Advances in Applied Mathematics 应用数学进展, **2023, 12(11), 4806-4813**
Published Online November 2023 in Hans. https://www.hanspub.org/journal/aam
https://doi.org/10.12677/aam.2023.1211474

Hans汉斯

关于最小二乘拟合的Succesive Over Relaxation渐进迭代逼近

田 沂[1*]，**杜勇奇**[2]

[1]长沙理工大学数学与统计学院，湖南 长沙
[2]哈尔滨工程大学数学科学学院，黑龙江 哈尔滨

收稿日期：2023年10月23日；录用日期：2023年11月17日；发布日期：2023年11月24日

摘 要

本文以Guass-Seidel progressive iterative approximation for least squares fitting (LSPIA)算法为基础，提出一种基于Succesive Over Relaxation (SOR) 迭代的LSPIA算法，简称SOR-LSPIA。我们分析了SOR-LSPIA算法的收敛性，数值实验表明，当拟合精度相同时，SOR-LSPIA算法比GS-LSPIA算法迭代步数更少、运行时间更短。

关键词

渐进迭代逼近，Guass-Seidel迭代法，Succesive Over Relaxation迭代法，曲线逼近

The Succesive Over Relaxation Progressive Iterative Approximation for Least Squares Fitting

Yi Tian[1*], **Yongqi Du**[2]

[1]School of Mathematics and Statistics, Changsha University of Science and Technology, Changsha Hunan
[2]College of Mathematical Sciences, Harbin Engineering University, Harbin Heilongjiang

Received: Oct. 23^{rd}, 2023; accepted: Nov. 17^{th}, 2023; published: Nov. 24^{th}, 2023

* 通讯作者

文章引用: 田沂, 杜勇奇. 关于最小二乘拟合的Succesive Over Relaxation渐进迭代逼近[J]. 应用数学进展, 2023, 12(11): 4806-4813. DOI: 10.12677/aam.2023.1211474

Abstract

Based on the Guass-Seidel progressive iterative approximation for least squares fitting (LSPIA) algorithm, a Succesive Over Relaxation LSPIA (SOR-LSPIA) algorithm is proposed in this paper. We analysis the convergence of this.Furthermore some numerical is tests experiments are shown, our algorithm has fewer number iteration steps and shorter cpu time than the GS-LSPIA algorithm does if the fitting accracies are requared the same.

Keywords

Progressive-Iterative Approximation, Guass-Seidel Iterative Method, Succesive Over Relaxation Iterative Method, Curve Fitting

1. 引言

几何迭代法[1],又名渐进迭代逼近(progressive-iterative approximation, PIA)，是一种有效的数据拟合方法.因为迭代法具有数值稳定, 有明确的几何意义和简单的迭代格式等性质, 吸引了广大学者的关注. 特别地, 几何迭代法被广泛的应用在CAGD、数据拟合、逆向工程等方面[1, 2].

几何迭代法始于1957年齐东旭[3]学者等提出的均匀三次B样条曲线的盈亏修正算法. 1979年, de Boor [4]证明了该算法的收敛性. 2004年, Lin [5]等给出非均匀三次B样条曲线插值的收敛方式和收敛证明. 2005年, Lin [6]等证明所有全正基混合曲线曲面都具有该性质,并创造了英文术语progressive-iterative approximation (PIA: 渐进迭代逼近)来描述该方法. 在PIA算法中, 每个数据点所对应的参数值在迭代过程中是不会改变的.

近年来, 有许多学者对PIA算法进行改进. 在经典PIA算法中, 控制点个数等于数据点个数. Lin [7]等设计了一种扩展的渐进迭代逼近(extended progressive and iterative approximation for least squares B-spline curve and surface fitting,EPIA)算法来拟合大规模数据点, 它允许控制点个数小于数据点个数. Deng [8]等提出最小二乘渐进迭代逼近算法, 使得拟合的极限曲线曲面是对给定点集的最小二乘拟合结果. 对于广义B样条基, Zhang [9, 10] 等提出了一种具有不同权重的LSPIA算法来拟合更复杂的数据点. PIA算法的局部特征也被广泛用于数据拟合领域. Lin [11]等提出了一种适用于拟合大规模数据的T样条LSPIA算法, 实现在微机上快速拟合超过几千万像素的高精度图像. Lin [12]等提出了基于Gauss-Seidel迭代法的快速PIA算法，解决了经典LSPIA算法中收敛速度较慢的问题. Wu [13]等人也提出异步LSPIA方法来拟合数据点, 结果表明在奇异和非奇异的最小二乘拟合情况下, 异步LSPIA比LSPIA快.

2. 经典的LSPIA算法

设$\{Q_j\}_{j=0}^m$是待拟合的有序点集, t_i是每个点Q_j的参数值, 且满足$0 = t_0 < t_1 < \cdots < t_m) = 1$,迭代开始前, 从给定点集$\{Q_j\}_{j=0}^m$里选取$n$个点作为初始控制点列$\{\mathbf{P}_i^0\}_{i=0}^n$, 构造一条初始曲线

$$P^{(0)}(t) = \sum_{i=0}^{n} \mathbf{B}_i(t)P_i^{(0)}, \quad t \in [0,1] \qquad \delta_j^0 = Q_j - P^0(t_j) \quad j = 0,1,\cdots,m$$

取第i个控制点的第一个调整向量为

$$\Delta_i^0 = \sum_{j=0}^{m} \mathbf{B}_i(t_j)\delta_j^0$$

得到新的控制点:

$$P_i^1 = P_0^k + \Delta_i^k \quad i = 0,1,\cdots,n$$

得到新曲线

$$P^{(1)}(t) = \sum_{i=0}^{n} \mathbf{B}_i(t)P_i^{(1)}$$

类似地, 在第k次迭代后得到了第k条曲线为

$$P^{(k)}(t) = \sum_{i=0}^{n} \mathbf{B}_i(t)P_i^{(k)}, \quad t \in [0,1]$$

为进行第$k+1$ 次迭代, 首先计算校正向量

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

$$\Delta_i^k = \sum_{j=0}^{m} \mathbf{B}_i(t_j)\delta_j^k \quad i = 0,1,\cdots,n$$

则新的控制点为

$$P_i^{k+1} = P_i^k + \Delta_i^k \quad i = 0,1,\cdots,n \tag{2.1}$$

进而得到第$k+1$条曲线为

$$P^{(k+1)}(t) = \sum_{i=0}^{n} \mathbf{B}_i(t)P_i^{(k+1)}$$

将上式方程(2.1)用矩阵形式表示如下:

$$P^{(k+1)} = P^{(k)} + B^T(Q - BP^k) \tag{2.2}$$

其中$\{B_i(t)\}_{i=0}^n$是全正基函数，在t_i上的配置矩阵为:

$$B = \begin{bmatrix} B_0(t_0) & B_1(t_0) & \cdots & B_n(t_0) \\ B_0(t_1) & B_1(t_1) & \cdots & B_n(t_1) \\ \vdots & \vdots & & \vdots \\ B_0(t_m) & B_1(t_m) & \cdots & B_n(t_m) \end{bmatrix} \tag{2.3}$$

为了加快(2.1)的收敛速度,文献[8]提出了WLSPIA算法,其矩阵形式为:

$$P^{(k+1)} = P^{(k)} + \omega B^T(Q - BP^k) \tag{2.4}$$

为进一步改进WLSPIA算法的收敛速度,文献[12]将Guass-Seidel迭代法与经典的LSPIA方法结合起来,即GS-LSPIA算法,其矩阵形式为:

$$P^{(k+1)} = P^{(k)} + (D - L)^{-1}B^T(Q - BP^k) \tag{2.5}$$

其中矩阵$(D-L)$是矩阵B^TB的严格下三角部分.

为进一步加速,提出SOR-LSPIA算法,其矩阵形式为:

$$P^{(k+1)} = P^{(k)} + (\frac{1}{v}D - L)^{-1}B^T(Q - BP^k) \tag{2.6}$$

3. SOR-LSPIA算法及其收敛性

作者蔺在文献[12]中将Guass-Seidel迭代法与经典的LSPIA方法结合起来,提出了GS-LSPIA方法,其中GS-LSPIA曲线拟合算法可改写为:

$$P^{k+1} = D^{-1}(B^TQ - LP^{k+1} - UP^k) = -(L+D)^{-1}UP^k + (L+D)^{-1}B^TQ \tag{3.7}$$

上式中, B是配置矩阵式(2.3);矩阵D、L、U分别是矩阵$A = B^TB$的严格对角、下三角、上三角矩阵,即$A = B^TB = D + L + U$.

为了进一步加快其收敛速度, 我们提出WSOR-LSPIA方法, 其迭代格式如下:

$$P^{k+1} = P^k + \omega(\frac{1}{v}D - L)^{-1}(B^TQ - B^TBP^k) \tag{3.8}$$

我们回顾一下解一般的线性方程组$AX = b$, 由文献[14]可知,

SOR的迭代格式为:

$$x_i^{(k+1)} = (1-\omega)x_i^{(k)} + \frac{\omega}{a_{ii}}\left\{b_i - \sum_{j=1}^{i-1} a_{ij}x_j^{k+1} - \sum_{j=i+1}^{n} a_{ij}x_j^{(k)}\right\}. \tag{3.9}$$

因此, 当$\omega = 1$时, SOR迭代等价于GS迭代; 则GS收敛, SOR也收敛.

下面将讨论SOR-LSPIA算法的收敛性, 有以下定理:

引理 1.[14, 15]若矩阵A对称正定, 则SOR迭代在$0 < \omega < 2$收敛.

引理 2.[14] 1.若矩阵A有相容次序且$\omega \neq 0$,μ是R_J的一个特征值出现且$(\lambda + \omega - 1)^2 = \lambda\omega^2\mu^2$

2.假设矩阵A具有相容次序, R_J有实特征值, 且$\mu = \rho(R_J) < 1$.则

$$\omega_{opt} = \frac{2}{1 + \sqrt{1-\mu^2}}. \tag{3.10}$$

$$\rho(R_{SOR(\omega_{opt})}) = \omega_{opt} - 1 = \frac{\mu^2}{[1 + \sqrt{1-\mu^2}]^2}$$

其中, R_J是$Jacobi$迭代, μ是$Jacobi$迭代矩阵的谱半径.

接下来表述我们的主要结果：

定理 1. 当基函数是非均匀三次B样条基时,当ω在(0,2)时,SOR-LSPIA迭代(3.8)是收敛的.

证明：因为基函数$\mathbf{B_i}(\mathbf{t})$是全正基函数，在参数列$0 = t_0 < t_1 < \cdots < t_m = 1$上生成的配置矩阵$B$是$Toeplitz$矩阵且列满秩，所以矩阵$B^TB$是对称正定的.由引理1、2可知,SOR-LSPIA迭代(3.8)是收敛的.

4. 数值实验

4.1. 算法实现

给定有序点集$\{Q_j\}_{j=0}^m$,使用弦长累加法分配参数$\{t_i, i = 0, 1, \cdots, m\}$即

$$\begin{cases} t_0 = 0, t_m = 1 \\ t_i = t_{i-1} + \frac{|q_i - q_{i-1}|}{L}, i = 1, 2, \cdots, m-1 \end{cases} \tag{4.11}$$

DOI: 10.12677/aam.2023.1211474

其中, $L=\sum_{i=0}^{m}\|q_i-q_{i-1}\|$是总弦长.$P(t)=\sum_{i=0}^{n}p_iB_{i,3}(t)$是B样条曲线, $B_{i,3}$是B样条基函数, 定义在节点向量$\{0,0,0,0,u_4,u_5,\cdots,u_n,1,1,1,1\}$上.节点向量满足

$$\begin{aligned} &u_{j+3}=(1-\alpha)t_{i-1}+\alpha t_i, j=1,\cdots,n-3,\\ &i=[jd],\alpha=jd-i,d=\tfrac{m+1}{n-2}. \end{aligned} \tag{4.12}$$

其中, $[jd]$为不大于jd的最大整数.另外, 选取初始控制点$P=\{p_i,i=0,1,\cdots,n\}$的规则如下:

$$\begin{cases} p_0=q_0\\ p_n=q_m\\ p_i=q_{f(i)}, i=1,2,\cdots,n-i \end{cases} \tag{4.13}$$

其中,$f(i)=[\frac{(m+1)i}{n}]$.

4.2. 数值实例

在本小节中, 我们将给出几个具体算例来验证SOR-LSPIA算法的有效性.首先定义算法在第k次迭代的拟合误差为:

$$E_k=\max_j\{e_j^k\}, e_j^k=\left\|q_j-\sum_{i=0}^{n}B_i(t_j)p_i^k\right\| \tag{4.14}$$

关于ω的选取, 按照引理5的公式计算出的ω_{opt}是具有相容性的,但在我们的例子中，矩阵是不具备有相容性.通过大量数值实验发现, 当我们取到$\omega=1.2$时, 效果是比较好的.具体实例如下:

例 1. 球心的参数方程为

$$\begin{cases} x=2cos(t)-cos(3t)\\ y=2sin(t)-sin(3t)\\ z=2cos(t/2) \end{cases}$$

其中, $t\in[0,4\pi]$. 从球心曲线上采样得到10000个数据点, 从中分别选取了3000、4000、5000个点作为控制点, 其迭代步数、运算时间以及迭代误差如表 1所示.

Table 1. The two algorithms interpolated the numerical results of the sampling points on the spherical line

表 1. 两种算法插值球心线上采样点的数值结果

	(10000,3000)		(10000,4000)		(10000,5000)	
(m,n)	GS-LSPIA	SOR-LSPIA	GS-LSPIA	SOR-LSPIA	GS-LSPIA	SOR-LSPIA
IT	65	45	84	56	79	51
CPU	1.01×10^{-2}	6.4×10^{-3}	1.31×10^{-2}	8.6×10^{-3}	1.76×10^{-2}	1.33×10^{-2}
E_k	9.4254×10^{-11}	7.3714×10^{-11}	8.7129×10^{-11}	9.9039×10^{-11}	9.2878×10^{-11}	9.0873×10^{-11}

例 2. 水滴的参数方程为

$$r=a+bcos(2t+\pi/4)+cos(3t+\pi/4) \qquad 0\le t\le 2\pi$$

其中, $a=2,b=4$. 从水滴曲线上采样得到10000个数据点, 从中分别选取了3000、4000、5000个点作为控制点, 其迭代步数、运算时间以及迭代误差如表 2所示.

Table 2. Numerical results of the sampling points on the water drop curves of the two algorithms

表 2. 两种算法水滴曲线上采样点的数值结果

	(10000,3000)		(10000,4000)		(10000,5000)	
(m,n)	GS-LSPIA	SOR-LSPIA	GS-LSPIA	SOR-LSPIA	GS-LSPIA	SOR-LSPIA
IT	96	58	134	90	418	252
CPU	1.10×10^{-2}	7.2×10^{-3}	1.84×10^{-2}	1.25×10^{-2}	7.55×10^{-2}	5.37×10^{-2}
E_k	8.7110×10^{-11}	8.6870×10^{-11}	9.9395×10^{-11}	9.0053×10^{-11}	9.9887×10^{-11}	9.6325×10^{-11}

例 3. 蝴蝶结的参数方程为

$$\begin{cases} x = cos(t)) \\ y = sint(t)cos(t) \end{cases}$$

其中, $t \in [0, 2\pi]$.在蝴蝶结曲线上采样得到10000个数据点, 从中分别选取了3000、4000、5000个点作为控制点, 其迭代步数、运算时间以及迭代误差如表 3所示.

Table 3. Numerical results of the sampled points on the bow curves of the two algorithms

表 3. 两种算法蝴蝶结曲线上采样点的数值结果

	(10000,3000)		(10000,4000)		(10000,5000)	
(m,n)	GS-LSPIA	SOR-LSPIA	GS-LSPIA	SOR-LSPIA	GS-LSPIA	SOR-LSPIA
IT	80	48	107	73	328	193
CPU	8.10×10^{-3}	4.99×10^{-3}	1.42×10^{-2}	9.7×10^{-3}	6.62×10^{-2}	3.66×10^{-2}
E_k	8.8553×10^{-11}	8.5861×10^{-11}	9.9294×10^{-11}	9.8631×10^{-11}	9.8708×10^{-11}	9.8204×10^{-11}

例 4. 螺旋的参数方程为

$$\begin{cases} x = cos(t))(2 - cos(2t/2k + 1)) \\ y = sin(t))(2 - cos(2t/2k + 1)) \\ z = sin(2t/2k + 1) \end{cases}$$

其中, $k = 2, t \in [0, (4k + 2)\pi]$.在螺旋曲线上采样得到10000个数据点, 从中分别选取了3000、4000、5000个点作为控制点, 其迭代步数、运算时间以及迭代误差如表 4所示.

Table 4. Numerical results of the sampled points on the bow curves of the two algorithms

表 4. 两种算法蝴蝶结曲线上采样点的数值结果

	(10000,3000)		(10000,4000)		(10000,5000)	
(m,n)	GS-LSPIA	SOR-LSPIA	GS-LSPIA	SOR-LSPIA	GS-LSPIA	SOR-LSPIA
IT	96	58	133	90	398	241
CPU	1.45×10^{-2}	9.0×10^{-3}	2.81×10^{-2}	1.79×10^{-2}	9.14×10^{-2}	5.85×10^{-2}
E_k	8.6068×10^{-11}	8.3891×10^{-11}	9.7807×10^{-11}	8.7631×10^{-11}	9.7799×10^{-11}	9.5824×10^{-11}

例 5. 心形的参数方程为

$$\begin{cases} x = (1 - cos(t))sin(t) \\ y = (1 - cos(t))cos(t) \end{cases}$$

其中, $t \in [0, 2\pi]$.在蝴蝶结曲线上采样得到10000个数据点, 从中分别选取了3000、4000、5000个点作为控制点, 其迭代步数、运算时间以及迭代误差如表 5所示.

Table 5. Numerical results of the sampled points on the cardioid curves of the two algorithms
表 5. 两种算法心形曲线上采样点的数值结果

	(10000,3000)		(10000,4000)		(10000,5000)	
(m,n)	GS-LSPIA	SOR-LSPIA	GS-LSPIA	SOR-LSPIA	GS-LSPIA	SOR-LSPIA
IT	58	40	77	52	55	39
CPU	5.3×10^{-3}	3.7×10^{-3}	9.2×10^{-3}	6.3×10^{-3}	1.15×10^{-2}	8.1×10^{-3}
E_k	9.0852×10^{-11}	8.1322×10^{-11}	8.1422×10^{-11}	7.4398×10^{-11}	9.1106×10^{-11}	7.7933×10^{-11}

5. 结论

本文讨论了一种基于SOR迭代的PIA算法,研究了该方法的收敛性. 以三次非均匀B样条基曲线插值为例, 验证了SOR-LSPIA算法的有效性, 大量的数值实例结果表明, SOR-LSPIA 算法的收敛速度优于GS-LSPIA算法. 我们的方法可以应用于大型稀疏矩阵求解, 还可以推广到三次B样条基函数下曲面插值逼近求解问题.

基金项目

湖南省研究生科研创新项目(CX20220953)。

参考文献

[1] 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报, 2015(4): 582-589.

[2] Lin, H.W., Maekawa, T. and Deng, C.Y. (2017) Survey on Geometric Iterative Methods and Their Applications. *Computer-Aided Design*, **95**, 40-51. https://doi.org/10.1016/j.cad.2017.10.002

[3] Qi, D.X., Tian, Z.X., Zhang, Y.X., *et al.* (1975) The Method of Numeric Polish in Curve Fitting. *Acta Mathematica Sinica*, **18**, 173-184. (In Chinese)

[4] de Boor, C. (1979) How Does Agee's Smoothing Method Work? Army Research Office, Washington DC.

[5] Lin, H.W., Wang, G.J. and Dong, C.S. (2004) Constructing Iterative Non-Uniform B-Spline Curve and Surface to Fit Data Points. *Science in China Series: Information Sciences*, **47**, 315-331. https://doi.org/10.1360/02yf0529

[6] Lin, H.W., Bao, H.J. and Wang, G.J. (2005) Totally Positive Bases and Progressive Iteration Approximation. *Computers & Mathematics with Applications*, **50**, 575-586. https://doi.org/10.1016/j.camwa.2005.01.023

[7] Lin, H.W. and Zhang, Z.Y. (2011) An Extended Iterative Format for the Progressive-Iteration Approximation. *Computers & Graphics*, **35**, 967-975. https://doi.org/10.1016/j.cag.2011.07.003

[8] Deng, C.Y. and Lin, H.W. (2014) Progressive and Iterative Approximation for Least Squares B-Spline Curve and Surface Fitting. *Computer-Aided Design*, **47**, 32-44. https://doi.org/10.1016/j.cad.2013.08.012

[9] Zhang, L., Ge, X.Y. and Tan, J.Q. (2016) Least Square Geometric Iterative Fitting Method for Generalized B-Spline Curves with Two Different Kinds of Weights. *The Visual Computer*, **32**, 1109-1120. https://doi.org/10.1007/s00371-015-1170-3

[10] Zhang, L., Tan, J.Q., Ge, X.Y., *et al.* (2018) Generalized B-Splines' Geometric Iterative Fitting Method with Mutually Different Weights. *Journal of Computatianal and Applied Mathematics*, **329**, 331-343. https://doi.org/10.1016/j.cam.2017.05.034

[11] Lin, H.W. and Zhang, Z.Y. (2013) An Efficient Method for Fitting Large Data Sets Using T-Splines. *SIAM Journal on Scientific Computing*, **35**, A3052-A3068. https://doi.org/10.1137/120888569

[12] Yusuf Fatihu Hamza, 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.

[13] Wu, N.C. and Liu, C.Z. (2022) Asynchronous Progressive Iterative Approximation Method for Least-Squares Fitting. arXiv:2211.06556

[14] J. W. Demme. 应用数值线性代数[M]. 王国荣, 译. 北京: 人民邮电出版社, 2007.

[15] Saad, Y. (2003) Iterative Methods for Sparse Linear Systems. 2nd Edition, SIAM, Philadelphia, PA.