, NLSPIA 能大幅加快收敛速度. 在 LSPIA 的迭代误差随着迭代次数的增加而缓慢地减少时, NLSPIA 已经在有限次迭代中取得理想的结果. 表 5 为不同的迭代拟合精度  $E$  下, 经典 LSPIA 和 NLSPIA 逼近例 3 的四叶线达到不同拟合精度的运行时间. 从表 5 看到, NLSPIA 所需的时间是要少于 LSPIA, 这也代表着 NLSPIA 的收敛速度优于 LSPIA. 虽然 NLSPIA 中 Hessian 矩阵的奇异值分解需要花费一部分时间, 但是经过分解后, 最小二乘渐进迭代逼近所需的次数显著减少, 最后使得 NLSPIA 总的消耗时间仍小于经典 LSPIA.

5.2.4. 曲线 LSPIA 与 INLSPIA 的比较

例 4. 参数曲线  $\{x(t), y(t)\} = \{\cos t + \cos 3t, \sin t + \sin 5t\}, t \in [0, 2\pi]$  上取得 651 个数据点, 从中选取 221 个点作为初始控制顶点 (如图 4 所示).

图 4 为运用 LSPIA<sup>[12]</sup> 和 INLSPIA 逼近例 4 所得到的迭代混合曲线, 两种方法所得到的迭代误差如表 6 所示, 达到不同的迭代拟合精度所需的时间如表 7 所示. 从表 6 可以看到, 在相同的迭代次数下, INLSPIA 的迭代误差要小于 LSPIA. 表 7 也表明 INLSPIA 所需的时间比经典 LSPIA 少, 所以 INLSPIA 的收敛速度也要优于 LSPIA. 类似于 NLSPIA, INLSPIA 虽然也需要消耗一部分时间进行内迭代 CG 法求解, 但其使外迭代最小二乘渐进迭代逼近所需的次数减少, 使得最后的综合时间仍少于经典 LSPIA.

表 4 例 3 中 LSPIA 和 NLSPIA 的迭代误差  $e_k$

	$k = 1$	$k = 5$	$k = 10$
LSPIA	3.7211e-01	2.7226e-01	1.8579e-01
NLSPIA	<b>3.8612e-04</b>	<b>1.3506e-15</b>	<b>8.8991e-16</b>

表 5 例 3 中 LSPIA 和 NLSPIA 达到拟合精度  $E$  所需的运行时间  $t$

	$E = 1 \times 10^{-4}$	$E = 1 \times 10^{-7}$	$E = 1 \times 10^{-10}$
LSPIA	2.7615e-02s	3.9023e-02s	5.3312e-02s
NLSPIA	<b>2.6151e-02s</b>	<b>3.1975e-02s</b>	<b>4.81331e-02s</b>

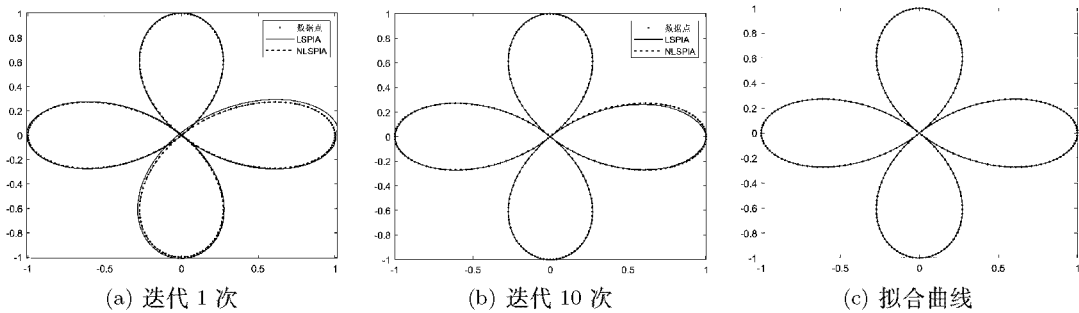


图 3 例 3 的 LSPIA 和 NLSPIA 的迭代结果



表 6 例 4 中 LSPIA 和 INLSPIA 的迭代误差  $e_k$

	$k = 1$	$k = 5$	$k = 10$
LSPIA	5.1265e-01	2.2562e-01	1.1028e-02
INLSPIA	<b>8.0388e-03</b>	<b>3.1861e-05</b>	<b>7.0463e-08</b>

表 7 例 4 中 LSPIA 和 INLSPIA 达到拟合精度  $E$  所需的运行时间  $t$

	$E = 1 \times 10^{-4}$	$E = 1 \times 10^{-7}$	$E = 1 \times 10^{-10}$
LSPIA	0.1423s	0.1674s	0.2185s
INLSPIA	<b>0.1396s</b>	<b>0.1541s</b>	<b>0.1655s</b>

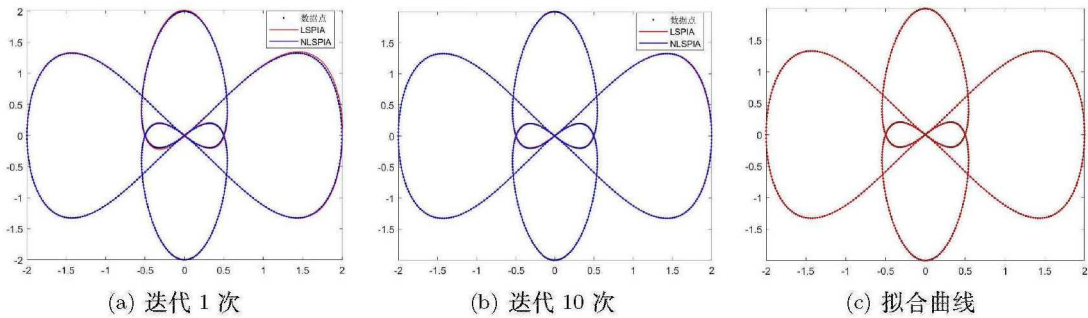


图 4 例 4 的 LSPIA 和 INLSPIA 的迭代结果

表 8 例 3 中 NLSPIA 和 INLSPIA 的迭代误差  $e_k$

	NLSPIA	INLSPIA	INLSPIA ( $q = p$ )
$k = 1$	<b>3.8612e-04</b>	3.7008e-03	5.1387e-04
$k = 5$	<b>1.3506e-15</b>	8.5767e-06	2.7448e-15
$k = 10$	8.8991e-16	1.3034e-08	<b>8.8829e-16</b>

表 9 例 3 中 NLSPIA 和 INLSPIA 达到迭代拟合精度  $E$  的运行时间  $t$

	NLSPIA	INLSPIA	INLSPIA ( $q = p$ )
$E = 1 \times 10^{-4}$	2.6151e-02s	<b>2.0946e-02s</b>	2.3354e-02s
$E = 1 \times 10^{-7}$	<b>3.1975e-02s</b>	3.2243e-02s	3.4384e-02s
$E = 1 \times 10^{-10}$	4.8133e-02s	<b>4.6254e-02s</b>	4.9437e-02s

5.2.5. 曲线 NLSPIA 与 INLSPIA 的比较

运用 NLSPIA 与  $q = \lfloor \frac{p}{2} \rfloor$  和  $q = p$  的 INLSPIA 去逼近例 3 中的四叶线, 得到的迭代误差  $e_k$  如表 8 所示, 达到拟合精度所需的运行时间如表 9 所示. 从表 8 可以得到, 取  $q = p$  时 INLSPIA 和 NLSPIA 的迭代误差相近, 因为此时逼近矩阵  $N = C^T C$ , 奇异值分解和内迭代 CG 法处理的都为同一个矩阵. 从表 9 看到,  $q = p$  时, INLSPIA 和 NLSPIA 所需的运行时间也极为接近, 虽然  $q = \lfloor \frac{p}{2} \rfloor$  时, INLSPIA 的迭代误差是三者中最大的, 但大部分情况下, 其运行时间是小于其它两者的, 这也符合 5.2.2 节中讨论的  $q$  值对迭代的影响.

5.2.6. 曲面拟合

下面以两个曲面拟合的数值实例来验证曲面 NLSPIA 和 INLSPIA 的有效性和收敛性.

例 5. 曲面  $T(u, v) = \frac{\sin \sqrt{u^2 + v^2}}{\sqrt{u^2 + v^2}}, u, v \in [-8, 10]$  上采样 12221 个数据点, 从中选取 120 个点作为初始控制顶点 (如图 5 所示).

例 6. 瓶子轮廓曲面 (见图 6(a)) 上采样 12986 个数据点, 从中选取 2352 个点作为初始控制顶点.

经典 LSPIA<sup>[12]</sup>、NLSPIA 和 INLSPIA 逼近例 5 和例 6 中曲面的迭代结果如图 5 和图 6 所示, 其中红色为数据点列, 从图中可以看到, NLSPIA 和 INLSPIA 很好地逼近给定的数据点列. 对比图 5 和图 6 中三种方法得到的迭代混合曲面, NLSPIA 和 INLSPIA 的收敛速度是明显快于经典 LSPIA 的. 且再对比图 5(b)、图 5(c) 和图 5(d), 可以发现, 经典 LSPIA 经过 25 次迭代后仍然有一些边缘部分的逼近效果没有 NLSPIA 和 INLSPIA 理想. 三种方法逼近例 5 和例 6 中曲面的迭代误差  $e_k$  如表 10 和表 11 所示, 从表 10 和表 11 中可以看到, 相同的迭代步数下, NLSPIA 和 INLSPIA 得到的迭代误差比经典 LSPIA 的小, 且误差减小的速度也优于经典 LSPIA. 同样的, 类似于 5.2.5 节中的曲线情形, 取  $q = \lfloor \frac{p}{2} \rfloor$  时, INLSPIA 迭代的收敛速度没有 NLSPIA 快, 只有  $q = p$  时, 其收敛速度是才是和 NLSPIA 接近的. 通过对比图 5(c) 和图 5(d) 也可以明显地看出, 相同的迭代次数下, INLSPIA 仍有边缘部分的逼近效果没有 NLSPIA 的逼近效果理想.

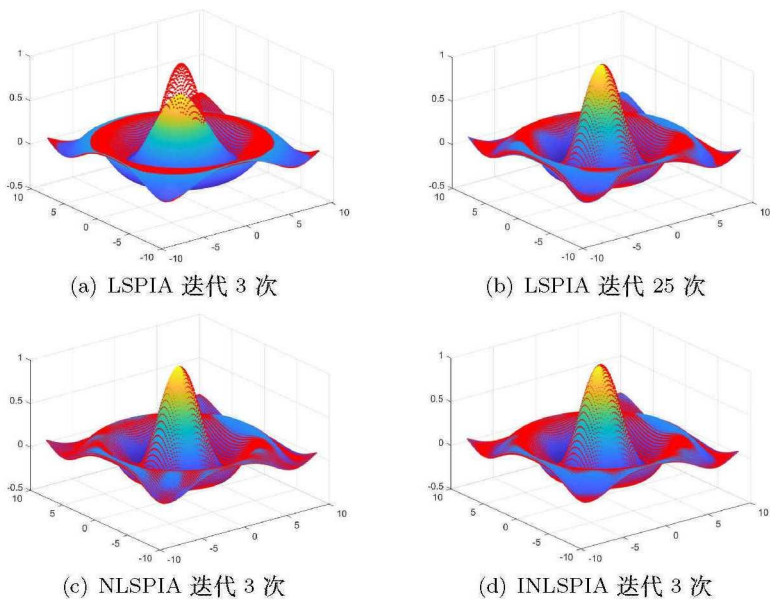


图 5 例 5 的 LSPIA、NLSPIA 和 INLSPIA 的迭代结果

表 10 例 5 中 LSPIA、NLSPIA 和 INLSPIA 的迭代误差  $e_k$

	LSPIA	NLSPIA	INLSPIA
$k = 1$	1.2473e03	<b>3.4764e-01</b>	9.2254
$k = 10$	2.3877	<b>1.3081e-07</b>	8.5837e-05
$k = 25$	5.1840e-01	<b>1.1374e-12</b>	2.2070e-09

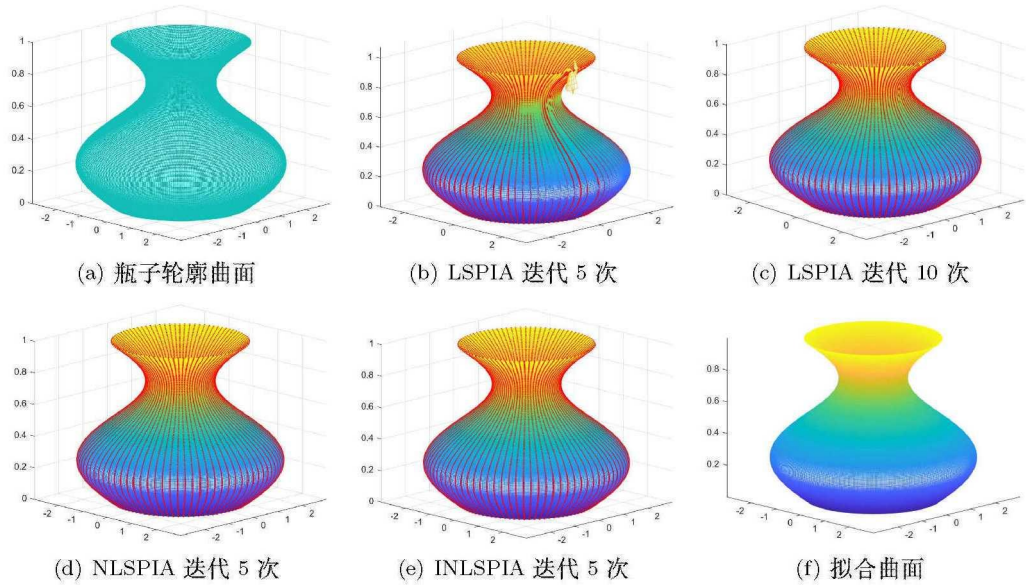


图 6 例 6 的 LSPIA、NLSPIA 和 INLSPIA 的迭代结果

表 11 例 6 中 LSPIA、NLSPIA 和 INLSPIA 的迭代误差  $e_k$

	LSPIA	NLSPIA	INLSPIA
$k = 1$	3.5939	<b>2.5116e-02</b>	4.0774e-01
$k = 10$	3.0736e-01	<b>2.4039e-09</b>	1.3410e-05
$k = 25$	2,2135e-02	<b>1.5177e-14</b>	3.1713e-10

6. 总结与展望

本文主要基于 Newton 迭代法求解  $n + 1$  元函数  $\sum_{i=0}^m (Q_i - P(t_i))^2$  的极小值, 提出两种新型 LSPIA 方法, 它们分别对 Hessian 矩阵  $C^T C$  进行奇异值分解和对每一步迭代调整向量采用 CG 法迭代求解, 分别记为 NLSPIA 和 INLSPIA. 数值实验结果表明, 在相同条件下, 本文提出的方法在收敛速度上优于经典的 LSPIA 算法, 且加速效果显著.

本文所提出的方法的核心本质是在迭代中计算  $(C^T C)^{-1} C^T (Q - CP^k)$ , 而提出的两种解决方法中, 直接分解法从矩阵  $C^T C$  本身出发, 构造易于求逆的矩阵去近似  $C^T C$ . 间接逼近法先简化  $C^T C$ , 得到逼近矩阵  $N$ , 再运用 CG 法求解每一步迭代中  $NX^k = C^T (Q - CP^k)$  的近似解. 作为未来的工作, 将构造更好的处理  $(C^T C)^{-1} C^T (Q - CP^k)$  的方法, 使得迭代收敛速度不降低的同时, 能够更进一步地减少计算量. 此外, 当数据点数和控制点数相等, 即  $m = n$  时, 本文提出的方法与 PIA 的收敛结果相同; 当数据点数大于控制点数时, 即  $m > n$  时, 本文提出的方法与 LSPIA 的收敛结果相同. 若需要调整的数据点数小于控制点数, 很自然会想到局部 PIA(LPIA) [19], 所以对于此种情况, 本文所提出的方法是否收敛, 得到的结果和 LPIA 是否相同也将作为未来的研究工作.

附录 A

在本节中, 给出前文涉及的所有定理的详细理论证明.

### A.1. 定理 3.1 的证明

**定理 3.1.** 标准全正基函数  $\{B_i(t)\}_{i=0}^n$  在参数值  $\{t_j\}_{j=0}^m$  上的配置矩阵  $C$  是列满秩矩阵, 其中  $n \leq m$ .

**证明.** 配置矩阵  $C$  是一个  $(m+1) \times (n+1)$  阶矩阵, 行数  $m+1$  是数据点个数, 列数  $n+1$  是控制顶点个数. 根据矩阵  $C$  的定义 (2.1), 取  $C$  的  $n+1$  阶子矩阵  $C_{n+1}$  如下

$$C_{n+1} = \begin{bmatrix} B_0(t_{n_0}) & B_1(t_{n_0}) & \cdots & B_n(t_{n_0}) \\ B_0(t_{n_1}) & B_1(t_{n_1}) & \cdots & B_n(t_{n_1}) \\ \vdots & \vdots & \ddots & \vdots \\ B_0(t_{n_n}) & B_1(t_{n_n}) & \cdots & B_n(t_{n_n}) \end{bmatrix}.$$

因为基函数  $\{B_i(t)\}_{i=0}^n$  为标准全正基, 所以子矩阵  $C_{n+1}$  为非奇异矩阵且其所有特征值都在  $(0, 1)$  内<sup>[4]</sup>, 又矩阵  $C$  的秩满足条件

$$0 \leq r(C) \leq \min(m+1, n+1) = n+1,$$

可得  $r(C) = n+1$ , 即矩阵  $C$  为列满秩矩阵.

### A.2. 定理 3.2 的证明

**定理 3.2.** 多元函数  $f(P_0, P_1, \cdots, P_n)$  的 Hessian 矩阵满足  $H(f) = 2C^T C$ , 且为正定矩阵, 其中  $C$  为基函数  $\{B_i(t)\}_{i=0}^n$  在参数值  $\{t_j\}_{j=0}^m$  上的配置矩阵.

**证明.** 根据多元函数  $f$  的表达式 (3.1):

$$f(P_0, P_1, \cdots, P_n) = \sum_{i=0}^m (Q_i - P(t_i))^2 = \sum_{i=0}^m \left( Q_i - \sum_{j=0}^n B_j(t_i) P_j \right)^2.$$

可以得到多元函数  $f$  关于变量  $P_a$  的偏导数为

$$\frac{\partial f}{\partial P_a} = -2 \sum_{i=0}^m B_a(t_i) (Q_i - \sum_{j=0}^n B_j(t_i) P_j), \quad (\text{A.1})$$

其中  $Q_i - \sum_{j=0}^n B_j(t_i) P_j$  就是数据点的调整误差, 将其表示为  $\delta_i$ , 则多元函数  $f$  关于各变量的偏导数为

$$h(f) = \left[ \frac{\partial f}{\partial P_0}, \frac{\partial f}{\partial P_1}, \cdots, \frac{\partial f}{\partial P_n} \right]^T = -2 \left[ \sum_{i=0}^m B_0(t_i) \delta_i, \sum_{i=0}^m B_1(t_i) \delta_i, \cdots, \sum_{i=0}^m B_n(t_i) \delta_i \right]^T.$$

更进一步, 对式 (A.1) 又关于变量  $P_b$  求导得到

$$\frac{\partial^2 f}{\partial P_b \partial P_a} = 2 \sum_{i=0}^m B_b(t_i) B_a(t_i). \quad (\text{A.2})$$

于是可得多元函数  $f$  的 Hessian 矩阵为

$$H(f) = \begin{bmatrix} \frac{\partial^2 f}{\partial P_0^2} & \frac{\partial^2 f}{\partial P_0 \partial P_1} & \cdots & \frac{\partial^2 f}{\partial P_0 \partial P_n} \\ \frac{\partial^2 f}{\partial P_1 \partial P_0} & \frac{\partial^2 f}{\partial P_1^2} & \cdots & \frac{\partial^2 f}{\partial P_1 \partial P_n} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial^2 f}{\partial P_n \partial P_0} & \frac{\partial^2 f}{\partial P_n \partial P_1} & \cdots & \frac{\partial^2 f}{\partial P_n^2} \end{bmatrix}.$$

将式 (A.2) 代入可得

$$\begin{aligned} H(f) &= 2C^T C \\ &= 2 \begin{bmatrix} \sum_{i=0}^m B_0^2(t_i) & \sum_{i=0}^m B_0(t_i) B_1(t_i) & \cdots & \sum_{i=0}^m B_0(t_i) B_n(t_i) \\ \sum_{i=0}^m B_1(t_i) B_0(t_i) & \sum_{i=0}^m B_1^2(t_i) & \cdots & \sum_{i=0}^m B_1(t_i) B_n(t_i) \\ \vdots & \vdots & \ddots & \vdots \\ \sum_{i=0}^m B_n(t_i) B_0(t_i) & \sum_{i=0}^m B_n(t_i) B_1(t_i) & \cdots & \sum_{i=0}^m B_n^2(t_i) \end{bmatrix}, \end{aligned}$$

其中  $C$  是基函数  $\{B_i(t)\}_{i=0}^n$  在参数值  $\{t_j\}_{j=0}^m$  上的配置矩阵. 由于  $C$  为  $(m+1) \times (n+1)$  阶矩阵, 则  $C^T C$  为  $n+1$  阶方阵. 又因为  $(C^T C)^T = C^T C$ , 即  $C^T C$  为对称矩阵. 由定理 3.1 可知,  $C$  为列满秩矩阵, 即满足  $r(C) = n+1$ , 所以齐次线性方程组  $C\mathbf{x} = \mathbf{0}$  只有零解. 于是, 对于任意非零  $n+1$  维列向量  $\mathbf{x}$ , 满足  $C\mathbf{x} \neq \mathbf{0}$ , 则有下列式子成立:

$$(C\mathbf{x})^T (C\mathbf{x}) = \mathbf{x}^T (C^T C) \mathbf{x} > 0.$$

所以  $C^T C$  为正定矩阵, 也即函数  $f$  的 Hessian 矩阵  $H(f)$  为正定矩阵.

### A.3. 定理 3.3 的证明

**定理 3.3.** 多元函数  $f(P_0, P_1, \dots, P_n)$  在临界点处存在极小值.

**证明.** 由定理 3.2 可知, 函数  $f$  关于各变量的偏导数为

$$h(f) = -2 \left[ \sum_{i=0}^m B_0(t_i) \delta_i, \sum_{i=0}^m B_1(t_i) \delta_i, \dots, \sum_{i=0}^m B_n(t_i) \delta_i \right]^T,$$

其中  $\delta_j = Q_i - \sum_{j=0}^n B_j(t_i) P_j$ , 且函数  $f$  的 Hessian 矩阵  $H(f)$  为正定矩阵. 令满足偏导数  $h(f)$  为 0 的临界点为  $P^\lambda = [P_0^\lambda, P_1^\lambda, \dots, P_n^\lambda]^T$ , 即  $h(f(P^\lambda)) = 0$ ; 则根据多元函数极值的判断条件可得, 函数  $f$  在临界点  $P^\lambda$  处取得极小值.

### A.4. 定理 3.4 的证明

**定理 3.4.** NLSPIA 迭代格式 (3.3) 是收敛的, 且收敛结果为待拟合数据点列的最小二乘逼近结果.

证明. NLSPIA 的迭代格式 (3.3) 如下所示:

$$P^{k+1} = P^k + VWU^T C^T (Q - CP^k),$$

由  $B = C^T C$ , 则迭代格式等价于

$$P^{k+1} - B^{-1}C^T Q = (I - VWU^T B)(P^k - B^{-1}C^T Q).$$

迭代矩阵为  $D = I - VWU^T B$ , 易知矩阵  $VWU^T B$  的主对角元素全为 1 或  $1 + \tau$ , 其中  $\tau$  为接近 0 的误差值, 代表着奇异值分解所产生的误差. 所以矩阵  $D$  中的元素全为 0 或为接近 0 的误差值  $\tau$ , 即迭代矩阵  $D$  逼近零矩阵, 则有  $\lim_{k \rightarrow \infty} D^k = O_{n+1}$ ,  $O_{n+1}$  表示  $n+1$  阶零矩阵. 因此, 则有

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

等价于  $C^T C P^\infty = C^T Q$ , 即收敛结果为待拟合数据点列的最小二乘逼近结果.

#### A.5. 定理 3.5 的证明

**定理 3.5.** 矩阵  $B = C^T C$  按式 (3.7) 得到的逼近矩阵  $N$  是非奇异的且也为正定矩阵.

**证明.** 矩阵  $C$  为  $(m+1) \times (n+1)$  阶矩阵, 也即标准全正基函数  $\{B_i(t)\}_{i=0}^n$  在参数值  $\{t_j\}_{j=0}^m$  上的配置矩阵. 记  $C = (c_{ji})_{(m+1) \times (n+1)}$ , 则有  $c_{ji} = B_i(t_j) \geq 0$ . 记  $B = (b_{ij})_{n+1}$ , 则有  $b_{ij} = \sum_{k=0}^m B_i(t_k)B_j(t_k) \geq 0$ . 因为每一个基函数  $B_i(t)$  不会在所有参数值  $\{t_j\}_{j=0}^m$  处都取 0, 即存在一个或多个  $t_\alpha \in \{t_j\}_{j=0}^m$  使得  $B_i(t_\alpha) > 0$ , 所以矩阵  $B$  的对角元素满足

$$b_{ii} = \sum_{k=0}^m B_i^2(t_k) > 0, i = 0, 1, \dots, n.$$

又有  $B$  为正定矩阵,  $B_q$  为  $B$  中  $2q+1$  对角线上的元素所形成的对称矩阵. 对任意非零  $n+1$  维列向量  $\mathbf{x} = (x_0, x_1, \dots, x_n)^T$ , 满足

$$\begin{aligned} \mathbf{x}B_q\mathbf{x}^T &= \sum_{i=0}^n x_i^2 b_{ii}^2 + \sum_{i \in \Gamma_q} \sum_{j \in \Gamma_q} x_i x_j b_{ij} \\ \Gamma_q &= \{i, j \mid |i-j| \leq q, i \neq j, 0 \leq i, j \leq n, i, j \in \mathbb{Z}\}; \end{aligned}$$

其中  $\sum_{i=0}^n x_i^2 b_{ii}^2 > 0$ , 又  $b_{ij} \geq 0, i \neq j$ , 所以有

$$\sum_{i \in \Gamma_q} \sum_{j \in \Gamma_q} x_i x_j b_{ij} \geq 0,$$

所以  $\mathbf{x}B_q\mathbf{x}^T > 0$ , 即矩阵  $B_q$  为正定矩阵.

矩阵  $A$  为对角矩阵, 根据  $A$  的定义式 (3.6) 可得,  $A$  的对角元素满足条件  $a_{ii} \geq 0$ , 所以  $A$  为半正定矩阵. 则对任意非零  $n+1$  维列向量  $\mathbf{y}$ ,  $\mathbf{y}A\mathbf{y}^T \geq 0$  成立.

综上, 对于矩阵  $N = B_q + A$ , 有任意非零  $n+1$  维列向量  $\mathbf{x}$ , 满足:

$$\mathbf{x}N\mathbf{x}^T = \mathbf{x}B_q\mathbf{x}^T + \mathbf{x}A\mathbf{x}^T > 0,$$

所以矩阵  $N$  为正定矩阵, 其非奇异性得证.

### A.6. 定理 3.6 的证明

**定理 3.6.** 设  $P^*$  是  $C^T C X = C^T Q$  的精确解, 则 INLSPIA 生成的迭代序列  $\{\tilde{P}^k\}_{k=0}^\infty$  收敛于  $P^*$ , 也即 INLSPIA 迭代生成的曲线序列的极限曲线为待拟合数据点列的最小二乘逼近结果.

**证明.** 对于无调整向量近似解的迭代格式 (3.8)

$$P^{k+1} = P^k + N^{-1} C^T (Q - C P^k),$$

记其迭代过程中生成的迭代序列为  $\{P^k\}_{k=0}^\infty$ , 则可以得到

$$\begin{aligned} P^{k+1} &= P^k + N^{-1} C^T (Q - C P^k) \\ &= P^k + N^{-1} C^T (Q - C \tilde{P}^k + C \tilde{P}^k - C P^k) \\ &= P^k + N^{-1} \left[ C^T (Q - C \tilde{P}^k) + C^T C (\tilde{P}^k - P^k) \right]. \end{aligned} \quad (\text{A.3})$$

因为  $P^*$  是精确解, 即  $C^T C P^* = C^T Q$ , 所以式 (3.8) 中将  $C^T Q$  进行替换后得到

$$P^{k+1} = (I - N^{-1} C^T C) P^k + N^{-1} C^T C P^*,$$

这等价于

$$P^{k+1} - P^* = (I - N^{-1} C^T C) (P^k - P^*). \quad (\text{A.4})$$

由 INLSPIA 的迭代格式  $\tilde{P}^{k+1} = \tilde{P}^k + \tilde{X}^{(l,k)}$ , 再根据式 (A.3) 和 (A.4), 可以得到

$$\begin{aligned} \tilde{P}^{k+1} - P^* &= \tilde{P}^{k+1} - P^{k+1} + P^{k+1} - P^* \\ &= \tilde{P}^k + \tilde{X}^{(l,k)} - P^k - N^{-1} C^T (Q - C \tilde{P}^k) \\ &\quad - N^{-1} C^T C (\tilde{P}^k - P^k) + (I - N^{-1} C^T C) (P^k - P^*), \end{aligned}$$

化简得到

$$\tilde{P}^{k+1} - P^* = (I - N^{-1} C^T C) (\tilde{P}^k - P^*) - \left[ N^{-1} C^T (Q - C \tilde{P}^k) - \tilde{X}^{(l,k)} \right]. \quad (\text{A.5})$$

式 (A.5) 两边同时取范数, 由内迭代终止条件式 (3.10),

$$\frac{\|C^T (Q - C \tilde{P}^k) - N \tilde{X}^{(l,k)}\|}{\|C^T (Q - C \tilde{P}^k)\|} < \varepsilon$$

可以得到:

$$\|\tilde{P}^{k+1} - P^*\| \leq \|I - N^{-1} C^T C\| \|\tilde{P}^k - P^*\| + \varepsilon \|N^{-1}\| \|C^T (Q - C \tilde{P}^k)\|$$

再由  $P^*$  是  $C^T C P = C^T Q$  的精确解的条件, 进一步化简, 可得到

$$\|\tilde{P}^{k+1} - P^*\| \leq (\|I - N^{-1} C^T C\| + \varepsilon \|N^{-1} C^T C\|) \|\tilde{P}^k - P^*\|.$$

随着  $q$  在其取值范围内不断增大, 矩阵  $N$  越来越逼近矩阵  $C^T C$ , 因此  $N^{-1} C^T C$  越来越逼近单位矩阵  $E$ . 令  $\mu = \|I - N^{-1} C^T C\| + \varepsilon \|N^{-1} C^T C\|$ , 则有  $\mu < 1$ . 又因为

$$\|\tilde{P}^{k+1} - P^*\| \leq \mu \|\tilde{P}^k - P^*\| = \mu^{k+1} \|\tilde{P}^0 - P^*\|.$$

所以迭代序列  $\{\tilde{P}^k\}_{k=0}^\infty$  收敛到  $P^*$ , 即为待拟合数据点列的最小二乘逼近结果.

## 参 考 文 献

- [1] 齐东旭, 田自贤, 张玉心, 冯家斌. 曲线拟合的数值磨光方法 [J]. 数学学报, 1975(03): 173–184.
- [2] de Boor C. How does Agee's smoothing method work?[R] Washington D C: Army Research Office, 1979.
- [3] Lin Hongwei, Wang Guojin, Dong Chenshi. Constructing iterative non-uniform B-spline curve and surface to fit data points[J]. Science in China Series: Information Sciences, 2004, 47(3): 315–331.
- [4] Lin Hongwei, Bao Hujun, Wang Guojin. Totally positive bases and progressive iteration approximation[J]. Computers and Mathematics with Applications, 2005, 50(3): 575–586.
- [5] 史利民, 王仁宏. NURBS 曲线曲面拟合数据点的迭代算法 [J]. 数学研究与评论, 2006(04): 735–743.
- [6] Delgado J, Pea J M. Progressive iterative approximation and bases with the fastest convergence rates[J]. Computer Aided Geometric Design, 2006, 24(1): 10–18.
- [7] 陈杰, 王国瑾, 金聪健. 两类推广的渐进迭代逼近 [J]. 自动化学报, 2012, 38(01): 135–139.
- [8] Lu Lizheng. Weighted progressive iteration approximation and convergence analysis[J]. Computer Aided Geometric Design, 2009, 27(2): 129–137.
- [9] 张莉, 赵林, 檀结庆. 带互异权值的渐进迭代逼近算法及其应用 [J]. 浙江大学学报 (理学版), 2017, 44(01): 22–27.
- [10] 刘晓艳, 邓重阳. 非均匀三次 B 样条曲线插值的 Jacobi-PIA 算法 [J]. 计算机辅助设计与图形学学报, 2015, 27(03): 485–491.
- [11] Lin Hongwei, Zhang Zhiyu. An extended iterative format for the progressive-iteration approximation[J]. Computers & Graphics, 2011, 35(5): 967–975.
- [12] Deng Chongyang, Lin Hongwei. Progressive and iterative approximation for least squares B-spline curve and surface fitting[J]. Computer Aided Design, 2014, 47: 32–44.
- [13] 蔺宏伟. 几何迭代法及其应用综述 [J]. 计算机辅助设计与图形学学报, 2015, 27(04): 582–589.
- [14] Lin Hongwei, Maekawa Takashi, Deng Chongyang. Survey on geometric iterative methods and their applications[J]. Computer-Aided Design, 2018, 95: 40–51.
- [15] 王仁宏, 李崇君, 朱春钢. 计算几何教程 [M]. 北京: 科学出版社, 2008.
- [16] 邓少辉, 汪国昭. 渐进迭代逼近方法的数值分析 [J]. 计算机辅助设计与图形学学报, 2012, 24(07): 879–884.
- [17] Liu Chengzhi, Liu Zhongyun. Progressive iterative approximation with preconditioners[J]. Mathematics, 2020, 8(9): 1503.
- [18] 陈景良, 陈向晖. 特殊矩阵 [M]. 北京: 清华大学出版社, 2001.
- [19] Lin Hongwei. Local progressive-iterative approximation format for blending curves and patches[J]. Computer Aided Geometric Design, 2010, 27(4): 322–339.
- [20] Ando T. Totally positive matrices[J]. Linear Algebra and Its Applications, 1987, 90: 165–219.
- [21] 朱春钢, 李彩云. 数值逼近与计算几何 [M]. 北京: 高等出版社, 2020.
- [22] 张宏伟, 金光日, 施吉林, 董波. 计算机科学计算 [M]. 北京: 高等教育出版社, 2005.