

曲线曲面局部光顺渐进迭代逼近

林 佳¹⁾ 蔺 宏伟^{1,2)}

¹⁾(浙江大学数学科学学院 杭州 310058)

²⁾(浙江大学CAD&CG全国重点实验室 杭州 310058)

摘要 曲线曲面光顺问题源于工程设计与制造加工的实际需求,在汽车、航空航天、船舶制造等领域具有重要意义。曲线曲面的光顺性直接影响产品的质量、物理性能和美观性。因此,曲线曲面的光顺处理是计算机辅助几何设计中的一个研究热点,具有重要的理论与实用价值。国际上关于这一问题的研究可以追溯到20世纪60年代左右。传统的曲线曲面光顺方法主要包括基于能量最小化的全局光顺方法和修改选定“坏点”的局部光顺方法。尽管现在已存在许多曲线曲面光顺算法,但仍存在自动化水平低、计算复杂度高、效率低等问题。Fairing-PIA是一种通过调整控制顶点生成一系列光顺曲线曲面的几何迭代方法。Fairing-PIA算法赋予每个控制顶点单独权重来优化曲线与曲面的形状,为数据拟合生成光顺曲线曲面带来了更大的灵活性。本文设计了一种局部Fairing-PIA格式,称为Local-Fairing-PIA。在局部Fairing-PIA中,仅调整部分控制顶点及相应的光顺权重来优化局部曲线曲面形状。曲线曲面的光顺性可以通过局部Fairing-PIA被逐点或逐段进行调整。曲线和曲面不同部分的光顺程度可以不同。这种局部光顺的方法可以根据用户需求交互式地调整部分区域,既能达到局部光顺的效果,又能保留其余部分原本的特征。相较于传统的基于能量泛函极小的光顺方法,局部Fairing-PIA算法能够兼顾光顺效果与拟合误差对曲线曲面进行局部调整,同时避免了大规模矩阵运算,降低了计算成本。在本文中,我们证明了Local-Fairing-PIA迭代格式的收敛性。在多个曲线曲面拟合上的实验结果表明,局部Fairing-PIA是有效的。通过Local-Fairing-PIA算法,可将曲线曲面局部能量减少11%以上。Local-Fairing-PIA算法为曲线和曲面光顺问题提供了一种更为鲁棒、高效且灵活的解决方案。这种方法通过调整特定控制点和它们的权重,增强了曲线和曲面的平滑度和拟合精度,有助于提高产品在工程应用中的性能并减少计算开销。

关键词 渐进迭代逼近;光顺;局部;数据拟合;几何迭代方法

中图法分类号 TP391 **DOI号** 10.11897/SP.J.1016.2025.01327

Local Progressive and Iterative Approximation for Fairing Curve and Surface Generation

LIN Jia¹⁾ LIN Hong-Wei^{1,2)}

¹⁾(Department of Mathematical Sciences, Zhejiang University, Hangzhou 310058)

²⁾(State Key Laboratory of CAD&CG, Zhejiang University, Hangzhou 310058)

Abstract The fairing of curves and surfaces originates from the actual needs of engineering design and manufacturing, and is of great significance in various fields, such as automotive body design, aircraft wing design, and ship hull design. The fairness of these curves and surfaces directly influences the quality, physical performance, and aesthetics of the products. Consequently, the fairness of curves and surfaces is a prominent research focus on computer-aided geometric design, possessing substantial theoretical and practical value. International research on this topic began around the 1960s. Traditional methods for curve and surface fairing primarily include global fairing techniques based on minimizing energy functionals and local fairing methods that modify

收稿日期:2024-07-10;在线发布日期:2025-03-21。本课题得到国家自然科学基金(No. 62272406, 61932018)资助。林 佳,硕士研究生,主要研究领域为计算机辅助几何设计。E-mail: linjia@zju.edu.cn。蔺 宏伟(通信作者),博士,教授,中国计算机学会(CCF)会员,主要研究领域为计算机辅助几何设计、计算机辅助拓扑设计、拓扑数据分析、量子图形学。E-mail: hwlin@zju.edu.cn。

selected “bad points”. Despite the many existing methods for fairing curves and surfaces, several challenges remain, such as low levels of automation, high computational complexity, and inefficient algorithms. Fairing-PIA (Fairing-Progressive Iterative Approximation) is a geometric iterative method that generates a series of fairing curves and surfaces by adjusting control vertices. The Fairing-PIA algorithm assigns individual weights to each control vertex to optimize the shape of the curves and surfaces, providing greater flexibility for generating smooth curves and surfaces in data fitting. In this paper, we design a local Fairing-PIA scheme, named Local-Fairing-PIA. In Local-Fairing-PIA, only a subset of control vertices and their corresponding fairing weights are adjusted to optimize the local shape of the curves and surfaces. The fairing of curves and surfaces can be adjusted point by point or segment by segment using Local-Fairing-PIA, allowing different parts of a curve or surface to exhibit varying degrees of smoothness. This local fairing method can interactively adjust part of the region according to the user’s requirements, which can not only achieve the local fairing effect, but also retain the original characteristics of the rest of the region. Compared to traditional fairing methods based on minimizing energy functionals, the Local-Fairing-PIA algorithm balances fairing effects and fitting errors in local adjustments of curves and surfaces, while avoiding large-scale matrix computations and reducing computational costs. In this research, we establish the convergence of the Local-Fairing-PIA iterative format. Experimental results across multiple curve and surface fitting instances indicate that the Local-Fairing-PIA method is effective. Employing the Local-Fairing-PIA algorithm, it is possible to reduce the local energy of curves and surfaces by more than 11%. Overall, the Local-Fairing-PIA technique offers a robust, efficient, and flexible solution to the problem of curve and surface fairing. By fine-tuning specific control points and their weights, this method enhances the smoothness and fitting accuracy of curves and surfaces, ultimately contributing to improved performance and reduced computational overhead in engineering applications.

Keywords progressive iterative approximation; fairing; local; data fitting; iterative geometric method

1 引 言

曲线曲面光顺处理一直是计算机辅助几何设计中的研究热点。用光顺的曲线曲面拟合给定的离散点，这在船舶曲线曲面造型^[1]、车身曲线曲面造型^[2]、飞机机翼曲线曲面造型^[3]等多领域有重要应用。光顺性对产品的质量、物理性能、美观性等都直接产生重要影响。国际上对该问题的研究大约开始于上世纪 60 年代。传统的曲线曲面光顺方法主要是基于能量泛函极小的全局光顺方法与修改不符合光顺准则的型值点的局部光顺方法。对于曲线曲面的光顺处理，虽然已有了许多方法，但仍存在自动化程度低、计算量大、算法效率不高等问题。

光顺渐进迭代逼近(Progressive-Iterative Approximation for Fairing Curve and Surface Generation, Fairing-PIA)^[4]是一种高效的几何迭代方法，它兼

顾对数据点的拟合与曲线曲面的光顺性，在每一次迭代中不断调整控制点，最后得到光顺的目标曲线或曲面。Fairing-PIA 方法赋予每个控制顶点独立的权重，这些权重可以调整曲线曲面，从而提高曲线曲面光顺性。Fairing-PIA 虽然具有生成和调整光顺曲线曲面的能力，但这种全局调整的方法计算量较大。即使只有部分曲线曲面没有达到要求，在每次迭代中仍需要调整曲线曲面的所有控制顶点。在实际光顺问题中，往往只有曲线曲面的部分区域没有达到光顺要求，因此，发展局部光顺算法具有重要意义。本文提出了一个局部的 Fairing-PIA 算法，称为局部的曲线曲面光顺渐进迭代逼近(Local Progressive and Iterative Approximation for Fairing Curve and Surface Generation, Local-Fairing-PIA)。在局部格式中，我们可以仅调整部分控制顶点，在兼顾拟合精度的基础上通过这些控制顶点的权重调整曲线曲面的局部形状，使其具有良好的光顺性。

局部 Fairing-PIA 使得数据拟合生成光顺曲线曲面更具灵活性:(1)曲线(曲面)的光顺性可以被逐点或逐段调整;(2)曲线(曲面)不同部分的光顺程度可以不同。

本文结构如下:第2节介绍国内外相关工作;第3节给出曲线局部 Fairing-PIA 算法,并对其详细描述;第4节从理论上分析 Fairing-PIA 算法的收敛性;第5节给出曲面局部 Fairing-PIA 算法;第6节通过数值实验验证局部 Fairing-PIA 算法的有效性,并详细分析实验结果;第7节总结全文并对未来的研究工作进行展望。

2 相关工作

2.1 曲线曲面光顺

曲线曲面光顺性是一个难以量化的概念。在实际工程中,设计者通常通过观察曲率分布来评估曲线光顺性。如果曲线的曲率变化均匀,则认为该曲线光顺^[5]。由于光顺性涉及几何外形美观性,受主观因素影响大,同时不同的实际问题中,对于光顺性的要求也不同,所以很难用统一的数量指标量化曲线光顺性^[6]。对于曲线 $C(t)$, 我们通常使用光顺模型

$$\min_{C(t)} \int \kappa^2(s) ds,$$

其中 $\kappa(s)$ 为曲线曲率, s 为曲线弧长。对于曲面 $S(u, v)$, 光顺模型为

$$\min_{S(u, v)} \int \kappa_1^2 + \kappa_2^2 d\sigma,$$

其中, κ_1, κ_2 为曲面主曲率, σ 为曲面面积。

但是在实际应用中,解上述能量模型会导致一个高度非线性问题,计算量非常大。因此,我们通常使用以下模型^[7]来近似曲线的光顺模型,

$$\min_{C(t)} \int \|C^{(r)}(t)\|^2 dt,$$

其中 $r=1, 2, 3$ 。

类似于曲线情形,我们通常用以下模型来近似曲面 $S(u, v)$ 的光顺模型^[8]:

$$\begin{aligned} \min_{S(u, v)} & \iint \|S_{uu}(u, v)\|^2 + 2\|S_{uv}(u, v)\|^2 + \\ & \|S_{vv}(u, v)\|^2 du dv. \end{aligned}$$

在传统的曲线曲面光顺方法中,常用能量最小化问题来解决光顺问题。例如, Hosaka^[9]以曲线曲面调整前后的偏离与光顺程度(样条的能量积分)的加权平均为目标函数,基于能量极值原理进行光

顺; Lott 和 Pullin^[10]以曲面曲率为目标函数,将原始曲面和修改曲面之间距离作为约束,通过约束最小化算法自动重新定位控制点,得到光顺曲线曲面; Greiner 等^[11]基于参数化曲面的简化薄板能量泛函进行光顺曲面逼近。这些方法可以对曲线曲面进行全局优化,但需要解一个大规模的线性方程,拟合效果也不能很好地控制。选点法是一种局部光顺方法,通过逐次找到不符合光顺准则的型值点(称为“坏点”),并对坏点进行修改,从而得到光顺的曲线曲面。Farin 等^[12]依据给定的光顺准则定义曲线上的坏点,通过删改该“坏点”的节点提高曲线在该点的光顺性。Stefanie^[13]等将曲线节点去除算法用于曲面,结合最佳搜索和模拟退火两种搜索算法,用最小区域算法实现曲面的局部光顺。Zhang 等^[14]将能量最小化方法与选点法相结合,一次性修改所有对“坏点”有影响的控制顶点。Vassilev^[15]开发了一种基于能量最小化和数据点插入的自适应光顺技术。该技术只在必要时添加新的数据点,能够大大减少需要求解的控制点数量。此外,小波光顺算法^[16-18]、升阶光顺算法^[19-20]等也是曲线曲面光顺问题的重要解决办法。

近年来,曲线曲面光顺性研究也得到了进一步发展。Wang 等^[21]提出了一种基于稀疏模型的曲线光顺算法,并对现有评判准则进行了总结。Yang 等^[22]提出了一种利用矩阵加权 NURBS 曲线拟合和光顺数据点序列以及数据点处的法线或切线的技术。Yu 等^[23]进一步完善了基于小波技术的 NURBS 曲线光顺算法,实现了任意控制顶点的 NURBS 曲线多分辨光顺。Jiang 等^[24]通过迭代的节点插入算法构造全局光滑的 B 样条曲面,构造的曲面插值于数据点且具有较少的控制顶点。这些算法大多关注曲线曲面整体的光顺性,在光顺的过程中会改变曲线曲面的整体形状。在本文中,我们提出了一个局部光顺算法,能够兼顾光顺效果与拟合误差对曲线曲面进行局部调整,使得曲线曲面不同部分的光顺程度可以不同,具有重要的应用价值。

2.2 演进迭代逼近

演进迭代逼近(Progressive-Iterative Approximation, PIA)是 Lin 等^[25-26]提出并命名的一种十分有效的数据拟合方法。演进迭代逼近通过不断调整控制顶点,生成一组极限插值或逼近给定数据点列的混合曲线或曲面序列。受 Qi 等提出的均匀 3 次 B 样条盈亏修正算法^[27-29]和 de Boor 等的相关研究^[30]启发, Lin 等^[25]首先证明了非均匀 3 次 B 样

条曲线曲面迭代算法的收敛性，随后又证明了全正基混合曲线曲面都具有这一性质^[26]，并将该方法命名为渐进迭代逼近。

相较于传统的数据拟合方法，渐进迭代逼近算法简单，几何意义明确，同时避免了大规模线性方程求解^[31]，越来越成为几何设计领域重要的数据拟合算法^[32]。近年来，渐进迭代逼近算法得到了进一步的发展。Lu^[33]提出了带权值的PIA算法(Weighted Progressive Iteration Approximation, WPIA)进一步提升了算法速度。Lin^[34]提出了局部PIA(Local Progressive Iterative Approximation, LPIA)，该算法仅调整部分控制点，数据点可以自适应地拟合，算法更加灵活。随后Lin等^[35]提出一种PIA的推广形式(Extended Progressive Iterative Approximation, EPIA)，该方法可以进行并行计算并允许控制点数小于数据点个数。Deng与Lin^[36]提出一种最小二乘渐进迭代逼近(Least Square Progressive Iterative Approximation, LSPIA)，能够高效稳定地进行大规模数据拟合并显著减少计算量。Hamza等^[37]提出了GS-LSPIA，进一步加快了LSPIA的收敛速度。Liu等^[38]提出了一种基于径向基函数的共轭梯度隐式渐进迭代逼近(Conjugate Gradient Implicit Progressive Iterative Approximation, CG-I-PIA)，该算法能够以近似或插值的方式很好地重构曲线、曲面甚至高维数据。Wang等^[39]提出了一种隐式随机渐进迭代逼近(Implicit Randomized Progressive Iterative Approximation, IR-PIA)，进一步加快了曲线曲面重建速度。Liu等^[40]提出了一种双三次B样条曲面的Jacobi渐进迭代逼近算法，基于配置矩阵的Jacobi分裂，进一步提升了算法收敛速度。Jiang等^[41]提出了一种基于渐进迭代逼近的曲线曲面光顺方法，即Fairing-PIA。该方法通过调整控制点，生成一系列光顺的曲线曲面。同时证明了传统的能量优化光顺模型就是Fairing-PIA的一种特殊情况^[4]。

传统的曲线曲面光顺算法通常通过计算能量泛函极小得到光顺曲线曲面，在光顺的过程中会改变曲线曲面整体形状，是一种全局的方法。同时传统的光顺算法往往伴随着大型方程组求解，计算成本高，算法运行时间长。Fairing-PIA算法虽然规避了矩阵计算，同时赋予每个控制顶点单独的光顺权重，使得算法更具有灵活性，但仍是一种全局的光顺方法。在实际应用中，曲线曲面不同部分的拟合精度与光顺性要求往往是不同的。我们在光顺曲线

曲面的同时，可能仍希望保留部分区域的尖锐特征。局部Fairing-PIA是一种局部光顺的方法，它能让用户交互式地调整曲线曲面局部区域，在兼顾拟合效果的同时快速地提升局部光顺性，同时保持其余区域不变。局部Fairing-PIA大大降低了曲线曲面光顺的运算成本，同时更贴近实际需求，具有重要的应用价值。

3 曲线的局部Fairing-PIA算法

Fairing-PIA通过调整控制顶点，生成一系列光顺曲线曲面。在每次迭代中，Fairing-PIA构建各个控制顶点对应的拟合向量与光顺向量。控制顶点的差向量为两向量的加权和。当前控制顶点加上差向量即为新的控制顶点。相较于基于能量泛函极小的传统光顺方法，Fairing-PIA避免了矩阵计算，减少了运算成本且更加灵活。但Fairing-PIA仍是一种全局光顺方法，在每次迭代中需要调整所有的控制顶点，曲线曲面的整体形状都会发生变化。

在本文中，我们将给出一个更灵活的迭代格式，也即局部Fairing-PIA格式。在实际应用中，可能仅有部分的曲线曲面需要光顺，同时希望保留其余部分原有的特征。局部Fairing-PIA通过调整局部控制顶点，对曲线曲面的形状进行局部调整。这种局部光顺的方法可以根据用户需求交互式地调整部分区域，既能达到局部光顺的效果，又能保留其余部分原本的特征。同时，在每次的迭代中，我们仅需要调整部分控制顶点，这将大大减少计算成本，降低算法时间。

事实上，Fairing-PIA对于包括Bézier曲线曲面、NURBS曲线曲面、T样条曲线曲面在内的混合曲线曲面都是有效的^[4]。局部Fairing-PIA格式也是如此。在本文中，我们用B样条曲线曲面来展示局部Fairing-PIA格式。

若给定数据点 $\{Q_i\}_{i=1}^m$ 以及对应参数 $\{t_i\}_{i=1}^m$ ，参数满足 $t_1 \leq t_2 \leq \dots \leq t_m$ 。从给定的 m 个数据点中选取 n 个点作为控制顶点 $\{P_j^{[0]}\}_{j=1}^n$ 。假设这是一条次数为 p 的B样条曲线，其节点向量为 $\{\hat{t}_1, \hat{t}_2, \dots, \hat{t}_{n+p+1}\}$ 。

初始的B样条曲线如下：

$$P^{[0]}(t) = \sum_{j=1}^n N_j(t) P_j^{[0]}, \quad t \in [t_1, t_m] \quad (1)$$

其中 $N_j(t)$ 为第 j 个控制顶点对应的B样条基函数。

在局部 Fairing-PIA 中, 仅曲线的部分区域需要调整, 这部分对应的数据点为 $\{Q_{q_1}, Q_{q_2}, \dots, Q_{q_Q}\}$, 记下标集合为 $I_{fit} = \{q_1, q_2, \dots, q_Q\}$ 。在这些区域中, 非零基函数对应的控制顶点, 称为活动控制顶点, 记为 $\{P_{i_1}, P_{i_2}, \dots, P_{i_l}\}$, 记下标集合为 $J_{act} = \{i_1, i_2, \dots, i_l\}$ 。剩余的控制顶点保持不动, 称为固定控制顶点, 记为 $\{P_{j_1}, P_{j_2}, \dots, P_{j_s}\}$, 记下标集合为 $J_{fix} = \{j_1, j_2, \dots, j_s\}$ 。

按照如下形式定义光顺泛函:

$$\mathcal{F}_{r,j}(f) = \int_{t_1}^{t_m} N_{r,j}(t) f dt, \quad (2)$$

$$j=1, 2, \dots, n, r=1, 2, 3$$

其中 $N_{r,j}$ 表示 $N_j(t)$ 的 r 阶导, f 表示任意函数。

基于 B 样条基函数的局部支撑性, 对于对应参数落在第 j 个基函数支持域的数据点 Q_i , 也即 $N_j(t_i) \neq 0$, 将被分到第 j 组, 对应第 j 个控制顶点。第 j 组中数据点的下标集记为 I_j 。

下面给出局部 Fairing-PIA 算法格式。在第一次迭代中, 我们首先计算数据点的差向量。不同于全局 Fairing-PIA, 这里只需要计算待拟合数据点的差向量:

$$\mathbf{d}_i^{[0]} = Q_i - P^{[0]}(t_i), i \in I_{fit}.$$

再计算活动控制顶点的拟合向量

$$\mathbf{d}_h^{[0]} = \sum_{l \in I_h \cap I_{fit}} N_h(t_l) \mathbf{d}_l^{[0]}, h \in J_{act}$$

以及光顺向量

$$\boldsymbol{\eta}_h^{[0]} = \sum_{l=1}^n \mathcal{F}_{r,l}(N_{r,h}(t)) P_l^{[0]}, h \in J_{act}.$$

由此可得新的控制顶点为

$$P_h^{[1]} = P_h^{[0]} + \mu_h [(1 - \omega_h) \delta_h^{[0]} - \omega_h \boldsymbol{\eta}_h^{[0]}],$$

$$h \in J_{act}, P_h^{[1]} = P_h, h \in J_{fix},$$

其中 μ_h 为标准化因子, 用于保证算法的收敛性, ω_h 为光顺权重, 用于平衡拟合向量与光顺向量。

因此第一次迭代后的 B 样条曲线为:

$$P^{[1]}(t) = \sum_{h=1}^n N_h(t) P_h^{[1]} = \sum_{h \in J_{act}} N_h(t) P_h^{[1]} + \sum_{h \in J_{fix}} N_h(t) P_h, t \in [t_1, t_m].$$

假设在第 k 次迭代后得到了曲线 $P^{[k]}(t)$, 为了计算得到新一轮控制顶点坐标, 我们先计算数据点的差向量:

$$\mathbf{d}_i^{[k]} = Q_i - P^{[k]}(t_i), i \in I_{fit}.$$

再计算活动控制顶点的拟合向量与光顺向量

$$\mathbf{d}_h^{[k]} = \sum_{l \in I_h \cap I_{fit}} N_h(t_l) \mathbf{d}_l^{[k]}, h \in J_{act}, \quad (3)$$

$$\boldsymbol{\eta}_h^{[k]} = \sum_{l=1}^n \mathcal{F}_{r,l}(N_{r,h}(t)) P_l^{[k]}, h \in J_{act}. \quad (4)$$

因此, 新的控制顶点为

$$P_h^{[k+1]} = P_h^{[k]} + \mu_h [(1 - \omega_h) \delta_h^{[k]} - \omega_h \boldsymbol{\eta}_h^{[k]}],$$

$$h \in J_{act}, P_h^{[k+1]} = P_h, h \in J_{fix}. \quad (5)$$

那么, 第 $k+1$ 次迭代后的 B 样条曲线为:

$$P^{[k+1]}(t) = \sum_{h=1}^n N_h(t) P_h^{[k+1]} = \sum_{h \in J_{act}} N_h(t) P_h^{[k+1]} + \sum_{h \in J_{fix}} N_h(t) P_h, t \in [t_1, t_m].$$

通过以上方法, 我们得到了一组曲线: $\{P^{[k]}(t), k=0, 1, 2, \dots\}$ 。在下一节中, 我们将证明这组曲线是收敛的。

曲线局部 Fairing-PIA 算法伪代码如算法 1 所示:

算法 1. 曲线局部 Fairing-PIA 算法

输入: 有序点列 $\{Q_i\}_{i=1}^m$, 调整待拟合数据点下标集合 I_{fit} , 相邻两次迭代拟合误差之差的绝对值上限 ϵ , 最大迭代次数 K , 曲线控制顶点个数 n

输出: 拟合曲线 $P^{[k]}(t)$

1. 对数据点进行参数化

2. 选择初始控制顶点构建初始曲线 $P^{[0]}(t) = \sum_{j=1}^n N_j(t) P_j^{[0]}$, 确定活动控制顶点下标集合 J_{act} 与固定控制顶点下标集合 J_{fix}

3. $k=0$

4. WHILE($k < K$)

5. 计算给定调整数据点差向量

$$\mathbf{d}_i^{[k]} = Q_i - P^{[k]}(t_i), i \in I_{fit}$$

6. 计算活动控制顶点拟合向量

$$\mathbf{d}_h^{[k]} = \sum_{l \in I_h \cap I_{fit}} N_h(t_l) \mathbf{d}_l^{[k]}, h \in J_{act}$$

7. 计算活动控制顶点光顺向量

$$\boldsymbol{\eta}_h^{[k]} = \sum_{l=1}^n \mathcal{F}_{r,l}(N_{r,h}(t)) P_l^{[k]}, h \in J_{act}$$

8. 计算第 $k+1$ 次迭代曲线的新控制顶点

$$P_h^{[k+1]} = P_h^{[k]} + \mu_h [(1 - \omega_h) \delta_h^{[k]} - \omega_h \boldsymbol{\eta}_h^{[k]}], h \in J_{act}, P_h^{[k+1]} = P_h, h \in J_{fix}$$

9. 计算第 $k+1$ 次迭代曲线

$$P^{[k+1]}(t) = \sum_{h=1}^n N_h(t) P_h^{[k+1]}$$

10. 计算局部拟合误差

$$E_{fit}^{[k+1]} = \max_{i \in I_{fit}} \|Q_i - P^{[k+1]}(t_i)\|$$

11. IF $|E_{fit}^{[k+1]} - E_{fit}^{[k]}| < \epsilon$

RETURN $P^{[k+1]}(t)$

12. $k=k+1$

13. RETURN $P^{[k]}(t)$

4 收敛性分析

在这一节，我们将进行曲线局部 Fairing-PIA 算法的收敛性分析。同时，证明局部能量光顺算法是局部 Fairing-PIA 的一种特殊形式。

为了证明局部 Fairing-PIA 的收敛性，我们先将其写成矩阵形式。

将控制顶点重新排序如下：

$$\begin{aligned} P_{act}^{[k]} &= [P_{i_1}^{[k]}, \dots, P_{i_l}^{[k]}]^T, \\ P_{fix} &= [P_{j_1}, \dots, P_{j_r}]^T, \\ P^{[k]} &= [P_{act}^{[k]}, P_{fix}]^T, \end{aligned}$$

其中 $P_{act}^{[k]}$ 为第 k 次迭代中的活动控制顶点， P_{fix} 为固定控制顶点， $P^{[k]}$ 为第 k 次迭代中的所有控制顶点。

因此曲线局部 fairing-PIA 中的迭代格式可被写成如下矩阵形式：

$$P_{act}^{[k+1]} = (I - \Lambda A_1) P_{act}^{[k]} - \Lambda A_2 P_{fix} + \Lambda B Q \quad (6)$$

其中

$$\begin{aligned} Q &= [Q_1, Q_2, \dots, Q_{m-1}, Q_m]^T, \\ A_1 &= (I - \Omega) D_1^T D_1 + \Omega F_1, \\ A_2 &= (I - \Omega) D_1^T D_2 + \Omega F_2, \\ B &= [(I - \Omega) D_1^T \quad (I - \Omega) D_2^T], \\ \Lambda &= diag(\mu_{i_1}, \mu_{i_2}, \dots, \mu_{i_l}), \\ \Omega &= diag(\omega_{i_1}, \omega_{i_2}, \dots, \omega_{i_l}), \end{aligned} \quad (7)$$

以及

$$\begin{aligned} D_1 &= \begin{bmatrix} N_{i_1}(t_1) & \cdots & N_{i_l}(t_1) \\ \vdots & \ddots & \vdots \\ N_{i_1}(t_m) & \cdots & N_{i_l}(t_m) \end{bmatrix}, \\ D_2 &= \begin{bmatrix} N_{j_1}(t_1) & \cdots & N_{j_r}(t_1) \\ \vdots & \ddots & \vdots \\ N_{j_1}(t_m) & \cdots & N_{j_r}(t_m) \end{bmatrix}, \\ F_I &= \begin{bmatrix} \mathcal{F}_{r,i_1}(N_{r,i_1}(t)) & \cdots & \mathcal{F}_{r,i_l}(N_{r,i_l}(t)) \\ \vdots & \ddots & \vdots \\ \mathcal{F}_{r,i_1}(N_{r,i_1}(t)) & \cdots & \mathcal{F}_{r,i_l}(N_{r,i_l}(t)) \end{bmatrix}, \\ F_J &= \begin{bmatrix} \mathcal{F}_{r,j_1}(N_{r,j_1}(t)) & \cdots & \mathcal{F}_{r,j_r}(N_{r,j_r}(t)) \\ \vdots & \ddots & \vdots \\ \mathcal{F}_{r,j_1}(N_{r,j_1}(t)) & \cdots & \mathcal{F}_{r,j_r}(N_{r,j_r}(t)) \end{bmatrix}. \end{aligned}$$

引理 1. 若 A_1 是严格对角占优矩阵，且 Λ 的对角元素满足 $0 < \mu_i < \frac{2}{\sum_{j=1}^l |a_{ij}|}$, $i = i_1, \dots, i_l$, (a_{ij} 为

A_1 中元素)，则有 $0 < \|I - \Lambda A_1\|_\infty < 1$ 。

证明。根据矩阵 A_1 定义，易知 A_1 的对角线元素 a_{ii} 恒正。类似文献[4]，可证得

(1) 若 $0 < \mu_i \leq \frac{1}{a_{ii}}$, 则有

$$\begin{aligned} \|I - \Lambda A_1\|_\infty &= \max_i (1 - \mu_i a_{ii} + \mu_i \sum_{j \neq i} |a_{ij}|) \\ &< \max_i (1 - \mu_i a_{ii} + \mu_i a_{ii}) = 1. \end{aligned}$$

(2) 若 $\frac{1}{a_{ii}} < \mu_i < \frac{2}{\sum_{j=1}^l |a_{ij}|}$, 则有

$$\begin{aligned} \|I - \Lambda A_1\|_\infty &= \max (\mu_i a_{ii} - 1 + \mu_i \sum_{j \neq i} |a_{ij}|) = \\ \max_i (\mu_i \sum_{j=1}^l |a_{ij}| - 1) &< 1. \end{aligned}$$

因此，当 $0 < \mu_i < \frac{2}{\sum_{j=1}^l |a_{ij}|}$ 时，有 $0 < \|I - \Lambda A_1\|_\infty < 1$ 。

定理 1. 若 A_1 是严格对角占优矩阵，则局部 Fairing-PIA 是收敛的。

证明。令 $C = -\Lambda A_2 P_{fix} + \Lambda B Q$ 。显然矩阵 C 为一个固定的常数矩阵，不会随着迭代的进行发生变化。

根据(6)式，我们可得

$$\begin{aligned} P_{act}^{[k+1]} &= (I - \Lambda A_1) P_{act}^{[k]} + C \\ &= (I - \Lambda A_1)((I - \Lambda A_1) P_{act}^{[k-1]} + C) + C \\ &= (I - \Lambda A_1)^2 P_{act}^{[k-1]} + (I - \Lambda A_1) C + C \\ &= \dots \\ &= (I - \Lambda A_1)^{k+1} P_{act}^{[0]} + \sum_{l=0}^k (I - \Lambda A_1)^l C \end{aligned} \quad (8)$$

基于引理 1, $I - \Lambda A_1$ 的谱半径满足

$$0 < \rho(I - \Lambda A_1) < \|I - \Lambda A_1\|_\infty < 1,$$

由此可得

$$\begin{aligned} \lim_{k \rightarrow \infty} (I - \Lambda A_1)^k &= O, \\ \lim_{k \rightarrow \infty} \sum_{l=0}^k (I - \Lambda A_1)^l C &= (\Lambda A_1)^{-1} C. \end{aligned}$$

因此，当 $k \rightarrow \infty$, (6)式趋于

$$\begin{aligned} P_{act}^{[\infty]} &= (\Lambda A_1)^{-1} C \\ &= (\Lambda A_1)^{-1} (-\Lambda A_2 P_{fix} + \Lambda B Q) \\ &= -A_1^{-1} A_2 P_{fix} + A_1^{-1} B Q. \end{aligned}$$

上式等价于如下线性方程组的解：

$$A_1 P_{act}^{[\infty]} = -A_2 P_{fix} + B Q. \quad (9)$$

在本文的数值实验中，为了保证算法的收敛性，

我们均取标准化因子 $\mu_i = \frac{1}{\sum_{j=1}^{i_1} |a_{ij}|}$ ($i = i_1, \dots, i_{I_1}$)。

在实际操作中, 若我们取光顺权重 $\omega_h = \omega$, $h = i_1, i_2, \dots, i_{I_1}$, 则局部 Fairing-PIA 有如下形式:

$$\bar{P}_{act}^{[k+1]} = (\mathbf{I} - \boldsymbol{\Lambda}\bar{\mathbf{A}}_1)\bar{P}_{act}^{[k]} - \boldsymbol{\Lambda}\bar{\mathbf{A}}_2 P_{fix} + (1 - \omega)\boldsymbol{\Lambda}\bar{\mathbf{B}}\mathbf{Q} \quad (10)$$

其中

$$\begin{aligned} \bar{\mathbf{A}}_1 &= (1 - \omega)D_1^T D_1 + \omega F_1, \\ \bar{\mathbf{A}}_2 &= (1 - \omega)D_1^T D_2 + \omega F_2, \\ \bar{\mathbf{B}} &= [D_1^T \ D_2^T]. \end{aligned}$$

易证, 在上述 $\Omega = \omega I$ 的特殊情况下, 局部 Fairing-PIA 收敛于如下情况:

$$\bar{\mathbf{A}}_1 \bar{P}_{act}^{[\infty]} = -\bar{\mathbf{A}}_2 P_{fix} + (1 - \omega)\bar{\mathbf{B}}\mathbf{Q}. \quad (11)$$

在传统光顺方法中, 对于不光顺的曲线段, 常采用能量法的方法对其进行光顺。其目标为找到如下能量泛函的极小值:

$$E = \frac{1 - \omega}{2} f_1 + \frac{\omega}{2} f_2, \quad (12)$$

其中 f_1 和 f_2 分别用于衡量局部曲线拟合效果与光顺效果, 定义如下

$$\begin{aligned} f_1 &= \sum_{i=q_1}^{q_Q} \|P(t_i) - Q_i\|^2, \\ f_2 &= \int_{t_{q_1}}^{t_{q_Q}} \|P^{(r)}(t)\|^2 dt \end{aligned}$$

在这里 $P^{(r)}(t)$ 表示曲线 $P(t)$ 的 r 阶导。

下面我们将证明当活动控制点光顺权重均为 ω 时, 局部能量泛函极小与局部 Fairing-PIA 算法是等价的, 也即局部能量泛函极小是局部 Fairing-PIA 的一种特殊形式。

将 f_1 与 f_2 的表达式代入能量函数(12), 上述问题等价于

$$\begin{aligned} \min_{P_h} E &= \min_{P_h} \left[\frac{1 - \omega}{2} \sum_{i=q_1}^{q_Q} \|P(t_i) - Q_i\|^2 + \right. \\ &\quad \left. \frac{\omega}{2} \int_{t_{q_1}}^{t_{q_Q}} \|P^{(r)}(t)\|^2 dt \right]. \end{aligned}$$

为解上述能量优化问题, 我们令 $\frac{\partial E}{\partial P_h} = 0$, 得

$$\begin{aligned} i_1, i_2, \dots, i_{I_1}, \text{得} \\ \frac{\partial E}{\partial P_h} &= (1 - \omega) \sum_{i=q_1}^{q_Q} (P(t_i) - Q_i) N_h(t_i) + \\ &\quad \omega \int_{t_{q_1}}^{t_{q_Q}} P^{(r)}(t) N_h^{(r)}(t) dt = 0. \end{aligned}$$

整理得

$$(1 - \omega) \sum_{i=1}^m N_h(t_i) Q_i = (1 - \omega) \sum_{i=q_1}^{q_Q} \sum_{j=i_1}^{i_{I_1}} N_h(t_i) N_j(t_i) P_j + \omega \int_{t_{q_1}}^{t_{q_Q}} \left(\sum_{j=i_1}^{i_{I_1}} N_j^{(r)}(t) N_h^{(r)}(t) P_j \right) dt,$$

这也意味着

$$\begin{aligned} \left[(1 - \omega) \sum_{i=q_1}^{q_Q} N_h^2(t_i) + \omega \int_{t_{q_1}}^{t_{q_Q}} (N_h^{(r)})^2 dt \right] P_h = \\ (1 - \omega) \sum_{i=q_1}^{q_Q} N_h(t_i) Q_i - (1 - \omega) \sum_{i=q_1}^{q_Q} \sum_{j=i_1, j \neq h}^{i_{I_1}} N_h(t_i) N_j(t_i) P_j - \\ \omega \int_{t_{q_1}}^{t_{q_Q}} \left(\sum_{j=i_1, j \neq h}^{i_{I_1}} N_j^{(r)}(t) N_h^{(r)}(t) P_j \right) dt. \end{aligned}$$

上述结果等价于矩阵

$$\bar{\mathbf{A}}_1 \bar{P}_1 = -\bar{\mathbf{A}}_2 \bar{P}_2 + (1 - \omega) \bar{\mathbf{B}}\mathbf{Q}$$

其中,

$$\bar{P}_1 = [P_{i_1}, \dots, P_{i_{I_1}}]^T,$$

$$\bar{P}_2 = [P_{j_1}, \dots, P_{j_{I_2}}]^T.$$

根据(11)式, 局部能量泛函极小的结果与局部 Fairing-PIA 在特殊情况下的收敛结果是一致的。

类似于文献^[4], $r = 1, 2, 3$ 时分别对应拉伸能量^[7]、弯曲能量^[41]和扭曲能量^[42]。在后续数值实验中, 我们将比较 r 的不同取值对于局部光顺性的影响。

5 曲面的局部 Fairing-PIA 算法

曲线的局部光顺渐进迭代逼近可以推广到曲面情况。在这一节, 我们将介绍曲面情况下的局部 Fairing-PIA 细节。

给定一列有序的数据点 $\{Q_{ij}\}_{i=1, j=1}^{m_1, m_2}$, 以及它们对应的参数列 $\{s_i, t_j\}_{i=1, j=1}^{m_1, m_2}$, 满足 $s_1 \leq s_2 \leq \dots \leq s_{m_1}$ 和 $t_1 \leq t_2 \leq \dots \leq t_{m_2}$ 。初始的 B 样条曲面 $P^{[0]}(s, t)$ 可以被表示如下:

$$P^{[0]}(s, t) = \sum_{h=1}^{n_1} \sum_{l=1}^{n_2} N_h(s) N_l(t) P_{hl}^{[0]}, \quad 0 \leq s, t \leq 1, \quad (13)$$

其中 $\{P_{hl}\}_{h=1, l=1}^{n_1, n_2}$ 是初始控制顶点, $N_h(s)$ 与 $N_l(t)$ 为 B 样条基函数。我们将数据点以及对应参数按照字典序重新排列如下:

$$\begin{aligned} [Q_1, Q_2, \dots, Q_{m_1 m_2}] &= [Q_{11}, Q_{12}, \dots, Q_{m_1, m_2}], \\ [(s_1, t_1), (s_2, t_2), \dots, (s_{m_1 m_2}, t_{m_1 m_2})] &= \\ [(s_1, t_1), (s_1, t_2), \dots, (s_{m_1}, t_{m_2})]. \end{aligned}$$

按照同样的方式，重新排列B样条基函数以及控制顶点如下：

$$\begin{aligned} [N_1(s, t), N_2(s, t), \dots, N_{n_1 n_2}(s, t)] &= \\ [N_1(s)N_1(t), N_1(s)N_2(t), \dots, N_{n_1}(s)N_{n_2}(t)], \\ [P_1, P_2, \dots, P_{n_1 n_2}] &= [P_{11}, P_{12}, \dots, P_{n_1, n_2}]. \end{aligned}$$

那么初始曲面可被重写为

$$P^{[0]}(s, t) = \sum_{j=1}^{n_1 n_2} N_j(s, t) P_j^{[0]}. \quad (14)$$

定义曲面光顺泛函如下：

$$\begin{aligned} \mathcal{F}_{ss,j}(f) &= \int_{s_1}^{s_m} \int_{t_1}^{t_m} N_{ss,j}(s, t) f ds dt, \\ \mathcal{F}_{st,j}(f) &= \int_{s_1}^{s_m} \int_{t_1}^{t_m} N_{st,j}(s, t) f ds dt, \\ \mathcal{F}_{tt,j}(f) &= \int_{s_1}^{s_m} \int_{t_1}^{t_m} N_{tt,j}(s, t) f ds dt, \\ j &= 1, 2, \dots, n_1 n_2, \end{aligned} \quad (15)$$

其中 $N_{ss,j}(s, t)$, $N_{st,j}(s, t)$ 和 $N_{tt,j}(s, t)$ 分别代表 $N_j(s, t)$ 关于 s 和 t 的二阶偏导, f 为任意函数。

假设只有部分曲面需要调整, 对应的数据点为 $\{Q_{q_1}, Q_{q_2}, \dots, Q_{q_Q}\}$, 记下标集合为 $I_{fit} = \{q_1, q_2, \dots, q_Q\}$ 。这部分曲面参数域中非零的基函数对应的控制顶点称为活动控制顶点, 记为 $\{P_{i_1}, P_{i_2}, \dots, P_{i_l}\}$, 记下标集合为 $J_{act} = \{i_1, i_2, \dots, i_l\}$ 。其余控制顶点保持不动, 称为固定控制顶点, 记为 $\{P_{j_1}, P_{j_2}, \dots, P_{j_s}\}$, 记下标集合为 $J_{fix} = \{j_1, j_2, \dots, j_s\}$ 。

类似于曲线的情况, 在第一次迭代中, 我们先计算数据点 $\{Q_{q_1}, Q_{q_2}, \dots, Q_{q_Q}\}$ 的差向量:

$$\delta_i^{[0]} = Q_i - P^{[0]}(s_i, t_i), i \in I_{fit},$$

再计算活动控制点的拟合向量 $\delta_h^{[0]}$ 与光顺向量 $\eta_h^{[0]}$:

$$\delta_h^{[0]} = \sum_{l \in \hat{I}_h \cap I_{fit}} N_h(s_l, t_l) \delta_l^{[0]}, h \in J_{act}$$

$$\begin{aligned} \eta_h^{[0]} &= \sum_{l=1}^{n_1 n_2} [\mathcal{F}_{ss,l}(N_{ss,h}(s, t)) + 2\mathcal{F}_{st,l}(N_{st,h}(s, t)) + \\ &\quad \mathcal{F}_{tt,l}(N_{tt,h}(s, t))] P_l^{[0]}, h \in J_{act}. \end{aligned}$$

其中 \hat{I}_h 为所有落在第 h 个基函数支撑域的参数的指标集。

因此, 第一次迭代曲面的新控制顶点为

$$P_h^{[1]} = P_h^{[0]} + \mu_h[(1 - \omega_h)\delta_h^{[0]} - \omega_h\eta_h^{[0]}], h \in J_{act},$$

$$P_h^{[1]} = P_h^{[0]}, h \in J_{fix}.$$

类似于曲线情形, 这里的 μ_h 为标准化因子, 用

于保证算法的收敛性。 ω_h 为光顺权重, 用于平衡拟合向量与光顺向量。

这使得第一次迭代曲面的表达式为:

$$\begin{aligned} P^{[1]}(s, t) &= \sum_{h=1}^{n_1 n_2} N_h(s, t) P_h^{[1]} = \\ \sum_{h \in J_{act}} N_h(s, t) P_h^{[1]} + \sum_{h \in J_{fix}} N_h(s, t) P_h. \end{aligned}$$

假设在第 k 次迭代后得到了第 k 个曲面 $P^{[k]}(s, t)$, 那么在第 $(k+1)$ 次迭代中, 我们先计算数据点的差向量:

$$\delta_i^{[k]} = Q_i - P^{[k]}(s_i, t_i), i \in I_{fit},$$

再计算活动控制点的拟合向量与光顺向量:

$$\begin{aligned} \delta_h^{[k]} &= \sum_{l \in \hat{I}_h \cap I_{fit}} N_h(s_l, t_l) \delta_l^{[k]}, h \in J_{act}, \\ \eta_h^{[k]} &= \sum_{l=1}^{n_1 n_2} [\mathcal{F}_{ss,l}(N_{ss,h}(s, t)) + 2\mathcal{F}_{st,l}(N_{st,h}(s, t)) + \\ &\quad \mathcal{F}_{tt,l}(N_{tt,h}(s, t))] P_l^{[k]}, h \in J_{act}. \end{aligned}$$

由此可得第 $(k+1)$ 个曲面的新控制顶点为:

$$\begin{aligned} P_h^{[k+1]} &= P_h^{[k]} + \mu_h[(1 - \omega_h)\delta_h^{[k]} - \omega_h\eta_h^{[k]}], h \in J_{act}, \\ P_h^{[k+1]} &= P_h^{[k]}, h \in J_{fix}. \end{aligned} \quad (16)$$

这使得第 $(k+1)$ 个曲面的表达式为:

$$\begin{aligned} P^{[k+1]}(s, t) &= \sum_{h=1}^{n_1 n_2} N_h(s, t) P_h^{[k+1]} = \\ \sum_{h \in J_{act}} N_h(s, t) P_h^{[k+1]} + \sum_{h \in J_{fix}} N_h(s, t) P_h. \end{aligned}$$

通过如上方法, 我们得到了一系列曲面 $\{P^{[k]}(s, t), k=1, 2, \dots\}$, 该曲面序列的收敛性分析类似于曲线序列。

曲面局部 Fairing-PIA 算法伪代码如算法 2 所示:

算法 2. 曲面局部 Fairing-PIA 算法

输入: 有序点列 $\{Q_{ij}\}_{i=1, j=1}^{m_1, m_2}$, 调整待拟合数据点下标集合 I_{fit} , 相邻两次迭代拟合误差之差的绝对值上限 ϵ , 最大迭代次数 K , 曲面 s 方向与 t 方向控制顶点个数 n_1, n_2
输出: 拟合曲面 $P^{[k]}(s, t)$

1. 对数据点进行参数化
2. 选择初始控制顶点构建初始曲线, 按字典序重新排列数据点、基函数、控制顶点, 得到初始曲面

$$P^{[0]}(s, t) = \sum_{j=1}^{n_1 n_2} N_j(s, t) P_j^{[0]}, \text{确定活动控制顶点下标集}$$

合 J_{act} 与固定控制顶点下标集合 J_{fix}

3. $k = 0$

4. WHILE($k < K$)

5. 计算给定调整数据点差向量

$$\delta_i^{[k]} = Q_i - P^{[k]}(s_i, t_i), i \in I_{fit}$$

6. 计算活动控制顶点拟合向量

$$\delta_h^{[k]} = \sum_{l \in I \cap J_{\text{fit}}} N_h(s_l, t_l) d_l^{[k]}, h \in J_{\text{act}}$$
7. 计算活动控制顶点光顺向量

$$\eta_h^{[k]} = \sum_{l=1}^{n_1 n_2} [\mathcal{F}_{ss,l}(N_{ss,h}(s, t)) + 2\mathcal{F}_{st,l}(N_{st,h}(s, t)) + \mathcal{F}_{tt,l}(N_{tt,h}(s, t))] P_l^{[k]}, h \in J_{\text{act}}$$
8. 计算第 $k+1$ 次迭代曲面的新控制顶点

$$P_h^{[k+1]} = P_h^{[k]} + \mu_h [(1 - \omega_h) \delta_h^{[k]} - \omega_h \eta_h^{[k]}], h \in J_{\text{act}}$$

$$P_h^{[k+1]} = P_h, h \in J_{\text{fix}}$$
9. 计算第 $k+1$ 次迭代曲面

$$P^{[k+1]}(s, t) = \sum_{h=1}^{n_1 n_2} N_h(s, t) P_h^{[k+1]}$$
10. 计算局部拟合误差

$$E_{\text{fit}}^{[k+1]} = \max_{i \in I_{\text{fit}}} \|Q_i - P^{[k+1]}(s_i, t_i)\|$$
11. IF $|E_{\text{fit}}^{[k+1]} - E_{\text{fit}}^{[k]}| < \epsilon$
RETURN $P^{[k+1]}(s, t)$
12. $k = k + 1$
13. RETURN $P^{[k]}(s, t)$

6 数值实验

在这一节, 我们通过数值实验来验证局部 Fairing-PIA 的有效性。

在数值实验中, 对于曲线情形, 对数据点 $\{Q_i\}_{i=1}^m$, 我们用累加弦长法给每个数据点分配对应的参数 $\{t_i\}_{i=1}^m$, 即

$$\begin{cases} t_1 = 0, t_m = 1, \\ t_{i+1} = t_i + \frac{\|Q_i - Q_{i-1}\|}{\sum_{j=2}^m \|Q_j - Q_{j-1}\|}, i = 2, 3, \dots, m-1 \end{cases}$$

从 m 个数据点中选取 n 个点作为控制顶点, 满足 $P_j^{[0]} = Q_{i_j}, 1 = i_1 < i_2 < \dots < i_n = m$. 一般可用如下方法:

$$\begin{cases} P_1^{[0]} = Q_1, \\ P_n^{[0]} = Q_m, \\ P_j^{[0]} = Q_{\left[\frac{m(j-1)}{n-1}\right]}, j = 2, 3, \dots, n-1. \end{cases}$$

其中 $\left[\frac{m(j-1)}{n-1}\right]$ 表示不大于 $\frac{m(j-1)}{n-1}$ 的最大整数。

在实验中, 用 3 次 B 样条曲线来拟合数据点, 定义节点向量 $\{0, 0, 0, 0, \hat{t}_5, \dots, \hat{t}_n, 1, 1, 1, 1\}$, 其中 $\hat{t}_j = (t_{i_{j-3}} + t_{i_{j-2}} + t_{i_{j-1}})/3, j = 5, \dots, n$.

假设在数值实验中仅调整对应参数值为 $[t_a, t_b]$

的曲线段, 该段曲线对应的数据点的下标集为 I_{fit} 。我们用如下能量函数来度量这段曲线的局部光顺性:

$$E_{\text{eng}}^{\text{local}} = \int_{t_a}^{t_b} \|P^{(r)}(t)\|^2 dt, r = 1, 2, 3. \quad (17)$$

局部拟合误差为

$$E_{\text{fit}}^{\text{local}} = \max_{i \in I_{\text{fit}}} \|Q_i - P(t_i)\|. \quad (18)$$

若相邻两次迭代局部拟合误差(18)之差的绝对值小于给定常值 ϵ , 或迭代次数达到 K , 则迭代停止. 在本文数值实验中, 取 $\epsilon = 1 \times 10^{-7}, K = 800$ 。

曲面的例子中, 数据点对应参数、控制顶点的选取以及迭代停止条件类似于曲线情形。

所有的实验均在具有 16 GB RAM 和 Intel Core i9-12900H 2.50 GHz CPU 的计算机上完成。

6.1 曲线

本节将用 3 个例子来展示曲线局部 Fairing-PIA 的有效性. 这 3 个例子分别为:

(1) 海星: 从如下函数中均匀采样得到 100 个数据点

$$\begin{cases} x(t) = (1 + \frac{1}{5} \cos(5t)) \cos(t), t \in [0, 2\pi] \\ y(t) = (1 + \frac{1}{5} \cos(5t)) \sin(t) \end{cases}$$

(2) 音符: 从音符状曲线上采集的 305 个数据点

(3) 翼型: 飞机机翼翼面上采集的 49 个数据点

图 1 中数据为海星函数曲线上均匀取样得到的 100 个数据点。图 1(a) 为数据点的拟合曲线以及曲率梳, 图 1(b) 为图 1(a) 局部细节放大。我们仅调整部分曲线, 红色框内的橙色点为活动控制顶点。在算法迭代的过程中, 我们发现控制顶点的光顺向量(4)的数量级远大于拟合向量(3)。为了进一步提升曲面光顺性, 我们希望能够通过光顺权重将拟合项与光顺项控制在同一数量级, 并且希望光顺项大于拟合项。基于以上考虑, 按逆时针方向, 我们将这五个活动控制顶点的光顺权重分别设为 $1 \times 10^{-6}, 1 \times 10^{-6}, 5 \times 10^{-5}, 8 \times 10^{-5}, 1 \times 10^{-5}$, 光顺泛函(2)中取 $r=2$ 。在后续数值实验中, 光顺权重的选取都基于类似的想法。图 1(c) 和图 1(e) 分别为局部迭代 1 次和 3 次后的拟合曲线以及曲率梳。图 1(d) 和图 1(f) 分别为这部分曲线的局部细节放大。可以看到部分曲线段对应的曲率梳发生了变化, 曲线变得更加光顺, 但其余部分曲线没有发生变化。

图 2 中的数据点为音符状曲线上采集的 305 个数据点。我们选取 68 个数据点作为初始控制顶点,

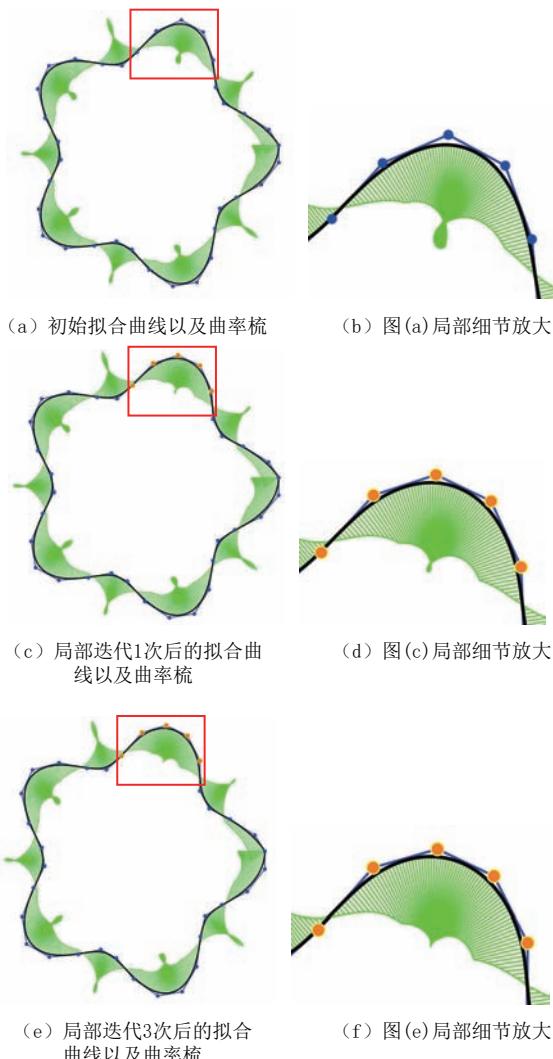


图1 曲线局部光顺实例1

得到了初始曲线以及对应曲率梳(图2(a))。我们选取两处同时进行局部迭代调整(红框内曲线段)，图2(b)描绘了两处红框内曲线细节。我们将两处的对应控制顶点(橙色点)光顺权重均设为 5×10^{-6} ，光顺泛函(2)中取 $r=2$ ，进行局部迭代。这两段曲线同时得到调整，图2(c)为局部迭代后的拟合曲线。我们可以看到曲线不但更贴近数据点，而且更为光顺。同时只有部分曲线段发生了变化，其余部分曲线保持不变。图2(d)描绘了对应曲线段细节。

图3为飞机机翼上采集的49个数据点。我们用19个控制顶点的B样条曲线拟合这些数据点。图3(a)为初始的拟合曲线以及曲率梳，可以看到机翼上侧光顺性较好，而右下侧曲线段光顺性不够好。因此我们仅调整机翼右下侧局部曲线，将对应的七个活动控制顶点的光顺权重均设为 1×10^{-5} 。图3(b)为局部迭代后的拟合曲线，橙色点为活动控

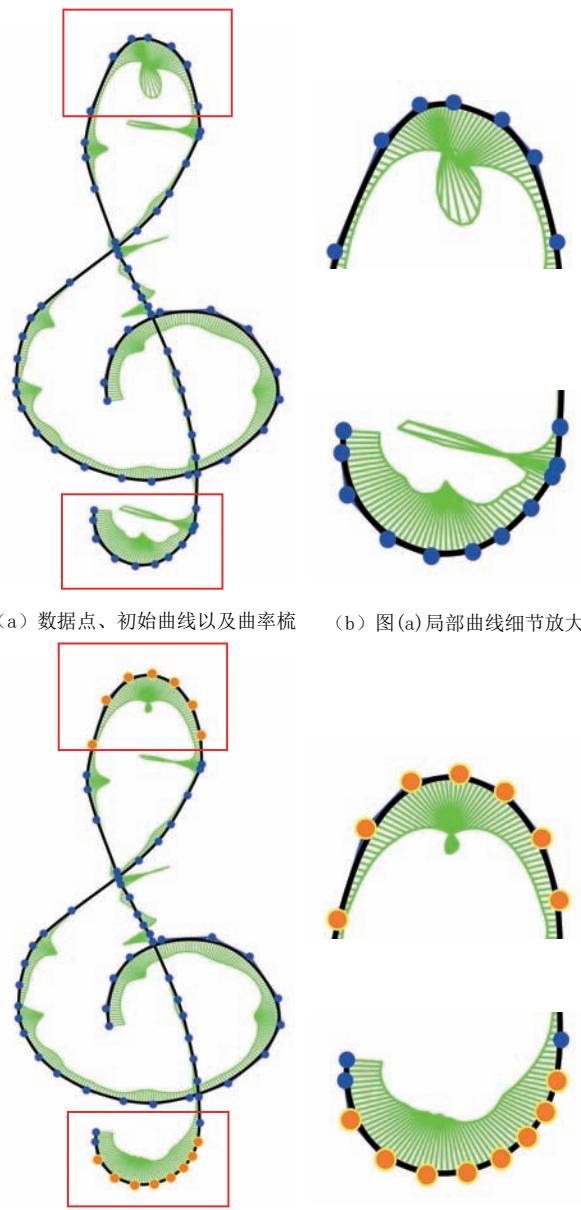


图2 曲线局部光顺实例2

制顶点。该段曲线光顺程度得到提高，其余部分曲线未发生变化。图3(c)为进一步提高第三和第四个控制顶点(红色点)的光顺权重后，再次局部迭代得到的拟合曲线以及曲率梳。其中第三和第四个控制顶点(红色点)的光顺权重分别设为 $5 \times 10^{-3}、1 \times 10^{-3}$ ，其余光顺权重不变。图3(d)为在图3(c)的基础上提高第六个控制顶点(红色点)的光顺权重至为 1.6×10^{-4} ，局部迭代后得到的拟合曲线与曲率梳。

在第三节，我们提到了 r (见公式(2))不同取值时分别对应的能量类型。通过数值实验，我们比较了不同光顺权重与不同 r 的取值对于曲线光顺效果

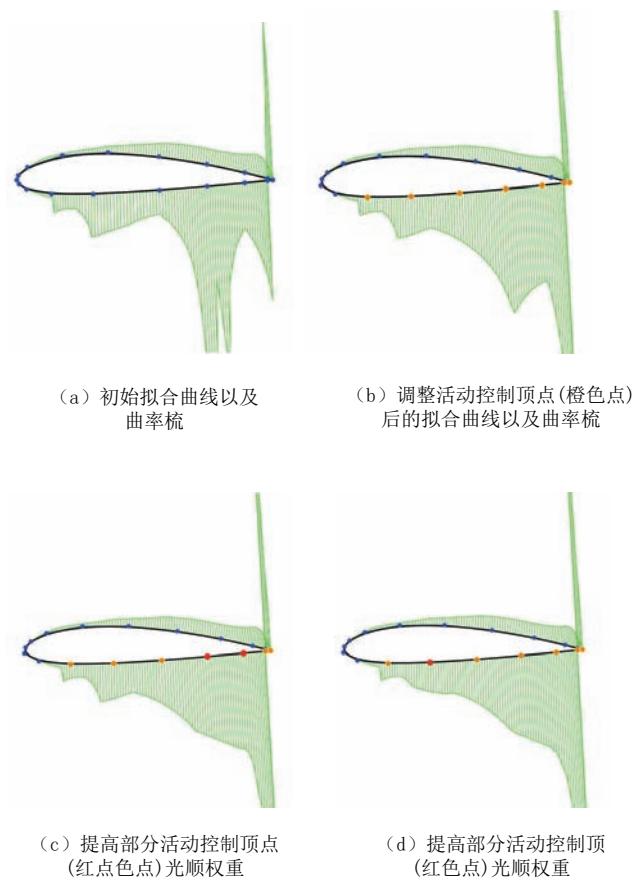


图3 曲线局部光顺实例3

影响(图4)。当光顺权重相同时, $r=3$ 的光顺效果最好, 拟合效果最差, $r=1$ 情形则正好相反。在 r 取定时, 曲线的光顺效果随光顺权重的增大而变好, 拟合效果则相反。根据该实验我们也意识到, r 的不同取值, 对应的合适的光顺权重范围不同。

在图4这个例子里, 对于 $r=3$ 的情况, 我们选取较小的值 1×10^{-8} 作为光顺权重; 对于 $r=1$ 与 $r=2$ 我们则分别选择光顺权重为 1×10^{-4} 与 1×10^{-6} 进行实验, 结果见图4(d)。对应的局部拟合误差与运行时间见表1。

r 为光顺泛函中基函数求导的阶数, r 越大, 计算成本越高。但根据光顺泛函的定义, 在算法的迭代过程中其值不变, 由此我们可以在迭代开始前通过数值积分的方法计算一次光顺泛函的结果, 并将该结果保存下来, 在之后的迭代过程中不断调用这些常值。因此, 整个算法的时间成本主要受算法的收敛速度和迭代次数影响。根据上述实验我们发现, r 的取值越小, 收敛速度越快, 迭代次数越少, 算法消耗的时间越少。

虽然 $r=1$ 时拟合误差与时间成本较小, 但其光顺效果远不如 $r=2$ 与 $r=3$ 。在本文的这个例子中,

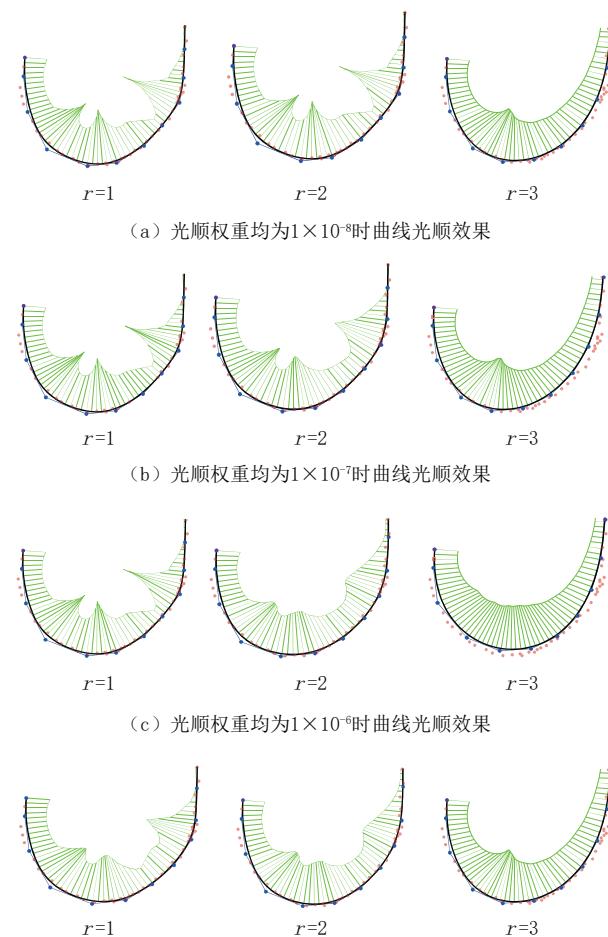
图4 光顺权重与 r 的取值对曲线光顺效果的影响

表1 算法迭代次数、局部拟合误差、运行时间对比

r 的取值	迭代次数	局部拟合误差	运行时间(s)
$r=1$	26	0.000 942 17	0.001 186
$r=2$	53	0.001 257 60	0.001 375
$r=3$	63	0.001 823 17	0.001 437

如果不考虑计算成本与运行时间, 选择 $r=3$ 以及较小的光顺权重例如 1×10^{-8} 能够达到较好的光顺效果与拟合效果。在实际应用中, 我们还需要考虑迭代次数与运行时间。结合算法效果与计算成本, 可以选择 $r=2$ 以及稍大的光顺权重, 例如 1×10^{-6} 。

6.2 曲面

在本节, 我们将用两个例子来验证曲面局部Fairing-PIA的有效性。这两个例子对应的数据集分别为人类牙齿上采集的 81×86 个数据点(图5)以及汽车外壳上采集的 258×279 个数据点(图6)。

图5为人类牙齿上采集的 81×86 数据点, 我们用 20×20 个控制点的B样条曲面拟合这些数据。

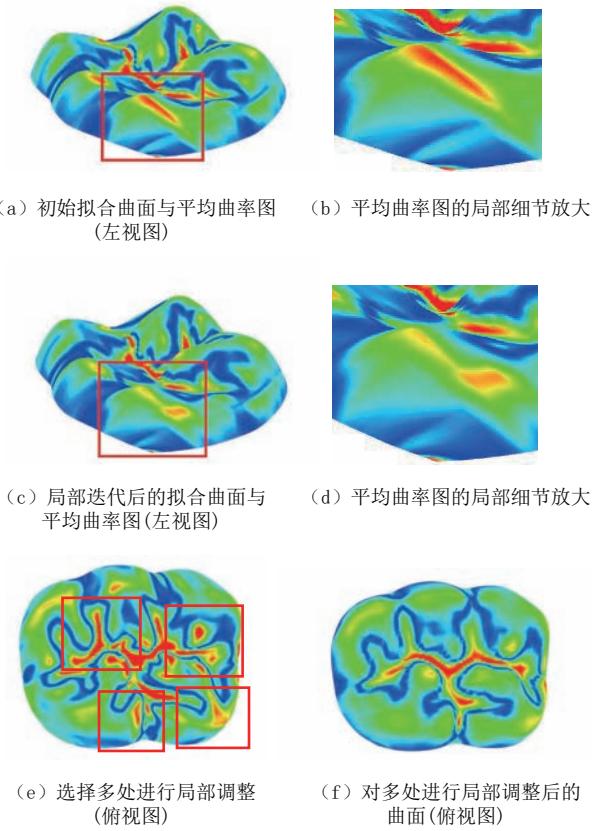


图5 曲面局部光顺实例1

图5(a)为初始的拟合曲面平均曲率图,我们选取了一部分曲面(红色框内曲面)进行调整。图5(b)为局部曲面细节。图5(c)为活动控制顶点迭代更新

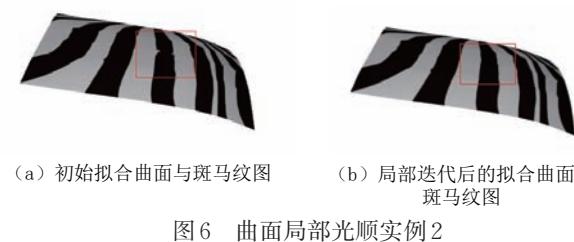


图6 曲面局部光顺实例2

后的拟合曲面平均曲率图,图5(d)为局部曲面细节。实例详细数据见表2。对于图5这种有多个曲率变化较大的区域,我们可以选择多处区域(图5(e)红框处)继续进行局部调整,调整结果如图5(f)。

我们选取了部分曲面对应的控制顶点进行调整。图6(b)描绘了经过局部迭代调整后的曲面斑马纹。实例详细数据见表2。

本文给出了所有数值实验中实例的详细数据,包含了数据点个数、控制点个数以及活动控制顶点的光顺权重,见表2。同时,本文比较了五个实例在全局Fairing-PIA与局部Fairing-PIA下在时间和局部拟合误差上的区别,见表3与图7。经数值实验,局部Fairing-PIA的在运行时间与拟合效果上都优于全局Fairing-PIA。此外,本文还比较了光顺前后的局部能量变化,结果见表4。从表4中可知,该算法能够有效降低曲线、曲面局部能量,达到光顺的效果,这也进一步证明了我们算法的有效性。

表2 曲线曲面实例相关数据

实例	数据点	控制顶点	活动控制顶点光顺权重				
			6 th —7 th	8 th	9 th	10 th	11 th
曲线局部光顺实例1	100	35	6 th —7 th	8 th	9 th	10 th	11 th
			1×10^{-6}	5×10^{-5}	8×10^{-5}	1×10^{-5}	
曲线局部光顺实例2	305	68	34 th —38 th	59 th —66 th			
			5×10^{-6}	5×10^{-6}			
曲线局部光顺实例3	49	17	1 st —2 nd	3 rd	4 th	5 th	6 th
			1×10^{-5}	5×10^{-3}	1×10^{-3}	1×10^{-5}	1.6×10^{-4}
曲面局部光顺实例1	81×61	20×20	75 th —80 th	95 th —100 th	115 th —120 th	135 th —140 th	155 th —160 th
			1×10^{-4}	5×10^{-4}	1×10^{-3}	1×10^{-3}	5×10^{-4}
曲面局部光顺实例2	258×279	40×40	821 st —826 st	861 st —866 st	901 st —906 st	941 st —946 st	981 st —986 st
			1×10^{-4}	1×10^{-3}	3×10^{-3}	3×10^{-3}	5×10^{-4}

表3 全局与局部Fairing-PIA对比

实例	运行时间(s)		局部拟合误差	
	全局Fairing-PIA	局部Fairing-PIA	全局Fairing-PIA	局部Fairing-PIA
曲线局部光顺实例1	0.0193	0.00321	0.006086	0.001391
曲线局部光顺实例2	0.0140	0.00286	0.007430	0.005450
曲线局部光顺实例3	0.0126	0.00556	0.005389	0.004717
曲面局部光顺实例1	30.365	4.325	0.051928	0.042609
曲面局部光顺实例2	8.632	1.623	0.024620	0.019231

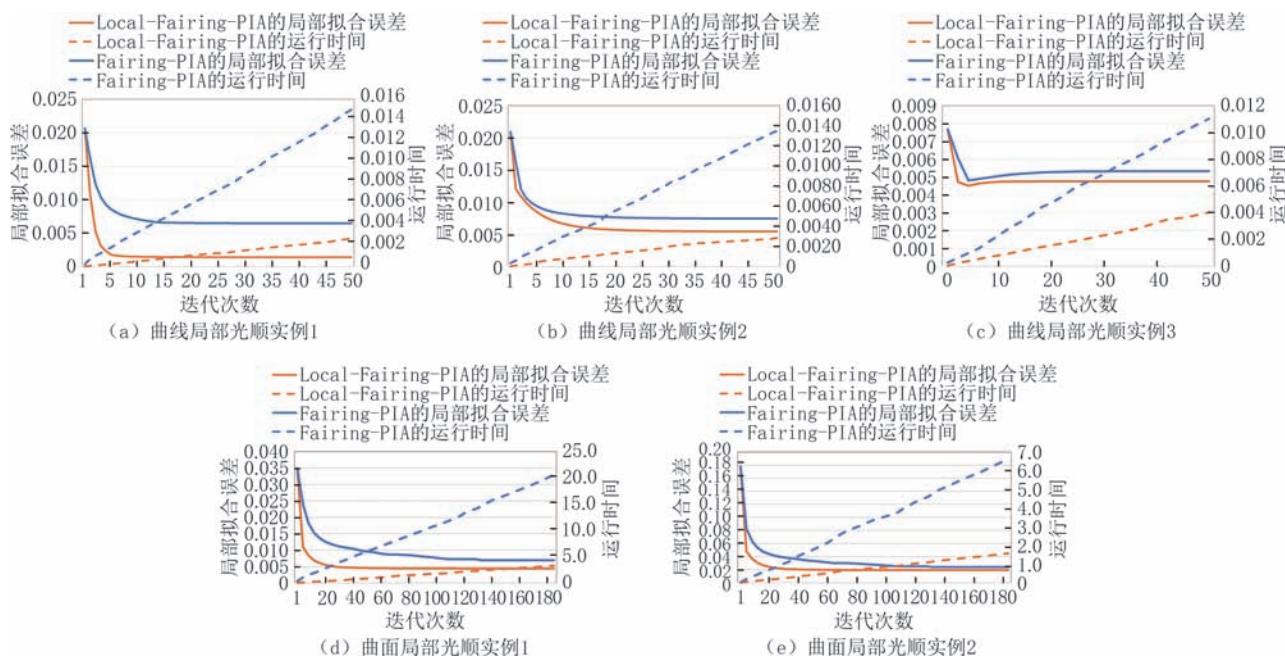


图7 Fairing-PIA与局部Fairing-PIA的运行时间与拟合误差对比

表4 曲线曲面实例光顺前后局部能量对比

实例	局部初始能量	局部最终能量	局部能量降低百分比
曲线局部光顺实例1	2090.7852	1824.5023	12.737%
曲线局部光顺实例2	34765.6951	29985.9176	13.749%
曲线局部光顺实例3	27854.3253	24358.8675	12.550%
曲面局部光顺实例1	109.8532	96.3248	12.315%
曲面局部光顺实例2	235.6879	198.6358	15.721%

7 结语

本文提出了一种混合曲线曲面的局部Fairing-PIA算法格式，并证明了该算法的收敛性。和全局Fairing-PIA算法相比，本文算法更加灵活，收敛速度更快。局部Fairing-PIA能够仅对部分控制顶点进行调整更新，在兼顾拟合精度的基础上通过这些控制顶点的光顺权重调整曲线曲面的局部形状，使其具有良好的光顺性。相较于传统的基于能量泛函极小的光顺方法，局部Fairing-PIA算法能够兼顾光顺效果与拟合误差对曲线曲面进行局部调整，同时避免了大规模矩阵运算，降低了计算成本。理论与数值实验证明了局部Fairing-PIA算法的有效性。

在后续工作中，我们将进一步研究光顺权重对于光顺效果与拟合误差的影响，在拟合精度确定的情况下，找到更合适的光顺权重范围。

参 考 文 献

- [1] Sariöz E. An optimization approach for fairing of ship hull forms. *Ocean Engineering*, 2006, 33(16): 2105-2118
- [2] Westgaard G, Nowacki H. A process for surface fairing in irregular meshes. *Computer Aided Geometric Design*, 2001, 18(7): 619-638
- [3] Li W, Krist S. Spline-based airfoil curvature smoothing and its applications. *Journal of Aircraft*, 2005, 42(4): 1065-1074
- [4] Jiang Y N, Lin H W, Huang W X. Fairing-PIA: progressive-iterative approximation for fairing curve and surface generation. *The Visual Computer*, 2023, 40(3): 1467-1484
- [5] Burchard H G. Approximation with Aesthetic Constraints. *Designing Fair Curves and Surfaces*, 1994: 3-28
- [6] Sapidis, Nicholas S. *Designing fair curves and surfaces: Shape Quality in Geometric Modelling and Computer-Aided Design*. USA: Society for Industrial and Applied Mathematics, 1994
- [7] Veltkamp, R C, Wesselink W. Modeling 3D curves of minimal energy. *Computer Graphics Forum*, 1995, 14(3): 97-110
- [8] Wang X F, Cheng F H, Barsky B A. Energy and B-spline interproximation. *Computer-Aided Design*, 1997, 29(7): 485-496
- [9] Hosaka M. Theory of curve and surface synthesis and their smooth fitting. *Ipsj Magazine*, 1969, 9: 60-68
- [10] Lott N J, Pulln D I. Method for fairing B-spline surfaces. *Computer-Aided Design*, 1988, 20(10): 597-600
- [11] Greiner G. Variational design and fairing of spline surfaces. *Computer Graphics Forum*, 2010, 13(3): 143-154
- [12] Farin G, Rein G, Sapidis N, Worsey A. Fairing cubic B-spline curves. *Computer Aided Geometric Design*, 1987, 4(1-2): 91-103
- [13] Hahmann S, Konz S. Knot-removal surface fairing using search

- strategies. Computer-Aided Design, 1998, 30(2): 131-138
- [14] Zhang C M, Cheng F H. Removing local irregularities of NURBS surfaces by modifying highlight lines. Computer-Aided Design, 1998, 30(12): 923-930
- [15] Tzvetomir, Ivanov, Vassilev. Fair interpolation and approximation of B-splines by energy minimization and points insertion. Computer-Aided Design, 1996, 28(9): 753-760
- [16] Sun Y K, Zhu X X. Multi-resolution representation and smoothing of arbitrary B-spline surface. Journal of Engineering Graphics, 1998, 19(3): 49-54 (in Chinese)
(孙延奎, 朱心雄. 任意B样条曲面的多分辨率表示及光顺. 工程图学报, 1998, 19 (3): 49-54)
- [17] Abdul-Rahman H S, Jiang X J, Scott P J. Freeform surface filtering using the lifting wavelet transform. Precision Engineering, 2013, 37(1): 187-202
- [18] Ceruti A, Liverani A, Caligiana G. Fairing with neighbourhood LOD filtering to upgrade interactively B-Spline into Class-A curve. International Journal on Interactive Design and Manufacturing, 2014, 8(2): 67-75
- [19] Shi F Z. Computer-Aided Geometric Design and Non-Uniform Rational B-Spline. Beijing: Higher Education, Press, 2001(in Chinese)
(施法中. 计算机辅助几何设计与非均匀有理B样条. 北京: 高等教育出版社, 2001)
- [20] Qin K H. A New Algorithm for Degree-Raising of Non-Uniform B-Spline Curves. Chinese Journal of Computers, 1996, 19(7): 537-542 (in Chinese)
(秦开怀. 非均匀B样条曲线升阶的新算法. 计算机学报, 1996, 19(7): 537-542)
- [21] Wang S W, Liu L G, Zhang J Y, Yang Z W, Dong G C. Sparsity optimization based curve fairing. Journal of Computer-Aided Design & Computer Graphics, 2016, 28(12): 2043-2051 (in Chinese)
(王士玮, 刘利刚, 张举勇, 等. 基于稀疏模型的曲线光顺算法. 计算机辅助设计与图形学学报, 2016, 28(12): 2043-2051)
- [22] Yang X N. Fitting and fairing Hermite-type data by matrix weighted NURBS curves. Computer-Aided Design, 2018, 102: 22-32
- [23] Yu Y C, Ji X G, Yan C, et al. Research on multi-resolution fairing for non-uniform rational B-spline curves. Machine Design and Research, 2019, 35(02): 1-5 (in Chinese)
(于益超, 纪小刚, 闫晨, 等. 非均匀有理B样条曲线的多分辨率光顺研究. 机械设计与研究, 2019, 35(02): 1-5)
- [24] Jiang X, Wang B L, Huo G Y, Su C, Yan D M, Zheng Z M. Scattered points interpolation with globally smooth B-spline surface using iterative knot insertion. Computer-Aided Design, 2022, 148: 103244
- [25] Lin H W, Wang G J, Dong C S. Constructing iterative non-uniform B-spline curve and surface to fit data points. Science China Information Sciences, 2004, 47(3): 315-331
- [26] Lin H W, Bao H J, Wang G J. Totally positive bases and progressive iteration approximation. Computers & Mathematics with Applications, 2005, 50(3): 575-586
- [27] Qi D X, Tian Z X, Zhang Y X, Feng J B. The method of numeric polish in curve fitting. Acta Mathematica Sinica, 1975, 18(3): 173-184 (in Chinese)
(齐东旭, 田自贤, 张玉心, 冯家斌. 曲线拟合的数值磨光方法. 数学学报, 1975, 18(3), 173-184)
- [28] Qi D X. Some notes on mathematical methods for computer-aided geometric modeling. Journal of North China University of Technology, 1991, 3(1): 1-8 (in Chinese)
(齐东旭. 关于计算机辅助几何造型数学方法的若干注记. 北方工业大学学报, 1991, 3(1): 1-8)
- [29] Qi D X. Fractals and Their Computer Generation. Beijing: Science Press, 1994 (in Chinese)
(齐东旭. 分形及其计算机生成. 北京: 科学出版社, 1994)
- [30] De Boor C. How does Agee's smoothing method work? // Proceedings of the 1979 Army Numerical Analysis and Computers Conference, ARO Report, 1979, 299-302
- [31] Deng C Y, Ma W Y. Efficient evaluation of subdivision schemes with polynomial reproduction property. Journal of Computational and Applied Mathematics, 2016, 294: 403-412
- [32] Lin H W, Maekawa T, Deng C Y. Survey on geometric iterative methods and their applications. Computer-Aided Design, 2018, 95: 40-51
- [33] Lu L Z. Weighted progressive iteration approximation and convergence analysis. Computer Aided Geometric Design, 2010, 27(2): 129-137
- [34] Lin H W. Local progressive-iterative approximation format for blending curves and patches. Computer Aided Geometric Design, 2010, 27(4): 322-339
- [35] Lin H W, Zhang Z Y. An extended iterative format for the progressive-iteration approximation. Computers & Graphics, 2011, 35(5): 967-975
- [36] Deng C Y, Lin H W. Progressive and iterative approximation for least squares B-spline curve and surface fitting. Computer-Aided Design, 2014, 47: 32-44
- [37] Hamza F Y, Jiang Y N, Lin H W. Gauss-Seidel progressive and iterative approximation for least squares fitting. Journal of Computer-Aided Design & Computer Graphics, 2021, 33(01): 1-10 (in Chinese)
(Hamza F Y, 蒋旖旎, 林宏伟. Gauss-Seidel最小二乘渐进迭代逼近. 计算机辅助设计与图形学学报, 2021, 33(01): 1-10)
- [38] Liu S J, Liu T, Hu L, Shang Y Y, Liu X R. Variational progressive-iterative approximation for RBF-based surface reconstruction. The Visual Computer, 2021, 37(9-11): 1-13
- [39] Wang H D. Implicit randomized progressive-iterative approximation for curve and surface reconstruction. Computer-Aided Design, 2022, 152: 103376
- [40] Liu C Z, Li J C, Hu L J. Jacobi-PIA algorithm for Bi-cubic B-spline interpolation surfaces. Graphical Models, 2022, 120: 101134
- [41] Zhang C M, Zhang P F, Cheng F H. Fairing spline curves and surfaces by minimizing energy. Computer-Aided Design, 2001, 33(13): 913-923
- [42] Meier H, Nowacki H. Interpolating curves with gradual changes in curvature. Computer Aided Geometric Design, 1987, 4(4): 297-305



LIN Jia, M. S. candidate. Her research interests include computer aided geometric design.

LIN Hong-Wei, Ph. D., professor. His research interests include computer aided geometric design, computer aided topology design, topological data analysis, quantum graphics.

Background

Curve and surface fairing hold significant value in various fields, such as automotive body design, aircraft wing design, and ship hull design. The fairness of these curves and surfaces directly influences the quality, physical performance, and aesthetics of the products. Curve and surface fairing has long been a critical concern in the field of computer-aided design. International research on this topic began around the 1960s. Traditional methods for curve and surface fairing primarily include global fairing techniques based on minimizing energy functionals and local fairing methods that modify selected "bad points." Despite the many existing methods for fairing curves and surfaces, several challenges remain, such as low levels of automation, high computational complexity, and inefficient algorithms.

The progressive iterative approximation (PIA) is a geometric iterative method that generates a series of curves by adjusting control vertices to interpolate or approximate given data points. This method originated from the uniform cubic B-spline gain-loss correction algorithm developed in 1975 and was formally proposed and named by Lin et al. in 2005. By avoiding large-scale matrix computations, the algorithm becomes more efficient and flexible. In recent years, the

progressive iterative approximation algorithm has seen further development, leading to improvements in speed and reductions in computational complexity.

Fairing-PIA is another geometric iterative method that generates a series of fairing curves and surfaces by adjusting control vertices. This paper introduces a local fairing method based on the global Fairing-PIA. In the local Fairing-PIA approach, only certain control vertices and their corresponding fairing weights are adjusted to optimize the local shape of the curve or surface. The fairing of curves and surfaces can be fine-tuned point by point or segment by segment using local Fairing-PIA. Different parts of a curve or surface can exhibit varying degrees of smoothness. Compared to traditional fairing methods, the local Fairing-PIA algorithm effectively balances fairing effects and fitting errors in local adjustments of curves and surfaces. Additionally, it avoids large-scale matrix computations, thus reducing computational costs significantly and providing a low-complexity, flexible, and lightweight solution to curve and surface fairing problems, making it a highly advantageous method in the field.

This paper is supported by the National Natural Science Foundation of China (No. 62272406 and 61932018).