高校应用数学学报
2025, 40(4): 379-393

混合曲面的改进GS-PIA算法

胡倩倩[1*], 董文晴[1], 姚振民[1], 王国瑾[2]

(1. 浙江工商大学 统计与数学学院, 浙江杭州 310018;
2. 浙江大学 数学科学学院, 浙江杭州 310027)

摘 要: 曲面拟合的PIA方法, 通常是通过垂直堆叠矩阵的列, 将曲面拟合转化成曲线拟合的PIA算法. 这需要计算矩阵的Kronecker积, 具有计算量大、运算时间长等缺点. 为了避免Kronecker积的计算, 将求解矩阵方程的改进Gauss-Seidel方法与经典PIA算法相结合, 文中提出了一种基于Gauss-Seidel型分裂的混合曲面的PIA算法(GSTS-PIA). 首先, 对待拟合数据点计算出差向量; 然后, 通过Gauss-Seidel型分裂矩阵计算出控制顶点的偏移量; 最后, 根据偏移量得到新的控制顶点, 从而生成拟合数据点的曲面. 理论分析证明了生成的曲面序列的极限插值于给定数据点. 用不同混合曲面拟合散乱数据点或规则曲面取样点的实验结果表明, 在达到相同拟合精度的情况下, GSTS-PIA算法比经典PIA算法在所需迭代次数上平均减少78.30%, 运算时间上平均减少80.96%.

关键词: 渐进迭代逼近; 数据拟合; 收敛速度; 混合曲面
中图分类号: TP391
文献标识码: A **文章编号**: 1000-4424(2025)04-0379-15

§1 引 言

数据插值是科学研究和工程计算中的一个重要问题. 通常, 可以通过直接求解相应的线性方程组来得到插值曲线或曲面. 然而, 对于大型数据拟合问题, 直接解法往往计算量大, 有时还不稳定. 经过进一步研究, 解决这一问题的各种迭代细化方法随之产生. 其中, 渐进迭代逼近法(progressive iterative approximation, PIA)是通过迭代地调整控制顶点, 生成一组曲线或曲面序列, 来插值原始数据点[1]. 该方法不仅避免了直接求解线性方程组, 而且还可以在求解过程中灵活加入用户所需的几何约束条件, 这也是与传统插值逼近方法的一个重要差别[2]. PIA方法因其动态操作、算法简单等优点, 目前已应用于几何外形设计、图像处理、数据压缩等领域[3-4].

收稿日期: 2024-07-04 修回日期: 2025-05-29
*通讯作者, E-mail: qianqian_hu@163.com
基金项目: 国家自然科学基金(62272406)

1975年, 齐东旭等[5]提出了均匀三次B样条曲线的盈亏修正算法. 1979年, de Boor[6]给出了其收敛性证明. 随后, Lin等[7]将这一思想推广到非均匀三次B样条曲线曲面. 2005年, Lin等[8]证明了具有归一化全正基的混合曲线曲面具有同样的特性, 并将其命名为渐进迭代逼近性质. 此后, 国内外学者提出了多种不同的PIA格式[9-16], 比如引入权因子来提高收敛速度的加权渐进迭代逼近(WPIA)[17], 具有更大灵活性的局部渐进迭代逼近(LPIA)[18], 用于处理大型数据拟合问题的最小二乘渐进迭代逼近(LSPIA)[19], 提高隐式曲线曲面重建效率的隐式渐进迭代逼近(IPIA)[20], 用于Catmull-Clark细分的渐进迭代逼近[21], 以及用于曲线曲面光顺的渐进迭代逼近(Fairing-PIA)[22].

对于张量积混合曲面, 其基函数可以看作2个单变量全正基函数的乘积. 因此, 曲线的各种PIA性质可以自然地推广到张量积混合曲面, 其传统方法是通过垂直堆叠矩阵的列将点阵数据排列成列向量, 利用Kronecker积把曲面的PIA迭代格式转化为曲线形式. 这种方法虽然简单明了, 易于操作, 但是在实际运用中, 因矩阵Kronecker积的计算量非常大, 使得运算时间较长, 特别是在处理大型数据拟合问题上, 问题尤显严重. 那么, 如何解决这个问题呢? 笔者想到, 本质上, PIA方法可被视为求解线性方程组的Richardson迭代算法[23]. 经典迭代法还包括Jacobi迭代法、Gauss-Seidel迭代法、SOR迭代法等[24]. 如果将基于矩阵分裂的迭代方法融合到曲面的PIA算法中, 将可有效避免使用Kronecker积, 计算量会遽降. 例如, 双三次B样条曲面的Jacobi-PIA算法就是基于矩阵方程$AXB = C$的Jacobi分裂, 加上一个松弛因子而得到[14]. 混合曲面的HSS-PIA方法则是在Hermitian和skew-Hermitian分裂迭代的基础上得到的[25].

参考以上有关分裂迭代的思维, 运用基于Gauss-Seidel分裂方法[26], 对矩阵方程$AXB = C$中的系数矩阵A和B分别进行多次诱导分裂, 并与经典PIA方法相结合, 本文开发了一种混合曲面的GSTS-PIA方法. 为了分析该方法的收敛性, 利用Kronecker积的性质, 将迭代格式中的矩阵形式改写成列向量形式, 证明了由迭代格式生成的曲面序列是收敛的, 并且其极限曲面插值于给定数据点. 实验结果表明了该方法的有效性.

§2 混合曲面的GSTS-PIA算法

给定一个数据点阵$\{Q_{ij}\}_{i=1,j=1}^{m,n}$, 对每个数据点$Q_{ij}$赋予参数值$(u_i, v_j)$, 并满足

$$u_1 < u_2 < \cdots < u_m, \quad v_1 < v_2 < \cdots < v_n.$$

为了保证边界点处的插值[7], 通过以下方式将原始数据点$\{Q_{ij}\}_{i=1,j=1}^{m,n}$扩充得到初始的控制顶点$\{P_{ij}^{(0)}\}_{i=0,j=0}^{m+1,n+1}$,

$$\begin{cases} P_{ij}^{(0)} = Q_{ij}, i = 1, 2, \cdots, m, j = 1, 2, \cdots, n; \\ P_{i0}^{(0)} = P_{i1}^{(0)}, P_{i,n+1}^{(0)} = P_{i,n}^{(0)}, P_{0j}^{(0)} = P_{1j}^{(0)}, P_{m+1,j}^{(0)} = P_{m,j}^{(0)}, i = 1, 2, \cdots, m, j = 1, 2, \cdots, n; \\ P_{00}^{(0)} = P_{11}^{(0)}, P_{0,n+1}^{(0)} = P_{1,n}^{(0)}, P_{m+1,0}^{(0)} = P_{m,1}^{(0)}, P_{m+1,n+1}^{(0)} = P_{m,n}^{(0)}, \end{cases}$$

从而构造一张初始混合曲面

$$S^{(0)}(u, v) = \sum_{i=0}^{m+1} \sum_{j=0}^{n+1} P_{ij}^{(0)} B_i(u) B_j(v), u_1 \le u \le u_m, v_1 \le v \le v_n,$$

其中$\{B_i(u), i = 1, \cdots, m\}$是一组$m$次全正混合(normalized totally positive, NTP)基函数[9], 这

组基在参数列$\{u_1 < u_2 < \cdots < u_m\}$上的配置矩阵为

$$A_1 = (B_j(u_i))_{\substack{i=1,\ldots,m\\ j=1,\ldots,m}} \triangleq \begin{pmatrix} B_1(u_1) & B_2(u_1) & \ldots & B_m(u_1) \\ B_1(u_2) & B_2(u_2) & \cdots & B_m(u_2) \\ \vdots & \vdots & \ddots & \vdots \\ B_1(u_m) & B_2(u_m) & \cdots & B_m(u_m) \end{pmatrix}.$$

类似地, n次NTP基函数$\{B_j(v), j = 1, \cdots, n\}$在$\{v_1 < v_2 < \cdots < v_n\}$上的配置矩阵为

$$A_2 = (B_j(v_i))_{\substack{i=1,\ldots,n\\ j=1,\ldots,n}}.$$

扩充后的数据点对应的配置矩阵为

$$B_1 = \begin{pmatrix} 1 & & \\ & A_1 & \\ & & 1 \end{pmatrix}, \quad B_2 = \begin{pmatrix} 1 & & \\ & A_2 & \\ & & 1 \end{pmatrix}. \tag{1}$$

本文要解决的曲面拟合问题是寻找控制顶点为$\{P_{ij}\}_{i=0,j=0}^{m+1,n+1}$的混合曲面去插值原始数据点$\{Q_{ij}\}_{i=1,j=1}^{m,n}$. 传统的PIA方法是采取垂直堆叠矩阵的列[14],即

$$p = [P_{00}, P_{10}, \cdots, P_{m+1,0}, P_{01}, \cdots, P_{m+1,1} \cdots, P_{m+1,n+1}]^{\mathrm{T}},$$
$$q = [Q_{00}, Q_{10}, \cdots, Q_{m+1,0}, Q_{01}, \cdots, Q_{m+1,1} \cdots, Q_{m+1,n+1}]^{\mathrm{T}},$$

此时, 曲面PIA方法中的配置矩阵是B_1和B_2的Kronecker积, 即

$$b = B_2 \otimes B_1,$$

其中$\otimes$表示Kronecker积, 那么曲面的PIA迭代格式可转化为曲线形式

$$\begin{cases} \eta^{(k)} = q - bp^{(k)}, \\ p^{(k+1)} = p^{(k)} + \eta^{(k)}. \end{cases}$$

上述做法需要计算矩阵的Kronecker积,计算量大. 为了避免Kronecker积的计算, 本文提出的GSTS-PIA算法舍去了传统的转化过程, 直接给出了迭代格式的矩阵表示, 从而极大地减小运算量.

对(1)定义的配置矩阵B_1进行如下Gauss-Seidel分裂

$$B_1 = L_1 - U_1, \tag{2}$$

其中

$$L_1 = \begin{pmatrix} 1 & & & & & \\ 0 & B_1(u_1) & & & & \\ 0 & B_1(u_2) & B_2(u_2) & & & \\ \vdots & \vdots & \vdots & \ddots & & \\ 0 & B_1(u_m) & B_2(u_m) & \cdots & B_m(u_m) & \\ 0 & 0 & \cdots & 0 & 0 & 1 \end{pmatrix}, U_1 = \begin{pmatrix} 0 & 0 & 0 & \cdots & 0 & 0 \\ & 0 & -B_2(u_1) & \cdots & -B_m(u_1) & 0 \\ & & 0 & \cdots & -B_m(u_2) & 0 \\ & & & \ddots & \vdots & \vdots \\ & & & & 0 & 0 \\ & & & & & 0 \end{pmatrix},$$

类似地, 对B_2^{T}进行如下Gauss-Seidel分裂

$$B_2^{\mathrm{T}} = L_2 - U_2, \tag{3}$$

其中矩阵L_2, U_2的定义类似(2)中的L_1, U_1.

定义方阵X的一个n次多项式$f_n(X)$为

$$f_n(X) = I + X + \cdots + X^n, \tag{4}$$

其中I是与X同阶的单位矩阵. 若(2)和(3)中的矩阵L_1, L_2是非奇异的, 那么令

$$M_1 = f_{r-1}(G_1)L_1^{-1}, \quad M_2 = L_2^{-1}f_{s-1}(G_2), \tag{5}$$

其中函数f_n如(4)所定义, 且

$$G_1 = L_1^{-1}U_1, \quad G_2 = U_2L_2^{-1}, \tag{6}$$

参数r, s是使得矩阵G_1和G_2的谱半径满足

$$(\rho^r(G_1)+1)^2 + (\rho^s(G_2)+1)^2 < 4 \tag{7}$$

的正整数.

首先, 对每个数据点Q_{ij}, 计算其对应的差向量[7]

$$\begin{cases} \delta_{ij}^{(0)} = Q_{ij} - S^{(0)}(u_i, v_j), i = 1, 2, \cdots, m, j = 1, 2, \cdots, n; \\ \{\delta_{h0}^{(0)} = \delta_{h1}^{(0)}, \delta_{h,n+1}^{(0)} = \delta_{h,n}^{(0)}\}_{h=1}^{m}, \{\delta_{0l}^{(0)} = \delta_{1l}^{(0)}, \delta_{m+1,l}^{(0)} = \delta_{m,l}^{(0)}\}_{l=1}^{n}; \\ \delta_{00}^{(0)} = \delta_{0,n+1}^{(0)} = \delta_{m+1,0}^{(0)} = \delta_{m+1,n+1}^{(0)} = 0, \end{cases}$$

将上式改写成矩阵形式为

$$\delta^{(0)} = Q - B_1P^{(0)}B_2^{\mathrm{T}},$$

其中B_1, B_2如(1)所示, $Q = (Q_{ij})_{\substack{i=0,\cdots,m+1\\ j=0,\cdots,n+1}} = P^{(0)}$是扩充数据点的矩阵表示,

$$P^{(k)} = (P_{ij}^{(k)})_{\substack{i=0,\cdots,m+1\\ j=0,\cdots,n+1}}, \ k = 0, 1, \cdots$$

是第k次迭代曲面的控制顶点的矩阵表示.

然后, 对每个控制顶点$P_{ij}^{(0)}$, 构造其差向量, 并表示成矩阵形式

$$\Delta^{(0)}{=}M_1\delta^{(0)}M_2,$$

其中矩阵M_1, M_2如(5)所定义, 由此得到新的迭代控制顶点

$$P^{(1)}{=}P^{(0)}{+}\Delta^{(0)},$$

从而生成新的混合曲面

$$S^{(1)}(u, v) = \sum_{i=0}^{m+1}\sum_{j=0}^{n+1} P_{ij}^{(1)}B_i(u)B_j(v).$$

类似地, 假设k次迭代后得到第k次迭代曲面的控制顶点$P^{(k)}$. 根据迭代格式

$$\begin{cases} \delta^{(k)} = Q - B_1P^{(k)}B_2^{\mathrm{T}} \\ \Delta^{(k)}{=}M_1\delta^{(k)}M_2 \\ P^{(k+1)}{=}P^{(k)}{+}\Delta^{(k)} \end{cases}, \tag{8}$$

其中矩阵M_1, M_2如式(5)所定义, 则第$k+1$次的迭代曲面可表示为

$$S^{(k+1)}(u, v) = \sum_{i=0}^{m+1}\sum_{j=0}^{n+1} P_{ij}^{(k+1)}B_i(u)B_j(v).$$

重复上述迭代过程可以生成一个曲面序列$\{S^{(k)}(u, v), \ k = 0, 1, \cdots\}$. 下面将证明这个曲面序列是收敛的, 并且其极限曲面插值于原始数据点$\{Q_{ij}\}_{i=1,j=1}^{m,n}$.

备注2.1 若参数$r = s = 1$满足(7)时, 迭代格式(8)退化为

$$P^{(k+1)} = P^{(k)} + L_1^{-1}(Q - B_1P^{(k)}B_2^{\mathrm{T}})L_2^{-1},$$

正如文献[26]所说，上式恰好是文献[27]提出的Gauss-Seidel-type迭代算法的迭代格式的矩阵表示形式. 在算法设计时, 先判断参数$r = s = 1$是否满足(7). 若不满足, 则r, s逐一增加, 直到满足(7)为止.

§3 GSTS-PIA算法的收敛性分析

引理3.1[28] 若矩阵A和$I-G$都非奇异, 那么存在唯一矩阵对W_G和N_G, 使得

$$G=W_GN_G,\quad A=W_G^{-1}-N_G.$$

其中W_G是非奇异矩阵. 称$A=W_G^{-1}-N_G$是矩阵A被G诱导出的分裂.

引理3.2[29-30] Kronecker积具有以下性质1)-3).

已知矩阵$A\in\mathbf{R}^{m\times m},B\in\mathbf{R}^{n\times n},C\in\mathbf{R}^{n\times n}$和$X\in\mathbf{R}^{m\times n}$, 那么

1) $vec(AXB)=(B^{\mathrm{T}}\otimes A)vec(X)$, 其中算子$vec(\cdot)$是通过垂直堆叠矩阵的列将矩阵转换成的列向量;

2) $(A\otimes B)(C\otimes D)=(AC)\otimes(BD)$;

3) $\lambda(I\otimes A+B^{\mathrm{T}}\otimes I-B^{\mathrm{T}}\otimes A)=\{\lambda_i+\mu_j-\lambda_i\mu_j|\lambda_i\in\lambda(A),\mu_j\in\lambda(B)\}$, 其中$\lambda$表示矩阵的谱集.

引理3.3 矩阵G_1和G_2如(6)定义, 若不等式(7)成立, 那么$\rho^r(G_1)<\sqrt{3}-1$, 且

$$\rho^s(G_2)<\sqrt{3}-1.$$

证 由于谱半径$\rho(G_1)>0$, $\rho(G_2)>0$, 那么$\rho^r(G_1)>0,\rho^s(G_2)>0$, 其中参数$r,s$是满足不等式(7)的正整数, 故$(\rho^r(G_1)+1)^2>1$. 又结合不等式(7), 可得到$(\rho^s(G_2)+1)^2<3$, 即$\rho^s(G_2)<\sqrt{3}-1$. 同理可证$\rho^r(G_1)<\sqrt{3}-1$.

引理3.4 若定义如(1)的配置矩阵B_1和B_2都是非奇异的, 其主对角线元素都非零, 并且存在$N_1,N_2\in\mathbf{N}^+$, 使得当$r\geq N_1,s\geq N_2$时, 由(6)定义的矩阵G_1和G_2的谱半径满足(7), 那么分别存在唯一的矩阵对M_1,N_1和M_2,N_2, 使得

$$B_1=M_1^{-1}-N_1,\quad B_2^{\mathrm{T}}=M_2^{-1}-N_2, \tag{9}$$

其中矩阵M_1,M_2如(5)所定义.

证 由于矩阵B_1的主对角线元素非零, 那么由(2)分裂出的下三角阵L_1是非奇异的. 已知存在$N_1\in\mathbf{N}^+$, 当$r\geq N_1$时, 矩阵G_1的谱半径满足(7), 根据引理3.3, 有$\rho(G_1^r)=\rho^r(G_1)<\sqrt{3}-1<1$, 那么矩阵$I-G_1^r$可逆. 又因为矩阵$B_1$是非奇异的, 根据引理3.1, 存在唯一的矩阵对$W_1,N_1$, 使得

$$G_1^r=W_1N_1, \tag{10}$$

而由G_1^r诱导出矩阵B_1的分裂为

$$B_1=W_1^{-1}-N_1,$$

其中W_1可逆. 由(10)得到$N_1=W_1^{-1}G_1^r$, 将其代入上式得

$$B_1=W_1^{-1}(I-G_1^r).$$

又配置矩阵B_1非奇异, 于是有

$$W_1=(I-G_1^r)B_1^{-1}=(I+G_1+\cdots+G_1^{r-1})(I-G_1)B_1^{-1}. \tag{11}$$

由于L_1可逆, 根据(2)和G_1的定义(6), 配置矩阵B_1可表示成

$$B_1=L_1-U_1=L_1(I-G_1).$$

将上式代入(11), 得到

$$W_1=(I+G_1+\cdots+G_1^{r-1})L_1^{-1}\ =f_{r-1}(G_1)L_1^{-1},$$

这里矩阵函数f_n如(4)所定义, 此时W_1就等于(5)中定义的M_1.

类似地, 根据引理3.1, 可证对矩阵B_2^{T}, 存在唯一的矩阵对M_2, N_2, 使得

$$B_2^{\mathrm{T}} = M_2^{-1} - N_2, \quad G_2^s = N_2 M_2,$$

其中矩阵M_2可逆, 如(5)所定义, G_2的定义如(6)所示.

定理3.1 若定义如(1)的配置矩阵B_1和B_2都是非奇异的, 其主对角线元素都非零, 并且存在$N_1, N_2 \in \mathbf{N}^+$, 使得当$r \geq N_1, s \geq N_2$时, 由(6)定义的矩阵$G_1$和$G_2$的谱半径满足(7), 那么根据迭代格式(8)生成的曲面序列$\{S^{(k)}(u,v),\ k = 0, 1, \cdots\}$是收敛的, 且它的极限曲面插值于数据点$\{Q_{ij}\}_{i=1,j=1}^{m,n}$.

证 根据迭代格式(8), 可以得到第k次迭代的控制顶点和第$k+1$次迭代的控制顶点之间的关系

$$P^{(k+1)} = P^{(k)} + M_1(Q - B_1 P^{(k)} B_2^{\mathrm{T}}) M_2,$$

其中, M_1, M_2如(5)所示, 由于配置矩阵B_1和B_2是非奇异的, 那么上式可改写成

$$P^{(k+1)} - B_1^{-1} Q (B_2^{\mathrm{T}})^{-1} = P^{(k)} - B_1^{-1} Q (B_2^{\mathrm{T}})^{-1} - M_1 B_1 [P^{(k)} - B_1^{-1} Q (B_2^{\mathrm{T}})^{-1}] B_2^{\mathrm{T}} M_2. \quad (12)$$

已知存在$N_1 \in \mathbf{N}^+$, 使得当$r \geq N_1$时, 如(6)定义的矩阵G_1的谱半径满足(7), 那么根据引理3.4, 存在唯一矩阵对M_1与N_1, 将配置矩阵B_1分裂成(9). 又根据(10), 有

$$M_1 B_1 = M_1(M_1^{-1} - N_1) = I - M_1 N_1 = I - G_1^r,$$

同理, 可以得到

$$B_2^{\mathrm{T}} M_2 = I - G_2^s,$$

其中可逆矩阵M_1, M_2如(5)所定义. 将以上两式同时代入(12), 得到

$$\begin{aligned}
&P^{(k+1)} - B_1^{-1} Q (B_2^{\mathrm{T}})^{-1} = \\
&P^{(k)} - B_1^{-1} Q (B_2^{\mathrm{T}})^{-1} - (I - G_1^r)[P^{(k)} - B_1^{-1} Q (B_2^{\mathrm{T}})^{-1}](I - G_2^s) = \\
&G_1^r P^{(k)} + P^{(k)} G_2^s - G_1^r P^{(k)} G_2^s - G_1^r B_1^{-1} Q (B_2^{\mathrm{T}})^{-1} - B_1^{-1} Q (B_2^{\mathrm{T}})^{-1} G_2^s + G_1^r B_1^{-1} Q (B_2^{\mathrm{T}})^{-1} G_2^s.
\end{aligned}$$

为了方便证明该迭代算法的收敛性, 利用引理3.2中Kronecker积的性质1, 用vec算子将上式的矩阵形式改写成列向量形式

$$\begin{aligned}
&p^{(k+1)} - (B_2^{-1} \otimes B_1^{-1}) q = \\
&[I \otimes G_1^r + (G_2^s)^{\mathrm{T}} \otimes I - (G_2^s)^{\mathrm{T}} \otimes G_1^r] p^{(k)} - [B_2^{-1} \otimes (G_1^r B_1^{-1})] q - \left[((G_2^s)^{\mathrm{T}} B_2^{-1}) \otimes B_1^{-1}\right] q + \\
&\left[((G_2^s)^{\mathrm{T}} B_2^{-1}) \otimes (G_1^r B_1^{-1})\right] q,
\end{aligned}$$

其中$\otimes$表示Kronecker积, $p^{(k+1)} = vec(P^{(k+1)}), q = vec(Q)$分别是数据点$P^{(k+1)}$和$Q$的列向量表示形式.

根据引理3.2中Kronecker积的性质2, 上式可改写成

$$p^{(k+1)} - (B_2^{-1} \otimes B_1^{-1}) q = D\left[p^{(k)} - (B_2^{-1} \otimes B_1^{-1}) q\right],$$

其中D是迭代矩阵

$$D = I \otimes G_1^r + (G_2^s)^{\mathrm{T}} \otimes I - (G_2^s)^{\mathrm{T}} \otimes G_1^r. \quad (13)$$

重复上述迭代公式, 可以得到

$$\begin{aligned}
&p^{(k+1)} - (B_2^{-1} \otimes B_1^{-1}) q = \\
&D^2\left[p^{(k-1)} - (B_2^{-1} \otimes B_1^{-1}) q\right] = \cdots = D^{k+1}\left[p^{(0)} - (B_2^{-1} \otimes B_1^{-1})\right] q. \quad (14)
\end{aligned}$$

根据引理3.2中Kronecker积的性质3, 由(13)定义的迭代矩阵D的谱集可表示为

$$\lambda(D)=\{\lambda_i^r+\mu_j^s-\lambda_i^r\mu_j^s|\lambda_i\in\lambda(G_1),\mu_j\in\lambda(G_2),i=0,1,\cdots,m+1,j=0,1,\cdots,n+1\}.$$

由于

$$\left|\lambda_i^r+\mu_j^s-\lambda_i^r\mu_j^s\right|\leq\rho^r(G_1)+\rho^s(G_2)+\rho^r(G_1)\rho^s(G_2),$$

因此, 迭代矩阵D的谱半径满足

$$\rho(D)\leq\rho^r(G_1)+\rho^s(G_2)+\rho^r(G_1)\rho^s(G_2).$$

根据均值不等式, 有

$$\rho(D)\leq\rho^r(G_1)+\rho^s(G_2)+\frac{\left[\rho^r(G_1)\right]^2+\left[\rho^s(G_2)\right]^2}{2}=\frac{\left[\rho^r(G_1)+1\right]^2+\left[\rho^s(G_2)+1\right]^2}{2}-1,$$

又矩阵G_1和G_2的谱半径满足(7), 结合上式, 可知D的谱半径满足$\rho(D)<1$, 且

$$\lim_{k\to\infty}D^{k+1}=O,$$

其中O是$(m+2)(n+2)$阶零矩阵. 这意味着当$k\to\infty$时, (14)的极限是

$$p^{(\infty)}=(B_2^{-1}\otimes B_1^{-1})q,$$

其矩阵形式就是

$$P^{(\infty)}=B_1^{-1}Q(B_2^{\mathrm{T}})^{-1}.$$

这意味着$P^{(\infty)}$就是矩阵方程$B_1XB_2^{\mathrm{T}}=Q$的解. 故由迭代格式(8)生成的曲面序列$\{S^{(k)}(u,v),k=0,1,\cdots\}$是收敛的, 且它的极限曲面插值于数据点$\{Q_{ij}\}_{i=1,j=1}^{m,n}$.

§4 实例分析

本节将用4个数值实例来验证本文提出的GSTS-PIA方法的有效性, 并与经典PIA[8]、Jacobi-PIA(JPIA)[14]、加权PIA(WPIA)[17]和GS-PIA[31-32]这四种方法进行比较. 所有数值实验在安装Intel CPU处理器(1.80GHz, 8GB RAM, i5 64-bit)的PC端上运行Matlab R2019a实现. 由于B样条基函数和Bernstein基函数都是NTP基[33], 用双三次B样条曲面和张量积Bézier曲面去拟合这些数据点. 定义第k次迭代曲面的拟合误差为[12]

$$\varepsilon_k=\max\left\{\left\|\delta_{ij}^{(k)}\right\|,i=0,1,\cdots,m+1,j=0,1,\cdots,n+1\right\},$$

其中$\delta_{ij}^{(k)}$为迭代格式(8)中$\delta^{(k)}$的分量形式，且范数为欧几里得范数.

本文扩充点的参数和节点向量取法类似于JPIA算法的计算方法[14]. 首先根据累加弦长法[34], 计算每个数据点Q_{ij}的参数值u_i和v_j得到

$$u_i=\frac{1}{n}\sum_{j=1}^{n}t_{i,j},\ v_j=\frac{1}{m}\sum_{i=1}^{m}t_{i,j},\ i=1,\cdots,m,\ j=1,\cdots,n,$$

其中

$$t_{1,j}=0,\ t_{i,j}=t_{i-1,j}+\frac{\|Q_{ij}-Q_{i-1,j}\|}{\sum_{i=2}^{m}\|Q_{ij}-Q_{i-1,j}\|},\ i=2,\cdots,m,\ j=1,\cdots,n.$$

为了得到边界插值的曲面, 利用多重节点法[14]计算节点向量为$U=\{\bar{u}_i\}_{i=0}^{m+5}$, 其中

$$\bar{u}_{i+2}=u_i,\ i=1,2,\cdots,m,\tag{15}$$

且$\bar{u}_0=\bar{u}_1=\bar{u}_2=\bar{u}_3$, $\bar{u}_{m+2}=\bar{u}_{m+3}=\bar{u}_{m+4}=\bar{u}_{m+5}$. 节点向量$V=\{\bar{v}_j\}_{j=0}^{n+5}$可类似得到.

例4.1 用4×3次Bézier曲面拟合5×4个数据点(1,1,1), (1,2,4), (1,3,2), (1,4,2), (1,5,2), (2,1,2), (2,2,2), (2,3,3), (2,4,6), (2,5,4), (3,1,3), (3,2,2), (3,3,4), (3,4,4), (3,5,3), (4,1,4), (4,2,6), (4,3,1), (4,4,1), (4,5,2)[14].

例4.2 用双三次B样条曲面拟合Ear-cut模型上采样的21×21个网格数据点[12].

例4.3 用双三次B样条曲面拟合从一张Tranguloid-trefoil曲面上均匀采样得到的21×31个网格数据点, 其参数方程为[12]

$$\begin{cases} x = 2\sin(3t)/(2+\cos s), (t, s \in [-\pi, \pi]), \\ y = 2(\sin t + 2\sin(2t))/(2+\cos(s+2\pi/3\ )), \\ z = (\cos t - 2\cos(2t))(2+\cos s)(2+\cos(s+2\pi/3))/4. \end{cases}$$

例4.4 用双三次B样条曲面拟合Mannequin模型上采样的161×121个网格数据点[10].

例4.1中Bernstein基函数是NTP基[33],由于4次Bernstein基函数是

$$B_i^4(u) = C_4^i u^i (1-u)^{4-i}, i = 0, 1, \cdots, 4,$$

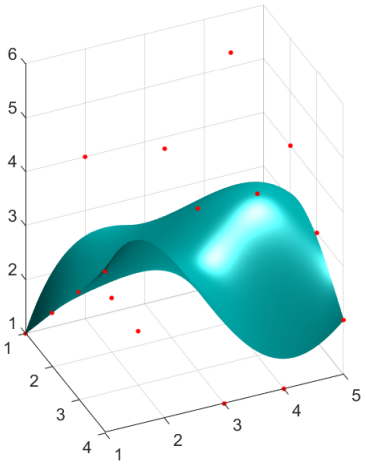
且$[u_2, u_4] \in (0, 1)$, 则(1)定义的配置矩阵B_1中元素$B_i(u_i) = B_{i-1}^4(u_i) > 0, i = 1, \cdots, 5$, 即$B_1$的主对角线元素皆大于0. 同理, 可证(1)中的矩阵B_2的主对角线元素皆大于0. 选取$r = s = 2$时, 条件(7)满足. 此时, 例4.1符合定理3.1的收敛条件.

例4.2-例4.4中三次B样条基函数也是NTP基[33], 此时根据(15), 配置矩阵B_1的主对角线元素是$1, B_1(\bar{u}_3), B_2(\bar{u}_4), \cdots, B_m(\bar{u}_{m+2}), 1$. 根据B样条基函数的局部支撑性和非负性, 其主对角线元素$B_i(\bar{u}_{i+2}) > 0,\ i = 1, \cdots, m$. 同理, (1)中的配置矩阵$B_2$的主对角线元素也都为正. 选取参数$r = s = 1$时, 条件(7)满足, GSTS-PIA方法退化为GS-PIA方法. 此时, 例4.2-4.4符合定理3.1的收敛条件.

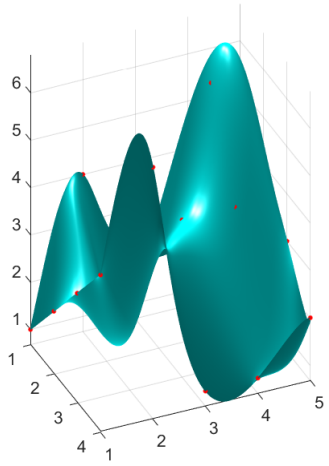
用GSTS-PIA算法对例4.1-例4.4的数据点进行拟合的结果如图1-图4所示, 其中图(a)是初始迭代曲面, 圆点表示原始数据点, 图(b)是迭代10次的拟合曲面, 图(c)是达到代数精度1×10^{-12}的极限拟合曲面, 图(d)是四种迭代方法在达到代数精度1×10^{-12}所需的迭代次数与拟合误差的关系比较图, 拟合误差用对数尺度绘制. 从图1(d)-图4(d)可以明显看出, GSTS-PIA算法所需的迭代次数要明显少于其他三种方法. 当用双三次B样条曲面拟合时, JPIA和WPIA方法所需的迭代次数相当.

表1列出了不同迭代次数下四种方法的拟合误差, 例4.1-例4.4的初始误差ε_0分别为2.4893×10^0 , 1.8916×10^{-2}, 2.9175×10^{-1}, 6.1956×10^{-3}. 在相同的迭代次数下, GSTS-PIA方法的拟合误差最小, 比其他三种方法的拟合误差要小至少一个数量级. 表2给出了在不同误差精度下, 用PIA, JPIA, WPIA, GS-PIA和GSTS-PIA方法拟合例4.1-例4.4的数据点所需的迭代次数与运行时间(运行10次取平均). 可以看出, 在误差达到相同精度下, GSTS-PIA方法所需的迭代次数最少, 并且运行时间最短. 由于例4.2-例4.4中GSTS-PIA方法退化为GS-PIA方法, 因此两种方法的迭代次数和拟合误差相同. 当误差精度达到1×10^{-7}时, 例4.1中本文方法所需的运行时间仅为经典PIA方法的3.36%, JPIA方法的12.66%, WPIA方法的6.45%, GS-PIA方法的76.92%. 当误差精度达到1×10^{-12}时, 例4.2-例4.4中本文方法所需的运行时间仅为经典PIA方法的0.53% ∼ 34.21%, JPIA方法的34.71% ∼ 44.39%, WPIA方法的0.93% ∼ 45.05%, GS-PIA方法的1.07% ∼ 68.12%. 进一步, 为了探究参数r和s对算法的影响, 表3给出了在误差精度为1×10^{-12}时, GSTS-PIA方法在不同的参数r和s下所需的迭代次数与运行时间(运行10次取平均). 可以看出, 随着r和s的增大, 迭代次数减少, 但运行时间并未持续缩短, 说明参数r和s并非越大越好. 这是由于参数r和s增大虽然加快了收敛速度, 但是其迭代格式更加复杂, 导致单次

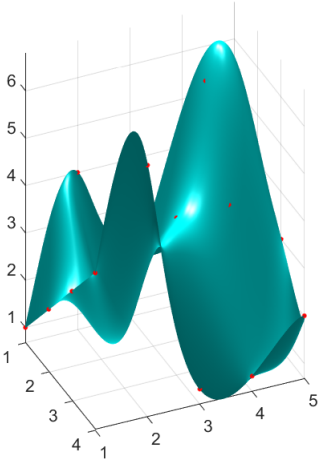
迭代的计算量增加.

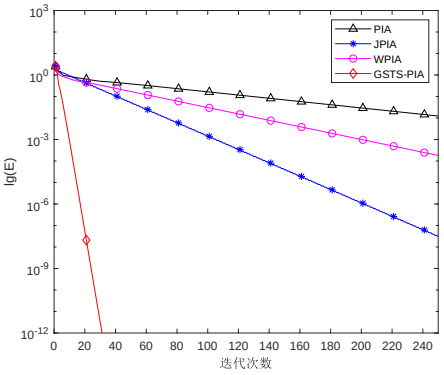


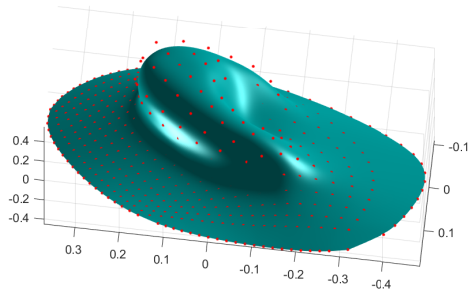
(a) 第0次迭代曲面

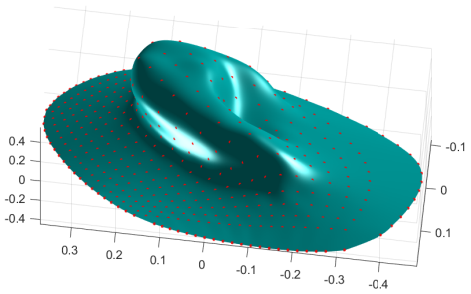


(b) 第10次迭代曲面



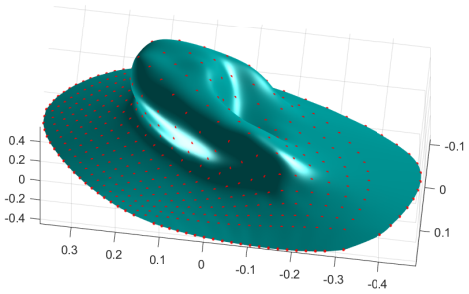




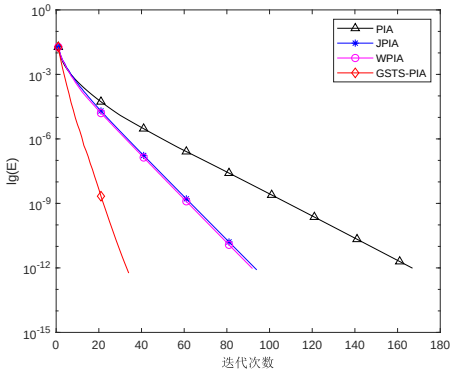


(a) 第0次迭代曲面

(b) 第10次迭代曲面

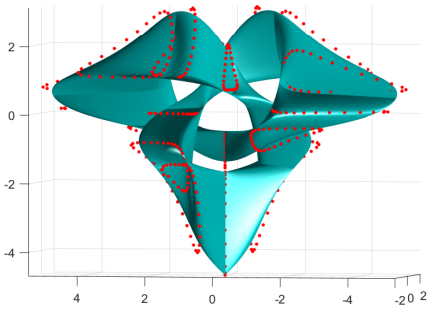


(c) 第33次迭代曲面

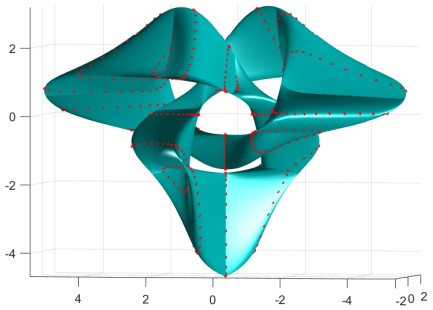


(d) 迭代次数与误差关系图

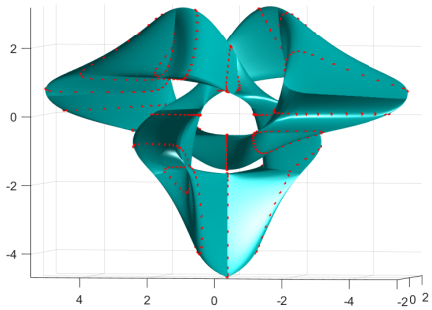
图 2 用GSTS-PIA算法拟合例4.2数据点的迭代曲面和误差曲线比较图



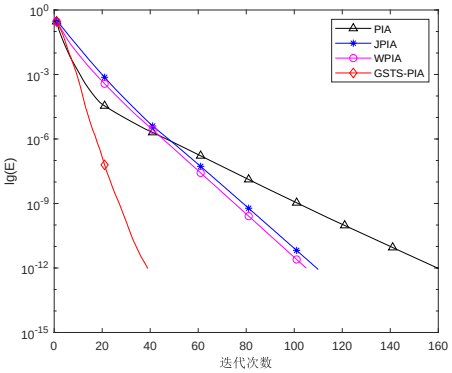
(a) 第0次迭代曲面



(b) 第10次迭代曲面

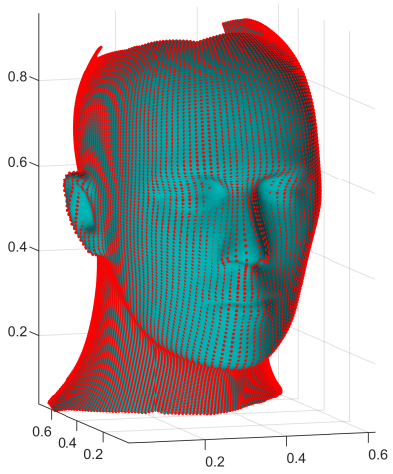


(c) 第38次迭代曲面

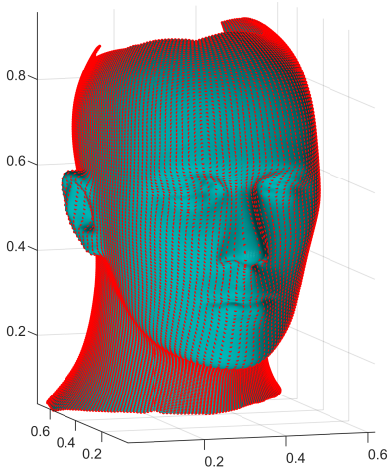


(d) 迭代次数与误差关系图

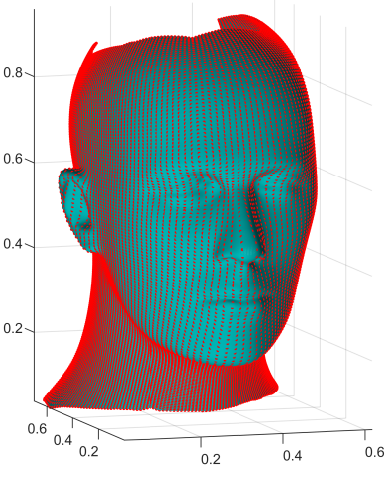
图 3 用GSTS-PIA算法拟合例4.3数据点的迭代曲面和误差曲线比较图



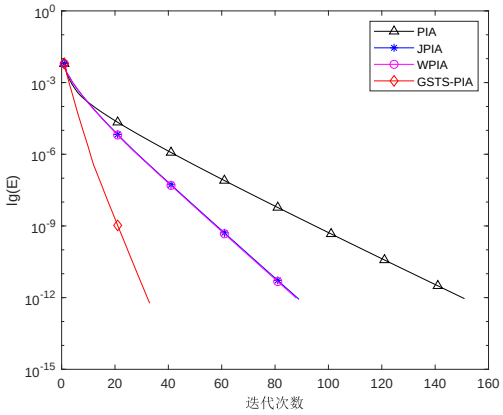
(a) 第0次迭代曲面



(b) 第10次迭代曲面



(c) 第32次迭代曲面



(d) 迭代次数与误差关系图

图 4 用GSTS-PIA算法拟合例4.4数据点的迭代曲面和误差曲线比较图

表 1 不同迭代次数下例4.1-例4.4的迭代拟合误差

实例	方法	ε_{10}	ε_{30}	ε_{50}
例4.1	PIA	8.3882×10^{-1}	5.0690×10^{-1}	3.7855×10^{-1}
	JPIA	9.0212×10^{-1}	2.1191×10^{-1}	5.0446×10^{-2}
	WPIA	6.3269×10^{-1}	3.2412×10^{-1}	1.6439×10^{-1}
	GSTS-PIA	3.3778×10^{-4}	1.2285×10^{-12}	6.2804×10^{-16}
例4.2	PIA	3.6666×10^{-4}	1.0950×10^{-5}	8.5431×10^{-7}
	JPIA	2.8170×10^{-4}	1.7806×10^{-6}	1.6578×10^{-8}
	WPIA	2.2191×10^{-4}	1.4490×10^{-6}	1.2985×10^{-8}
	GSTS-PIA	2.5156×10^{-6}	3.2528×10^{-12}	7.8505×10^{-17}
例4.3	PIA	8.9082×10^{-4}	7.6092×10^{-6}	5.8546×10^{-7}
	JPIA	1.4456×10^{-2}	4.9482×10^{-5}	4.8054×10^{-7}
	WPIA	6.5810×10^{-3}	2.8150×10^{-5}	2.6798×10^{-7}
	GSTS-PIA	2.4560×10^{-4}	7.0610×10^{-11}	1.0175×10^{-15}
例4.4	PIA	1.2114×10^{-4}	4.8749×10^{-6}	3.0211×10^{-7}
	JPIA	1.0672×10^{-4}	5.5585×10^{-7}	5.1799×10^{-9}
	WPIA	1.0365×10^{-4}	5.2579×10^{-7}	4.7776×10^{-9}
	GSTS-PIA	8.0940×10^{-7}	1.9992×10^{-12}	3.3537×10^{-16}

表 2 在不同误差精度下曲面拟合所需的迭代次数与运行时间比较

实例	方法	1×10^{-3}		1×10^{-5}		1×10^{-7}	
		次数	时间(s)	次数	时间(s)	次数	时间(s)
例4.1	PIA	395	0.0121	662	0.0191	929	0.0298
	JPIA	105	0.0054	169	0.0060	233	0.0079
	WPIA	200	0.0073	334	0.0104	469	0.0155
	GS-PIA	25	0.0007	38	0.0008	51	0.0013
	GSTS-PIA	9	0.0006	14	0.0007	19	0.0010
		1×10^{-7}		1×10^{-10}		1×10^{-12}	
		次数	时间(s)	次数	时间(s)	次数	时间(s)
例4.2	PIA	69	0.0105	128	0.0175	166	0.0210
	JPIA	43	0.0062	73	0.0103	93	0.0133
	WPIA	42	0.0064	71	0.0100	91	0.0124
	GS-PIA	15	0.0042	25	0.0057	33	0.0069
	GSTS-PIA	15	0.0023	25	0.0038	33	0.0047
例4.3	PIA	64	0.0168	120	0.0234	159	0.0266
	JPIA	58	0.0117	88	0.0188	109	0.0205
	WPIA	55	0.0152	85	0.0164	104	0.0202
	GS-PIA	20	0.0103	30	0.0150	38	0.0195
	GSTS-PIA	20	0.0050	30	0.0087	38	0.0091
例4.4	PIA	59	13.6659	113	25.6983	150	36.4984
	JPIA	38	0.2631	68	0.4310	88	0.5558
	WPIA	38	9.4997	67	15.9326	87	20.8493
	GS-PIA	13	8.4838	24	13.8064	32	18.0154
	GSTS-PIA	13	0.1132	24	0.1638	32	0.1929

表 3 在不同r, s下GSTS-PIA方法所需迭代次数与运行时间比较

r, s	例4.2		例4.3		例4.4	
	次数	时间(ms)	次数	时间(ms)	次数	时间(ms)
1,1	33	6.4394	38	10.0155	32	217.6481
1,2	22	5.2158	25	7.5891	21	165.3351
2,2	13	3.8350	14	5.4056	12	124.9828
2,3	10	3.3116	12	6.7277	10	111.4896
3,3	8	3.0477	9	4.2983	7	89.4204
3,4	7	2.8847	8	6.1720	7	94.3524
4,4	6	2.8135	6	3.6860	5	84.3101
4,5	5	2.5974	6	4.4231	5	96.9931
5,5	5	2.6835	5	4.2302	4	88.0025
5,6	4	2.5168	5	3.2287	4	81.3321
6,6	4	2.5441	4	3.2467	4	83.5687
6,7	4	2.6323	4	3.6730	3	75.8459
7,7	3	2.3876	4	3.8189	3	76.3972
7,8	3	2.3414	4	4.0158	3	78.6261
8,8	3	2.3636	3	3.1334	3	77.1841

§5 结语

本文将矩阵方程的改进Gauss-Seidel分裂方法与经典PIA算法相结合, 提出了混合曲面的GSTS-PIA方法, 并证明了该方法生成的曲面序列收敛到的极限曲面插值于原始数据点. 数值实例表明本文方法加快了收敛速度, 提高了计算效率. 后续将尝试应用于细分曲面、曲面光顺、隐式曲面重建等领域. 尽管本文的数值实验验证了随着参数r和s的增大, GSTS-PIA方法的所需迭代次数会减少, 但是每次迭代的计算量会增加. 如何找到最优的r和s使得算法的CPU执行时间最少是一个值得深入探讨的问题.

参考文献:

[1] Lin Hongwei, Maekawa T, Deng Chongyang. Survey on geometric iterative methods and their applications[J]. Comput Aided Design, 2018, 95: 40-51.

[2] 蔺宏伟. 几何迭代法及其应用[M]. 杭州: 浙江大学出版社, 2021.

[3] Ebadi M J, Ebrahimi A. Video data decompression by progressive iterative approximation[J]. Int J Interact Multi, 2021, 6(6): 189-196.

[4] Zhang Yunfeng, Wang Ping, Bao Fangxun, et al. A single image super-resolution method based on progressive-iterative approximation[J]. IEEE T Multimedia, 2019, 22(6): 1407-1422.

[5] 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.

[6] de Boor C. How does Agee's smoothing method work?[R]. Washington D C: Army Research Office, 1979, 299-302.

[7] Lin Hongwei, Wang Guojin, Dong Chenshi. Constructing iterative non-uniform B-spline curve and surface to fit data points[J]. Sci China, Ser F, 2004, 47: 315-331.
[8] Lin Hongwei, Bao Hujun, Wang Guojin. Totally positive bases and progressive iteration approximation[J]. Comput Math Appl, 2005, 50(3-4): 575-586.
[9] Liu Mingzeng, Li Baojun, Guo Qingjie, et al. Progressive iterative approximation for regularized least square bivariate B-spline surface fitting[J]. J Comput Appl Math, 2018, 327: 175-187.
[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] Wang Huidi. Implicit randomized progressive-iterative approximation for curve and surface reconstruction[J]. Comput Aided Design, 2022, 152: 103376.
[12] 胡倩倩, 梁如意, 王国瑾. 一类快速收敛的渐进迭代逼近方法[J]. 计算机辅助设计与图形学学报, 2023, 35(12): 1900-1909.
[13] 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(3): 485-491.
[14] Liu Chengzhi, Li Juncheng, Hu Lijuan. Jacobi-PIA algorithm for bi-cubic B-spline interpolation surfaces[J]. Graph Model, 2022, 120: 1524-0703.
[15] Wang Zhihao, Li Yajuan, Liu Jianzhen, et al. Gauss-Seidel progressive iterative approximation (GS-PIA) for subdivision surface interpolation[J]. Visual Comput, 2021, 39(1):1-10.
[16] 王志好, 李亚娟, 邓重阳. GS-PIA算法的收敛性证明[J]. 计算机辅助设计与图形学学报, 2018, 30(11): 2035-2041.
[17] Lu Lizheng. Weighted progressive iteration approximation and convergence analysis[J]. Comput Aided Geom Design, 2010, 27(2): 129-137.
[18] Lin Hongwei. Local progressive-iterative approximation format for blending curves and patches[J]. Comput Aided Geom Design, 2010, 27(4): 322-339.
[19] Deng Chongyang, Lin Hongwei. Progressive and iterative approximation for least squares B-spline curve and surface fitting[J]. Comput Aided Design, 2014, 47: 32-44.
[20] Hamza Y F, Lin Hongwei, Li Zihao. Implicit progressive-iterative approximation for curve and surface reconstruction[J]. Comput Aided Geom Design, 2020, 77(3): 101817.
[21] Chen Zhongxian, Luo Xiaonan, Tan Le, et al. Progressive interpolation based on Catmull-Clark subdivision surfaces[J]. Comput Graph Forum, 2008, 27(7): 1823-1827.
[22] Jiang Yini, Lin Hongwei, Huang Weixian. Fairing-PIA: Progressive-iterative approximation for fairing curve and surface generation[J]. Visual Comput, 2023, 40(3): 1467-1484.
[23] Carnicer J M, Delgado J, Peña J M. On the progressive iteration approximation property and alternative iterations[J]. Comput Aided Geom Design, 2011, 28(9): 523-526.
[24] Saad Y. Iterative Methods for Sparse Linear Systems[M]. 2nd ed. Philadelphia: Society for Industrial and Applied Mathematics, 2003.
[25] Hu Liangchen, Shou Huahao, Dai Zhenlei. HSS-iteration-based iterative interpolation of curves and surfaces with NTP bases[J]. Wirel Netw, 2021, 27: 3523-3535.
[26] Liu Zhongyun, Li Zhen, Ferreira C, et al. Stationary splitting iterative methods for the matrix equation $AXB = C$[J]. Appl Math Comput, 2020, 378: 125195.
[27] Tian Zhaolu, Tian Maoyi, Liu Zhongyun, et al. The Jacobi and Gauss-Seidel-type iteration methods for the matrix equation $AXB = C$[J]. Appl Math Comput, 2017, 292: 63-75.

[28] Lanzkron P J, Rose D J, Syzld D B. Convergence of nested classical iterative methods for linear systems[J]. Numer Math, 1991, 58: 685-702.
[29] Brewer J W. Kronecker products and matrix calculus in system theory[J]. IEEE Trans Circuits Syst, 1978, 25(9): 772-781.
[30] Liu Zhongyun, Zhou Yang, Zhang Yuelan, et al. Some remarks on Jacobi and Gauss-Seidel-type iteration methods for the matrix equation $AXB = C$[J]. Appl Math Comput, 2019, 354: 305-307.
[31] 刘晓艳, 邓重阳. 非均匀三次B样条曲线插值的GS-PIA算法[J]. 杭州电子科技大学学报(自然科学版), 2015, 35(2): 79-82.
[32] 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.
[33] Carnicer J M, Peña J M. Totally positive bases for shape preserving curve design and optimality of B-splines[J]. Comput Aided Geom Design, 1994, 11: 633-654.
[34] Piegl L, Tiller W. The NURBS Book[M]. 2nd ed. New York: Springer-Verlag, 1997.

Improved GS-PIA algorithm for blending surfaces

HU Qian-qian[1], DONG Wen-qing[1], YAO Zhen-min[1], WANG Guo-jin[2]
(1. School of Statistics and Mathematics, Zhejiang Gongshang University, Hangzhou 310018, China;
2. College of Mathematics, Zhejiang University, Hangzhou 310027, China)

Abstract: The PIA method for surface fitting usually transforms surface fitting into curve fitting by vertically stacking the columns of matrix. It needs to calculate the Kronecker product of the matrix, which has the disadvantages of large computation and long running-time. In order to avoid the calculation of the Kronecker product, a new PIA algorithm for blending surfaces is proposed based on Gauss-Seidel type splitting (GSTS-PIA), which combines the improved Gauss-Seidel method for solving matrix equation with the classical PIA algorithm. First, the difference vector is calculated for each fitting data point. Then, the difference vector for each control point is calculated according to the Gauss-Seidel type splitting matrix. Finally, new control points are obtained by the difference vectors, and a surface for fitting the data points is generated. Theoretical analysis proves that the limit of the surface sequence interpolates the given data points. The experimental results of fitting scattered data points or sampling points from regular surfaces with different blending surfaces demonstrate that compared with the classical PIA algorithm, the GSTS-PIA algorithm requires an average reduction of 78.30% in the number of iterations and an average reduction of 80.96% in running-time under the same fitting accuracy.

Keywords: progressive iterative approximation; data fitting; convergence rate; blending surfaces
MR Subject Classification: 65D18; 68U07