基于 PIA 的非均匀三次 B 样条曲线 Hermite 插值

吴硕琳 李亚娟 邓重阳
(杭州电子科技大学理学院 杭州 310018)

摘 要 提出基于渐进迭代逼近（Progressive Iteration Approximation，PIA）的非均匀三次 B 样条曲线 Hermite 插值算法. 首先，以给定数据点作为初始控制顶点，采用累加弦长法得到节点序列，通过构造误差向量更新控制顶点，迭代生成插值数据点的非均匀三次 B 样条曲线. 当需要同时插值数据点和单位切向时，在每个节点区间上插入一个节点；当需要同时插值数据点、单位切向和曲率向量时，在每个节点区间上插入两个节点；更新初始控制顶点，进而迭代得到插值 B 样条曲线. 理论分析表明算法是收敛的. 数值算例结果说明，与均匀三次 B 样条曲线插值算法相比，当相邻数据点间距离变化程度越大时，该算法的收敛速度越快，在相同误差条件下迭代次数更少.

关键词 非均匀三次 B 样条曲线；迭代算法；Hermite 插值；渐进迭代逼近；控制顶点
中图法分类号 TP391 **DOI 号** 10.11897/SP.J.1016.2023.02463

Hermite Interpolation of Non-Uniform Cubic B-Spline Curve Based on PIA

WU Shuo-Lin LI Ya-Juan DENG Chong-Yang
(*School of Science, Hangzhou Dianzi University, Hang zhou* 310018)

Abstract We propose an algorithm for Hermite interpolation by non-uniform cubic B-spline based progressive iteration approximation (PIA). Firstly, interpolate given data points. Data points that need interpolation are used as known conditions, and apply cumulative chord length parameterization to define the knot vector. We use these points that need to be interpolated as the initial control points. The number of initial control points are n+3, and the initial non-uniform cubic B-spline curve is obtained by these n + 3 control points. The error vectors are constructed by the errors between the initial control points and the corresponding points on the initial non-uniform cubic B-spline curve. And then update the control points by using the constructed error vectors. The B-spline curve is obtained by using the updated control points. Constantly update the above process of iteration. The non-uniform cubic B-spline curve of interpolation data points is generated, which the specified iteration stop condition is the average distance between the control points and the corresponding points on the curve. The iteration is stopped when the average distance between the two is less than the specified error. Secondly, data points and tangent vectors need to be interpolated simultaneously. Based on the interpolation of the node vector of given data points, we insert one knot into each knot interval by dichotomy method and update node sequence. The n+3 new control points obtained by the interpolation given data points algorithm are used to calculate the initial control points after

收稿日期：2022-08-19；在线发布日期：2023-04-04. 本课题得到国家自然科学基金（61872121）、浙江省重点研发计划项目（2021C0018）资助. **吴硕琳**，硕士研究生，主要研究方向为计算机辅助几何设计与计算机图形学（CAGD&CG）. E-mail: 1483099583@qq.com. **李亚娟**，博士，副教授，硕士生导师，主要研究方向为计算机辅助几何设计与计算机图形学（CAGD&CG）. **邓重阳**（通信作者），博士，教授，博士生导师，中国计算机学会（CCF）高级会员，主要研究方向为计算机辅助几何设计与计算机图形学（CAGD&CG）. E-mail: dcy@hdu.edu.cn.

increasing the degree of freedom. The initial non-uniform cubic B-spline curve is obtained by increasing the number of initial control points to 2n+4. The error vectors are constructed by controlling the errors between the control points and the corresponding points on the initial non-uniform cubic B-spline curve. The control points are updated by using the constructed error vectors. The B-spline curve is obtained by using the updated control points. The above process is iterated continuously, and finally the non-uniform cubic B-spline curve of interpolated data points is generated. The iteration stop condition is when the average distance between the control points of two adjacent iterations is less than the specified error, the iteration is stopped. Finally, data points, tangent vectors and curvatures need to be interpolated simultaneously. Based on the interpolation of the node vector of the given data points, two nodes are inserted into each node interval by the method of ternary subdivision. The control points after increasing the degree of freedom are obtained. The initial non-uniform cubic B-spline curve is obtained by increasing the number of control points to 3n+5 as the initial control points. Then the interpolated B-spline curve is obtained by iteration. The iteration stop condition is also the average distance between the control points of two adjacent iterations is less than the specified error. Theoretical analysis shows that the algorithm is convergent. Numerical examples show that compared with the interpolation algorithm of uniform cubic B-spline curve, our algorithm converges faster, that is, fewer iterative steps are needed under the same error conditions when the distance between two data points change more.

Keywords non-uniform cubic B-spline curve; iterative algorithm; Hermite interpolation; progressive iteration approximation; control points

1 引 言

用 B 样条曲线曲面插值数据点集是计算机辅助几何设计（Computer Aided Geometric Design，CAGD）领域的一个基本问题. 为得到插值 B 样条曲线曲面的控制顶点，通常需要直接求解一个线性方程组. 除直接的方法外，还可以利用渐进迭代逼近（Progressive Iteration Approximation，PIA）等迭代方法求插值 B 样条曲线曲面的控制顶点.

1975 年，齐东旭等[1]在研究样条拟合问题时，基于盈亏修正思想提出了均匀三次 B 样条曲线插值的算法. 随后 CAGD 领域著名学者 de Boor[2]于 1979 年也阐述了这一思想. 1991 年，齐东旭[3]给出了均匀三次 B 样条曲线的迭代格式，并简单证明了其收敛性. 进而 Lin 等[4]详细给出了非均匀三次 B 样条曲线曲面插值的盈亏收敛方式和收敛性证明，并构造了相应的渐进迭代逼近（PIA）格式. PIA 算法不但避免求解线性方程组，而且几何意义明显，因此很快引起了学界的兴趣. 2015 年，刘晓艳等[5]提出了非均匀三次 B 样条曲线插值的 Jacobi-PIA 算法；同年又提出了非均匀三次 B 样条曲线插值的 GS-PIA 算法. 随后，王志好等[6]证明了 GS-PIA 算法的收敛性. PIA 算法既用于均匀参数的三角 Bernstein-Bézier 曲面[7-8]、Loop 细分曲面[9]、Catmull-Clark 细分[10]和 NURBS 曲线[11]等插值，也可以将 PIA 方法推广到所有全正基混合曲线曲面上[12].

PIA 算法还广泛应用于曲线曲面拟合. 2011 年，Lin 等[13]提出了一种扩展的渐进迭代逼近（Extended Progressive and Iterative Approximation，EPIA）算法，用于拟合数量较多的数据点. 类似于 EPIA，Deng 等[14]提出最小二乘渐进迭代逼近（Least Square Progressive and Iterative Approximation，LSPIA）算法，能够通过少量控制顶点拟合大规模数据点. Lin 等[15-16]也从理论和应用两个方面详细阐述了 PIA 算法，并说明了其应用前景非常广泛. 随后，李莎莎等[17]提出了数据点加权的最小二乘渐进迭代逼近（Date-Weighted Least Square Progressive and Iterative Approximation，DW-LSPIA）算法，并证明了其收敛性. Hamza 等[18]提出一种基于渐进迭代逼近的隐式曲线和曲面重建（Implicit Progressive-Iterative Approximation，I-PIA）方法.

Gofuku 等[19]在 2009 年介绍了一种几何插值算法，利用平面三次 B 样条曲线插值 G^1 数据，即插值数据点和单位切向，但计算速度较慢. 随后，Abbas 等[20]提出了一种在平面上构造 B 样条曲线的方法，用于生成在点、法向以及曲率约束下的均匀三次 B

样条曲线，该算法无需迭代即可准确、快速插值 G^2 数据，即插值位置、单位切向和曲率向量，曲线能够达到 G^2 连续，但不能为空间曲线提供准确的主法向和副法向.

为了使曲线在平面和空间上光顺，Okaniwa 等[21]于 2012 年提出了均匀三次 B 样条曲线插值 G^1 数据和 G^2 数据的算法，但当相邻两点间距离变化较大时，节点序列分布较不均匀，如果采用均匀三次 B 样条曲线进行插值，节点向量影响比较大，从而影响 B 样条曲线的插值效果，而利用非均匀三次 B 样条曲线插值，节点向量影响较小，故考虑用非均匀三次 B 样条曲线进行插值. 于是在 Okaniwa 等[21]的算法基础上，本文提出了基于 PIA 的非均匀三次 B 样条曲线的 Hermite 插值算法，使得曲线分别插值 G^0 数据、G^1 数据及 G^2 数据，算法适用于平面曲线和空间曲线. 理论分析表明，非均匀三次 B 样条曲线的 Hermite 插值算法是收敛的. 数值算例结果说明，当相邻数据点间距离变化越不均匀，该算法的收敛速度相比均匀三次 B 样条曲线插值算法越快，在相同误差要求下迭代次数较少.

2　非均匀三次 B 样条曲线矩阵表达形式

给定控制顶点 $\{P_i\}_{i=0}^n$，并用累加弦长法定义节点序列 $L=[l_{-3},l_{-2},\cdots,l_{n+1}]$ 中的元素[10]

$$l_i=\sum_{j=0}^{i}\|P_{j+1}-P_j\|,\ i=0,1,\cdots,n,$$

对于节点序列，引入记号

$$\Delta_i^1=\Delta_i=l_{i+1}-l_i, i=0,1,\cdots,n-3,$$
$$\Delta_{-3}=\Delta_{-2}=\Delta_{-1}=\Delta_0,$$
$$\Delta_{n-3}=\Delta_{n-2}=\Delta_{n-1}=\Delta_n,$$
$$\Delta_i^2=\Delta_i+\Delta_{i+1}=l_{i+2}-l_i,\ i=-3,-2,\cdots,n-1,$$
$$\begin{aligned}\Delta_i^3&=\Delta_i+\Delta_{i+1}+\Delta_{i+2}\\&=l_{i+3}-l_i, i=-3,-2,\cdots,n-2;\end{aligned}$$

则有

$$l_{-3}=l_0-3\Delta_0,$$
$$l_{-2}=l_0-2\Delta_0,$$
$$l_{-1}=l_0-\Delta_0,$$
$$l_{n+1}=l_{n-1}+2\Delta_n.$$

由给定的控制顶点 $\{P_i\}_{i=0}^n$ 和节点序列 L 得到非均匀三次 B 样条曲线

$$r(l)=\sum_{i=0}^{n}B_{i,3}(l)P_i \tag{1}$$

其中，$B_{i,3}(l)$ 为三次 B 样条基函数.

令 $\frac{l-l_i}{\Delta_i}=u$，则非均匀三次 B 样条曲线（1）中节点区间 $[l_i,l_{i+1}]$ 对应的非均匀三次 B 样条曲线矩阵表达式可表示为[22]

$$r_i(u)=[1\ \ u\ \ u^2\ \ u^3]N_i[P]_i, i=0,1,\cdots,n.$$

其中，

$$[P]_i=[P_{i-1},P_i,P_{i+1},P_{i+2}]^T,$$

N_i 为控制顶点 $P_{i-1},P_i,P_{i+1},P_{i+2}$ 对应的基函数矩阵，n_{rl}^i 为矩阵中对应位置的元素，即

$$N_i=\left[n_{rl}^i\right]=\begin{bmatrix}\frac{(\Delta_i)^2}{\Delta_{i-1}^2\Delta_{i-2}^3} & \frac{\Delta_{i-2}^2\Delta_i}{\Delta_{i-1}^2\Delta_{i-2}^3}+\frac{\Delta_i^2\Delta_{i-1}}{\Delta_{i-1}^2\Delta_{i-1}^3} & \frac{(\Delta_{i-1})^2}{\Delta_{i-1}^2\Delta_{i-1}^3} & 0\\ -3n_{11}^i & n_{22}^i & \frac{3\Delta_i\Delta_{i-1}}{\Delta_{i-1}^2\Delta_{i-1}^3} & 0\\ 3n_{11}^i & n_{32}^i & n_{33}^i & 0\\ -n_{11}^i & (n_{11}^i-n_{43}^i-n_{44}^i) & n_{43}^i & \frac{(\Delta_i)^2}{\Delta_i^2\Delta_i^3}\end{bmatrix},$$

$$n_{22}^i=n_{11}-\frac{1}{3}n_{23}-\frac{2\Delta_{i-2}^2\Delta_i}{\Delta_{i-1}^2\Delta_{i-2}^3}+\frac{2\Delta_i^2\Delta_i}{\Delta_{i-1}^2\Delta_{i-1}^3},$$

$$n_{32}=-2n_{11}+n_{23}+\frac{\Delta_{i-2}^2\Delta_i}{\Delta_{i-1}^2\Delta_{i-2}^3}-\frac{\Delta_{i+1}\Delta_i}{\Delta_{i-1}^2\Delta_{i-1}^3},$$

$$n_{33}=-n_{43}-n_{44}-\frac{2}{3}n_{23}+\frac{(\Delta_i)^2+2\Delta_{i+1}\Delta_i}{\Delta_i^2\Delta_{i-1}^3},$$

$$n_{43}^i=-n_{44}^i-\frac{(\Delta_i)^2}{\Delta_{i-1}^2\Delta_{i-1}^3}-\frac{(\Delta_i)^2}{\Delta_i^2\Delta_{i-1}^3}.$$

3　插值 G^0 Hermite 数据的 PIA 方法

给定数据点 $\{V_i\}_{i=0}^n$，每个数据点 V_i 所对应的参数值为 l_i，我们的目标是利用非均匀三次 B 样条曲线插值给定的数据点，如图 1 所示，其中波浪线表示省略部分.

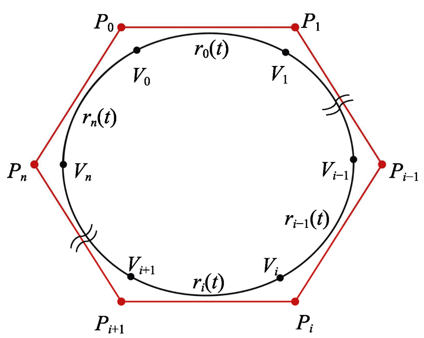


图 1　三次 B 样条曲线插值 $n+1$ 个数据点 $V_0,V_1,\cdots,V_n$（省略的点利用波浪线表示）

如果曲线闭合，规定控制顶点 $P_{-1}=P_n$，$P_{n+1}=$

P_0. 对于开曲线，Yamaguchi[23-24]在两端使用零曲率条件，规定控制顶点 $P_{-1}=P_0, P_{n+1}=P_n$.

非均匀三次 B 样条曲线插值给定数据点要满足如下条件

$$\begin{aligned} V_i = r_i(0) &= \begin{bmatrix}1 & 0 & 0 & 0\end{bmatrix} N_i[P]_i \\ &= n_{11}^i P_{i-1} + n_{12}^i P_i + n_{13}^i P_{i+1} \end{aligned} \tag{2}$$

通过式（2）获得误差向量

$$\delta_i = \frac{1}{n_{12}^i}V_i - P_i - \frac{n_{11}^i}{n_{12}^i}P_{i-1} - \frac{n_{13}^i}{n_{12}^i}P_{i+1}.$$

因此，构造迭代序列

$$\begin{aligned} P_i^{(k+1)} &= P_i^{(k)} + \delta_i \\ &= \frac{1}{n_{12}^i}(V_i - n_{11}^i P_{i-1}^{(k)} - n_{13}^i P_{i+1}^{(k)}) \end{aligned} \tag{3}$$

从而得到迭代矩阵

$$S^{(k)} = \begin{pmatrix} 1 & 1 & 1 \\ P_{i-2}^{(k)} & P_{i-1}^{(k)} & P_i^{(k)} \\ P_i^{(k)} & P_{i+1}^{(k)} & P_{i+2}^{(k)} \end{pmatrix},$$

其中，k 表示第 k 次迭代；第 0 次迭代时，令 $P_i^{(0)}=V_i$. 由于迭代矩阵 S 的谱半径小于 1，因此构造的迭代序列具有收敛性.

重复式（3）的过程，直到

$$r_i^{(k+1)}(0) - V_i < \varepsilon_p,\ \ i=0,1,\cdots,n.$$

这里 $r_i^{(k+1)}(u)$ 表示第 $k+1$ 次迭代时，节点区间 $[l_i, l_{i+1}]$ 对应的非均匀三次 B 样条曲线，ε_p 为规定误差.

最终，根据控制点集 $\{P_i^{(k+1)}\}_{i=-1}^{n+1}$ 得到非均匀三次 B 样条曲线.

4 插值 G^1 Hermite 数据的 PIA 方法

在第 3 节非均匀三次 B 样条曲线插值给定数据点基础上，本节将介绍曲线插值数据点和单位切向的算法.

4.1 每段节点区间上插入一个新节点

为使曲线插值给定数据点的同时，插值数据点处的单位切向，需要增加自由度，这可以通过在每段节点区间上插入一个新节点来完成.

每段节点区间中前半段曲线可以表示为

$$\begin{aligned} r_i\left(\frac{u}{2}\right) &= \begin{bmatrix}1 & \frac{u}{2} & \frac{u^2}{4} & \frac{u^3}{8}\end{bmatrix} N_i[P]_i \\ &= [1\ \ u\ \ u^2\ \ u^3] N_i [K]_i [P]_i, \end{aligned}$$

其中，

$$[K]_i = N_i^{-1}\begin{bmatrix}1 & & & \\ & \frac{1}{2} & & \\ & & \frac{1}{4} & \\ & & & \frac{1}{8}\end{bmatrix} N_i,$$

因此，得到新的控制顶点 $[Q]_j$：

$$\begin{aligned} [Q]_j &= [Q_{j-1}\ \ Q_j\ \ Q_{j+1}\ \ Q_{j+2}]^T \\ &= [K]_i[P]_i \\ &= N_i^{-1}\begin{bmatrix}1 & & & \\ & \frac{1}{2} & & \\ & & \frac{1}{4} & \\ & & & \frac{1}{8}\end{bmatrix} N_i[P]_i, \\ & i=0,1,\cdots,n;\ \ j=2i. \end{aligned} \tag{4}$$

下面图 2 中给出了需要插值的点 $\{V_i\}$、控制顶点 $\{P_i\}$、增加自由度后的新控制顶点 $\{Q_j\}$ 以及非均匀三次 B 样条曲线；其中，蓝色点表示新初始控制顶点.

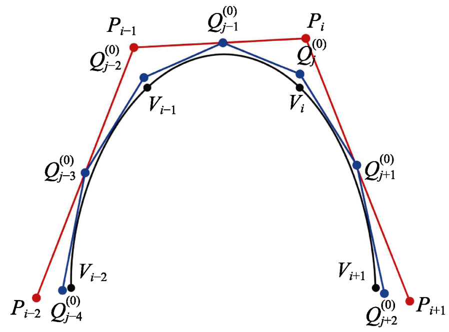


图 2 每个节点区间上插入一个节点后的三次 B 样条曲线

式（4）中利用 $P_i, i=-1,0,\cdots,n,n+1$ 产生新控制顶点 $Q_j, j=-1,0,\cdots,2n+1$. 此结果等同于通过非均匀三次 B 样条曲线的二进制细分[21]得到的结果. 因此可以根据 $r_i\left(\frac{u}{2}\right)$ 的控制顶点来表示 $r_i(0), r_i'(0), r_i''(0)$：

$$r_i(0) = n_{11}^i Q_{j-1} + n_{12}^i Q_j + n_{13}^i Q_{j+1} \tag{5}$$

$$r_i'(0) = n_{21}^i Q_{j-1} + n_{22}^i Q_j + n_{23}^i Q_{j+1} \tag{6}$$

$$r_i''(0) = n_{31}^i Q_{j-1} + n_{32}^i Q_j + n_{33}^i Q_{j+1} \tag{7}$$

而对于 $u=1$ 满足

$$r_i(1)=r_{i+1}(0),$$
$$r_i'(1)=r_{i+1}'(0),$$
$$r_i''(1)=r_{i+1}''(0).$$

4.2　平面曲线

我们的目标是插值 $n+1$ 个数据点 $V_0,V_1,\cdots,V_n$ 以及单位切向 $t_0,t_1,\cdots,t_n$，现在插值数据点和单位切向问题可以表示为

$$r_i(0)=V_i,\quad \frac{r_i'(0)}{\|r_i'(0)\|}=t_i,\quad i=0,1,\cdots,n. \tag{8}$$

将式（5）(6）代入到式（8）得到

$$\begin{gathered} n_{11}^iQ_{j-1}+n_{12}^iQ_j+n_{13}^iQ_{j+1}=V_i,\\ \frac{n_{21}^iQ_{j-1}+n_{22}^iQ_j+n_{23}^iQ_{j+1}}{\|n_{21}^iQ_{j-1}+n_{22}^iQ_j+n_{23}^iQ_{j+1}\|}=t_i,\\ i=0,1,\cdots,n;\ j=2i. \end{gathered} \tag{9}$$

根据式（9）可以得到迭代形式

$$Q_j^{(k+1)}=\frac{1}{n_{12}^i}(V_i-n_{11}^iQ_{j-1}^{(k)}-n_{13}^iQ_{j+1}^{(k)}) \tag{10}$$

$$\begin{gathered} Q_{j+1}^{(k+1)}=\frac{1}{n_{23}^i}\Big(\ \|n_{21}^iQ_{j-1}^{(k)}+n_{22}^iQ_j^{(k)}+\\ n_{23}^iQ_{j+1}^{(k)}\|t_i-n_{21}^iQ_{j-1}^{(k)}-n_{22}^iQ_j^{(k)}\Big),\\ i=0,1,\cdots,n；\ j=2i. \end{gathered} \tag{11}$$

在插值点和单位切向的算法中，为使迭代速度更快，这里引用高斯-赛德尔（Gauss-Seidel，GS）迭代法的思想[10,21]，式（10）不变，将式（11）中的 $Q_j^{(k)}$ 更新为 $Q_j^{(k+1)}$.

$$\begin{gathered} Q_{j+1}^{(k+1)}=\frac{1}{n_{23}^i}\Big(\ \|n_{21}^iQ_{j-1}^{(k)}+n_{22}^iQ_j^{(k+1)}+\\ n_{23}^iQ_{j+1}^{(k)}\|t_i-\ n_{21}^iQ_{j-1}^{(k)}-n_{22}^iQ_j^{(k+1)}\Big),\\ i=0,1,\cdots,n；\ j=2i. \end{gathered} \tag{12}$$

当相邻两次迭代之间任意控制顶点的位置变化小于规定误差时终止迭代，

$$\|Q_j^{(k+1)}-Q_j^{(k)}\|<\varepsilon_Q,\quad j=0,1,\cdots,2n.$$

最终，根据控制点集 $\{Q_j^{(k+1)}\}_{i=-1}^{2n+1}$ 得到非均匀三次 B 样条曲线.

4.3　空间曲线

对于平面曲线而言，主法向与副法向始终垂直于曲线所在的平面. 因此，当插值位置和切线方向确定时，法线方向会自动确定. 但此规则不再适用于空间曲线，因为副法向

$$b_i=\frac{r_i'(0)\times r_i''(0)}{\|r_i'(0)\times r_i''(0)\|},\quad i=0,1,\cdots,n.$$

不确定.

因此，对于空间曲线，需在每个点 V_i 处添加单位主法向 n_i 或副法向 b_i；当每个数据点没有施加法向约束时，可以直接利用 4.2 节的算法.

每次考虑三个控制顶点 $Q_{j-1}^{(k)}, Q_j^{(k)}$ 和 $Q_{j+1}^{(k)}$（$j=0,2,\cdots,2n$）. 其投影到经过 V_i 并由 t_i 和 n_i 构成的平面上[21]的点，分别表示为

$$\tilde{Q}_{j-1}^{(k)}=Q_{j-1}^{(k)}-((Q_{j-1}^{(k)}-V_i)\cdot b_i)b_i \tag{13}$$

$$\tilde{Q}_j^{(k)}=Q_j^{(k)}-((Q_j^{(k)}-V_i)\cdot b_i)b_i \tag{14}$$

$$\tilde{Q}_{j+1}^{(k)}=Q_{j+1}^{(k)}-((Q_{j+1}^{(k)}-V_i)\cdot b_i)b_i \tag{15}$$

其中，$b_i=t_i\times n_i$. 因为空间曲线的控制顶点在迭代过程中逐渐接近投影控制顶点，因此空间问题可以转化为平面问题，故可以直接使用 4.2 节中的算法.

4.4　收敛性分析

类似于文献[21]，我们将投影的极限曲线段表示为 $\tilde{r}_i^\infty$，从式（13）(14）(15）中可以得到

$$\begin{aligned} \tilde{r}_i^\infty(0)&=\lim_{k\to\infty}(n_{11}^i\tilde{Q}_{j-1}^{(k)}+n_{12}^i\tilde{Q}_j^{(k)}+n_{13}^i\tilde{Q}_{j+1}^{(k)})\\ &=\lim_{k\to\infty}(n_{11}^iQ_{j-1}^{(k)}+n_{12}^iQ_j^{(k)}+n_{13}^iQ_{j+1}^{(k)})-\\ &\lim_{k\to\infty}(n_{11}^iQ_{j-1}^{(k)}+n_{12}^iQ_j^{(k)}+n_{13}^iQ_{j+1}^{(k)}-\\ &(n_{11}^i+n_{12}^i+n_{13}^i)V_i)\cdot b_i\cdot b_i\\ &=V_i-(V_i-V_i)\cdot b_i\cdot b_i=V_i,\\ \tilde{r}_i^{'\infty}(0)&=\lim_{k\to\infty}(n_{21}^i\tilde{Q}_{j-1}^{(k)}+n_{22}^i\tilde{Q}_j^{(k)}+n_{23}^i\tilde{Q}_{j+1}^{(k)})\\ &=\lim_{k\to\infty}(n_{21}^iQ_{j-1}^{(k)}+n_{22}^iQ_j^{(k)}+n_{23}^iQ_{j+1}^{(k)})-\\ &\lim_{k\to\infty}\Big((n_{21}^iQ_{j-1}^{(k)}+n_{22}^iQ_j^{(k)}+n_{23}^iQ_{j+1}^{(k)}-\\ &(n_{21}^i+n_{22}^i+n_{23}^i)V_i)\cdot b_i\Big)\cdot b_i\\ &=\tilde{r}_i^{'}(0)-\tilde{r}_i^{'}(0)\cdot b_i\cdot b_i=\tilde{r}_i^{'}(0), \end{aligned}$$

这表明极限曲线 $\tilde{r}(t)$ 插值了给定的数据点 $\{V_i\}, i=0,1,\cdots,n$ 和单位切向 $t_i, i=0,1,\cdots,n$. 此外，极限曲线 $\tilde{r}(t)$ 还插值给定的副法向 b_i，因为

$$\begin{aligned} &\tilde{r}_i^{''\infty}(0)\cdot b_i\\ &=\lim_{k\to\infty}2(n_{31}^i\tilde{Q}_{j-1}^{(k)}+n_{32}^i\tilde{Q}_j^{(k)}+n_{33}^i\tilde{Q}_{j+1}^{(k)})\cdot b_i\\ &=\lim_{k\to\infty}2(n_{31}^iQ_{j-1}^{(k)}+n_{32}^iQ_j^{(k)}+n_{33}^iQ_{j+1}^{(k)})\cdot b_i-\\ &\lim_{k\to\infty}2\Big((n_{31}^iQ_{j-1}^{(k)}+n_{32}^iQ_j^{(k)}+n_{33}^iQ_{j+1}^{(k)}-\\ &(n_{31}^i+n_{32}^i+n_{33}^i)V_i)\cdot b_i\Big)\cdot b_i\cdot b_i\\ &=\lim_{k\to\infty}2(n_{31}^iQ_{j-1}^{(k)}+n_{32}^iQ_j^{(k)}+n_{33}^iQ_{j+1}^{(k)})\cdot b_i\\ &(1-b_i\cdot b_i)\\ &=(\tilde{r}_i^{''\infty}(0)\cdot b_i)(1-\|b_i^2\|)=0. \end{aligned}$$

5　插值 G^2 Hermite 数据的 PIA 方法

我们的目标除了插值 G^0 数据和 G^1 数据外，还要插值 G^2 数据．本节将介绍用非均匀三次 B 样条曲线插值数据点、单位切向及曲率向量的算法．

5.1　每段节点区间上插入两个新节点

如图 3 所示，在第 3 节基础上，每段节点区间上插入两个新节点来完成；其中，蓝色点表示初始控制顶点．利用 4.1 节的方法，可以直接得到增加自由度后的初始控制顶点 $[R]_j$：

$$[R]_j=[R_{j-1}\ R_j\ R_{j+1}\ R_{j+2}]^T$$
$$=N_i^{-1}\begin{bmatrix}1&&&\\&\frac{1}{3}&&\\&&\frac{1}{9}&\\&&&\frac{1}{27}\end{bmatrix}N_i[P]_i,$$
$$i=0,1,\cdots,n;\quad j=3i.$$

此结果等同于通过非均匀三次 B 样条曲线的三进制细分[21]得到的结果，因此可以根据 $r_i\left(\frac{u}{3}\right)$ 的控制顶点来表示 $r_i(0)$, $r_i'(0)$, $r_i''(0)$：

$$r_i(0)=n_{11}^iR_{j-1}+n_{12}^iR_j+n_{13}^iR_{j+1},\tag{16}$$
$$r_i'(0)=n_{21}^iR_{j-1}+n_{22}^iR_j+n_{23}^iR_{j+1},\tag{17}$$
$$r_i''(0)=n_{31}^iR_{j-1}+n_{32}^iR_j+n_{33}^iR_{j+1}.\tag{18}$$

而对于 $u=1$ 满足

$$r_i(1)=r_{i+1}(0),$$
$$r_i'(1)=r_{i+1}'(0),$$
$$r_i''(1)=r_{i+1}''(0).$$

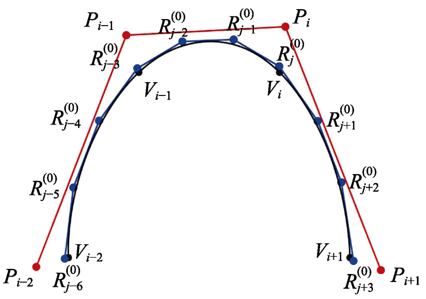


图3　每个节点区间上插入两个节点后的三次B样条曲线

5.2　平面曲线

我们的目标是插值 $n+1$ 个点 $V_0,V_1,\cdots,V_n$，单位切向量 $t_0,t_1,\cdots,t_n$ 以及曲率 $\kappa_0,\kappa_1,\cdots,\kappa_n$．平面曲线中，数据点、单位切向以及曲率向量的插值问题可以表示为

$$r_i(0)=V_i,\quad \frac{r_i'(0)}{\|r_i'(0)\|}=t_i,$$
$$\frac{\left(r_i'(0)\times r_i''(0)\right)\cdot e_z}{\|r_i'(0)\|^3}=\kappa_i,\quad i=0,1,\cdots,n.\tag{19}$$

其中，$e_z=(0,\ 0,\ 1)^T$．

进一步定义，

$$a=\|n_{21}^iR_{j-1}^{(k)}+n_{22}^iR_j^{(k)}+n_{23}^iR_{j+1}^{(k)}\|,\tag{20}$$
$$b=\|n_{21}^iR_{j-1}^{(k+1)}+n_{22}^iR_j^{(k+1)}+n_{23}^iR_{j+1}^{(k+1)}\|.\tag{21}$$

利用式（17）定义单位切向

$$t_i=\frac{1}{b}\left(n_{21}^iR_{j-1}^{(k)}+n_{22}^iR_j^{(k)}+n_{23}^iR_{j+1}^{(k)}\right).$$

与式（3）类似，使用式（16），修改如下：

$$R_j^{(k+1)}=\frac{1}{n_{12}^i}(V_i-n_{11}^iR_{j-1}^{(k)}-n_{13}^iR_{j+1}^{(k)}).\tag{22}$$

同时，得到 $R_{j+1}^{(k+1)}$、$R_{j-1}^{(k+1)}$ 的第 $k+1$ 次迭代公式：

$$R_{j+1}^{(k+1)}=\frac{1}{n_{23}^i}(at_i-n_{21}^iR_{j-1}^{(k)}-n_{22}^iR_j^{(k+1)}),\tag{23}$$
$$R_{j-1}^{(k+1)}=\frac{1}{n_{21}^i}(bt_i-n_{22}^iR_j^{(k+1)}-n_{23}^iR_{j+1}^{(k+1)}).\tag{24}$$

第 $k+1$ 次迭代的曲率可以表示为

$$\begin{aligned}\kappa_i&=\frac{\left(r_i'(0)\times r_i''(0)\right)\cdot e_z}{\|r_i'(0)\|^3}\\&=\frac{1}{b^3}\left\{2\left\|(n_{21}^in_{32}^i-n_{22}^in_{31}^i)(R_{j-1}^{(k+1)}\times R_j^{(k+1)}+\right.\right.\\&\quad\left.\left.R_{j+1}^{(k+1)}\times R_{j-1}^{(k+1)}+R_j^{(k+1)}\times R_{j+1}^{(k+1)}\right\|\right\}\end{aligned}\tag{25}$$

根据式（22）~（24），式（25）可以推导为

$$\begin{aligned}\frac{\kappa_i\|bt_i\|^3}{2(n_{21}^in_{32}^i-n_{22}^in_{31}^i)}&=\left\|\frac{b}{n_{12}^in_{21}^i}t_i\times V_i-\right.\\&\quad\left(\frac{n_{11}^i}{n_{12}^in_{21}^i}-\frac{1}{n_{23}^i}\right)bt_i\times R_{j-1}^{(k)}-\\&\quad\left.\frac{n_{13}^ib}{n_{12}^in_{21}^i}t_i\times R_{j+1}^{(k)}+\frac{n_{22}^ib}{n_{21}^in_{23}^i}t_i\times R_j^{(k)}\right\|\end{aligned}\tag{26}$$

5.2.1　当 $\kappa_i\neq 0$ 时

利用上式（26）可以推导出

$$b=\sqrt{\frac{2(n_{21}^in_{32}^i-n_{22}^in_{31}^i)}{\kappa_i\cdot\|\eta_1-\eta_2+\eta_3-\eta_4\|}}.\tag{27}$$

其中，

$$\eta_1=\frac{1}{n_{12}^in_{21}^i}t_i\times V_i,$$

$$\eta_2=\left(\frac{n_{11}^i}{n_{12}^i n_{21}^i}-\frac{1}{n_{23}^i}\right)t_i\times R_{j-1}^{(k)},$$

$$\eta_3=\frac{n_{22}^i}{n_{21}^i n_{23}^i}t_i\times R_j^{(k)},$$

$$\eta_4=\frac{n_{13}^i}{n_{12}^i n_{21}^i}t_i\times R_{j+1}^{(k)}.$$

式（20）、（27）分别给定 a,b，将 a,b 代入式（23）、（24），会得到迭代公式，如下：

$$R_j^{(k+1)}=\frac{1}{n_{12}^i}\left[V_i-n_{11}^i R_{j-1}^{(k)}-\frac{n_{13}^i}{n_{23}^i}(at_i-\right.$$

$$\left.n_{21}^i R_{j-1}^{(k)}-n_{22}^i R_j^{(k)})\right],$$

$$R_{j+1}^{(k+1)}=\frac{1}{n_{23}^i}\left[\ \left\|n_{21}^i R_{j-1}^{(k)}+n_{22}^i R_j^{(k+1)}+\right.\right.$$

$$\left.\left.n_{23}^i R_{j+1}^{(k)}\right\|t_i-n_{21}^i R_{j-1}^{(k)}-n_{22}^i R_j^{(k+1)}\right],$$

$$R_{j-1}^{(k+1)}=\frac{1}{n_{21}^i}(bt_i-n_{22}^i R_j^{(k+1)}-n_{23}^i R_{j+1}^{(k+1)}).$$

5.2.2　当 $\kappa_i=0$ 时

式（26）化简得到

$$\left\|\left(V_i-\frac{n_{11}^i n_{23}^i-n_{12}^i n_{21}^i}{n_{23}^i}R_{j-1}^{(k)}-\frac{n_{13}^i n_{23}^i}{n_{23}^i}R_{j+1}^{(k)}+\right.\right.$$

$$\left.\left.\frac{n_{12}^i n_{22}^i}{n_{23}^i}R_j^{(k)}\right)\times t_i\right\|=0.$$

令

$$d_i=V_i-\frac{n_{11}^i n_{23}^i-n_{12}^i n_{21}^i}{n_{23}^i}R_{j-1}^{(k)}-$$

$$\frac{n_{13}^i n_{23}^i}{n_{23}^i}R_{j+1}^{(k)}+\frac{n_{12}^i n_{22}^i}{n_{23}^i}R_j^{(k)}.$$

为满足上面的条件，d_i 需与 t_i 平行. 因此，满足

$$\frac{d_i}{\|d_i\|}=t_i,\quad i=0,1,\cdots,n.$$

故得到迭代公式：

$$R_j^{(k+1)}=\frac{1}{n_{12}^i}\left[V_i-n_{11}^i R_{j-1}^{(k)}-\frac{n_{13}^i}{n_{23}^i}\Big(at_i-\right.$$

$$\left.n_{21}^i R_{j-1}^{(k)}-n_{22}^i R_j^{(k)}\Big)\right],$$

$$R_{j-1}^{(k+1)}=\frac{n_{23}^i}{n_{11}^i n_{23}^i-n_{12}^i n_{21}^i}\left[V_i-n_{13}^i R_{j+1}^{(k)}+\right.$$

$$\left.\frac{n_{12}^i n_{22}^i}{n_{23}^i}R_j^{(k+1)}-\|d_i\|t_i\right],$$

$$R_{j+1}^{(k+1)}=\frac{1}{n_{13}^i}\left[V_i+\frac{n_{12}^i n_{22}^i}{n_{23}^i}R_j^{(k+1)}-\right.$$

$$\left.\frac{n_{11}^i n_{23}^i-n_{12}^i n_{21}^i}{n_{23}^i}R_{j-1}^{(k+1)}-\|d_i\|t_i\right].$$

当相邻两次迭代之间任意控制顶点的位置变化小于规定误差时终止迭代，

$$\|R_j^{(k+1)}-R_j^{(k)}\|<\varepsilon_R,\quad j=0,1,\cdots,3n.$$

最终,根据控制点集 $\{R_j^{(k+1)}\}_{i=-1}^{3n+1}$ 得到非均匀三次 B 样条曲线.

5.3　空间曲线

在数学上，可以通过添加以下条件到式（19）表示空间曲线插值点、单位切向和曲率向量问题：

$$\frac{r_i'(0)\times r_i''(0)}{\|r_i'(0)\times r_i''(0)\|}=b_i,\quad i=0,1,\cdots,n.$$

与第 4.3 节的处理方式相同，每次考虑三个控制顶点 $R_{j-1}^{(k)},R_j^{(k)}$ 和 $R_{j+1}^{(k)}$，$(j=0,2,\cdots,3n)$. 其投影到经过 V_i 并由 t_i 和 n_i 构成的平面上的点，分别表示为

$$\tilde{R}_{j-1}^{(k)}=R_{j-1}^{(k)}-((R_{j-1}^{(k)}-V_i)\cdot b_i)b_i,$$

$$\tilde{R}_j^{(k)}=R_j^{(k)}-((R_j^{(k)}-V_i)\cdot b_i)b_i,$$

$$\tilde{R}_{j+1}^{(k)}=R_{j+1}^{(k)}-((R_{j+1}^{(k)}-V_i)\cdot b_i)b_i.$$

此时 3D 问题转化为 2D 问题，因此可以直接使用 5.2 节算法. 类似于 4.4 收敛性分析该算法收敛.

6　数值实例

我们给出了如下四个实例. 实例图 4 至图 7 中(a)为输入的数据，(b)为曲线插值数据点和单位切向的效果图，(c)为插值数据点、单位切向及曲率向量的效果图，(d)为(c)的局部图，(e)为利用本文非均匀三次 B 样条曲线插值算法得到的曲率图，(f)为利用均匀三次 B 样条曲线插值算法得到的曲率图;其中，蓝色圆点表示需要插值的点，红色圆点表示控制顶点，蓝色箭头表示单位切向，橙色箭头为法向，黑色曲线表示插值曲线.

每个实例中的第一个表给出了在相同初始控制顶点的情况下，对比两种算法插值采样点的误差，这里我们采用平均误差

$$\bar{\sigma}_1=\frac{1}{n+1}\sum_{i=0}^{n}\|r_i^{(k)}(0)-V_i\|.$$

其中，$r_i^{(k)}(0)$ 是插值曲线上的点，V_i 是需要插值的点.

每个实例中的第二个表对比两种算法在相同迭代次数的条件下插值数据点、单位切向时产生的误

差；由于迭代过程具有收敛性，误差可以用迭代过程中对应控制顶点间的误差，这里我们通过平均误差计算

$$\bar{\sigma}_2=\frac{1}{2n+1}\sum_{j=0}^{2n}\|Q_j^{(k+1)}-Q_j^{(k)}\|. \tag{28}$$

其中，$Q_j^{(k)}$、$Q_j^{(k+1)}$分别表示第k、$k+1$次迭代的控制顶点.

同理，每个实例中的第三个表按照式（28）的方法计算迭代误差并对两种算法进行比较.

同时在相同误差情况下，比较均匀三次 B 样条曲线与非均匀三次 B 样条曲线插值G^2数据所运行的时间以及迭代次数，每个实例中的第四个表中对比了两种算法在相同误差条件下的运行时间及迭代次数.

6.1 平面曲线

这里给出了两个平面曲线的示例，六叶花瓣曲线及心形线.

（1）六叶花瓣曲线

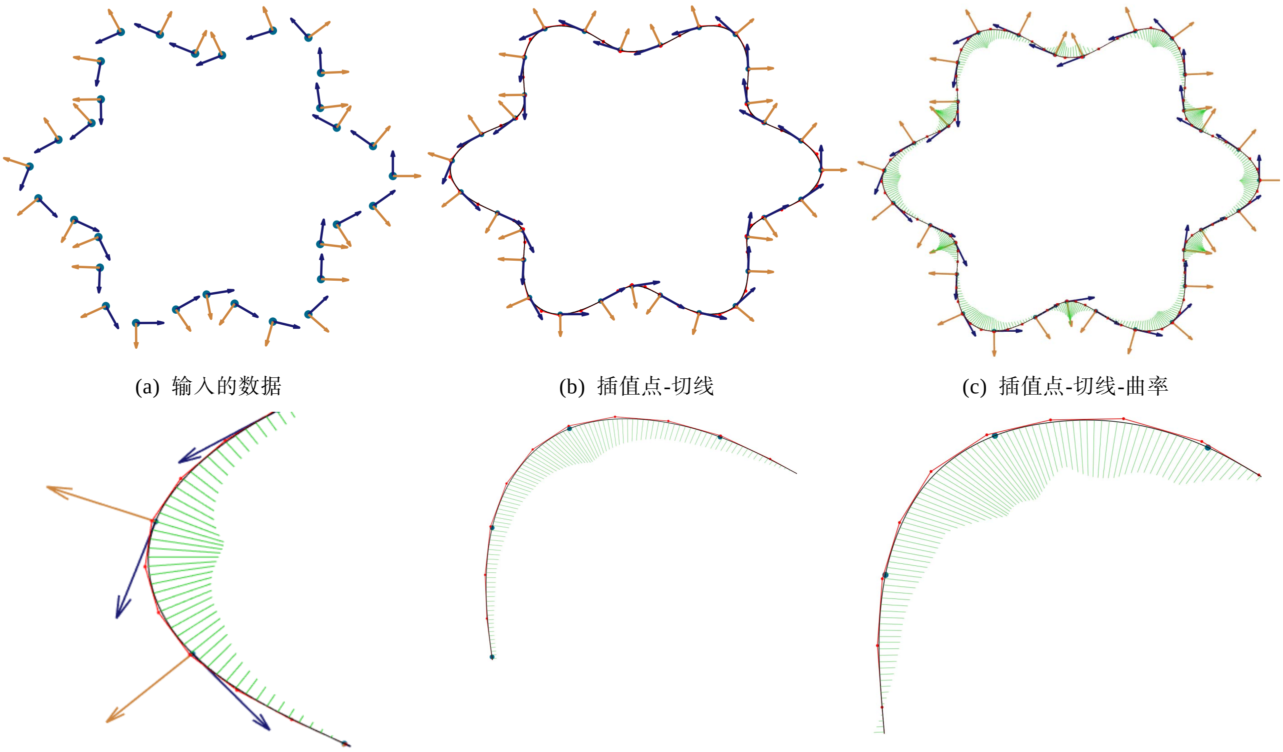
$$r(t)=[x(t),y(t)] \quad t\in[-4.8,\ 1.5]$$

其中，

$$\begin{cases} x(t)=\left[1+\dfrac{1}{6}\cos(6t)\right]\cos(t) \\ y(t)=\left[1+\dfrac{1}{6}\cos(6t)\right]\sin(t) \end{cases}$$

图 4 中六叶花瓣曲线的(e)、(f)曲率图对比说明非均匀三次 B 样条曲线插值算法得到的曲线更光顺.

在相同迭代次数时，表 1 至表 3 对比了六叶花瓣曲线的均匀三次 B 样条曲线与非均匀三次 B 样条曲线分别插值G^0数据、G^1数据和G^2数据的误差. 表 1 至表 3 数据说明，当迭代次数相同时，非均匀 B 样条曲线的插值误差更小.



(a) 输入的数据 (b) 插值点-切线 (c) 插值点-切线-曲率

(d) 局部图 (e) 非均匀三次 B 样条曲线 (f) 均匀三次 B 样条曲线

图 4 六叶花瓣曲线

表 1 两种算法插值六叶花瓣上相同采样点的误差

迭代次数	均匀 B 样条曲线	非均匀 B 样条曲线
1	0.0011	6.2811e-4
5	4.0320e-8	4.7559e-9
10	2.7351e-13	6.7933e-15
20	3.6215e-25	9.6679e-26

表 2 两种算法插值六叶花瓣上G^1数据的误差

迭代次数	均匀 B 样条曲线	非均匀 B 样条曲线
1	1.7708e-4	8.8325e-5
5	3.0830e-8	6.3098e-9
10	1.6629e-11	9.0669e-12
20	6.7411e-15	4.3198e-15

表 3　两种算法插值六叶花瓣上 G^2 数据的误差

迭代次数	均匀 B 样条曲线	非均匀 B 样条曲线
1	1.8857e-4	1.1144e-4
5	8.1790e-7	2.0358e-7
10	1.6216e-10	1.5070e-12
20	7.0293e-16	1.9412e-16

表 4　相同误差下插值六叶花瓣的两种算法比较

误差	迭代	均匀B样条曲线	非均匀B样条曲线
1e-5	次数（次）	9	8
	时间（秒）	0.323224	0.304444
1e-10	次数（次）	19	18
	时间（秒）	0.628927	0.578279
1e-15	次数（次）	29	28
	时间（秒）	0.958684	0.847357

相同误差条件下，表 4 对比了六叶花瓣曲线的均匀三次 B 样条曲线与非均匀三次 B 样条曲线插值 G^2 数据所运行的时间以及迭代次数．对比数据表明，非均匀三次 B 样条曲线的运行时间更短，迭代次数更少.

（2）心形曲线

$$r(t)=[x(t),y(t)]\qquad t\in[0,2\pi]$$

其中

$$\begin{cases} x(t)=\sin(t)-\dfrac{1}{2}\sin(2t) \\ y(t)=\cos(t)-\dfrac{1}{2}\cos(2t) \end{cases}$$

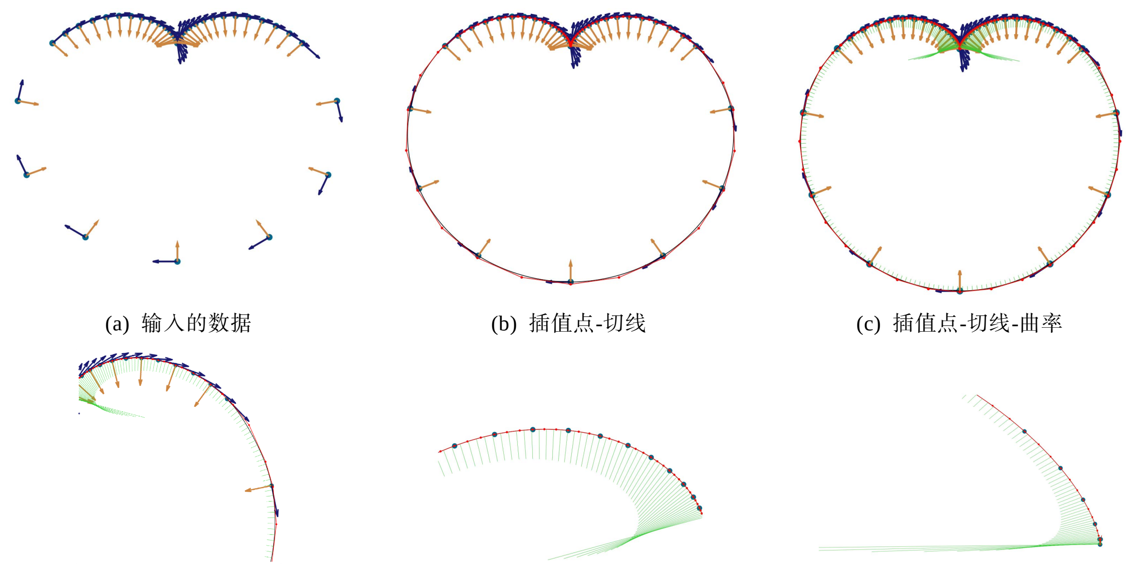
图 5 中心形曲线的(e)、(f)曲率图对比说明非均匀三次 B 样条曲线插值算法得到的曲线更光顺.

在相同迭代次数时，表 5 至表 7 对比了心形曲线的均匀三次 B 样条曲线与非均匀三次 B 样条曲线分别插值 G^0 数据、G^1 数据和 G^2 数据的误差．表中数据说明，当迭代次数相同时，非均匀三次 B 样条曲线的插值误差更小.

相同误差条件下，表 8 对比了心形曲线的均匀三次 B 样条曲线与非均匀三次 B 样条曲线插值 G^2 数据所运行的时间以及迭代次数．对比数据表明，非均匀三次 B 样条曲线的运行时间更短，迭代次数更少.

6.2　空间曲线

这里给出两个空间曲线的示例，非闭合螺旋线及一条不规则曲线.



(a) 输入的数据　(b) 插值点-切线　(c) 插值点-切线-曲率

(d) 局部图　(e) 非均匀三次 B 样条曲线　(d) 均匀三次 B 样条曲线

图 5　心形曲线

表 5　两种算法插值心形曲线上相同采样点的误差

迭代次数	均匀 B 样条曲线	非均匀 B 样条曲线
1	0.0112	0.0091
5	7.6762e-6	3.6158e-9
10	6.6613e-8	4.2067e-10
20	1.4745e-9	4.1633e-13

表 6　两种算法插值心形曲线上 G^1 数据的误差

迭代次数	均匀 B 样条曲线	非均匀 B 样条曲线
1	8.7430e-4	2.7756e-4
5	9.9920e-7	5.5511e-9
10	7.1471e-9	1.1102e-12
20	9.2981e-11	1.5613e-14

表 7 两种算法插值心形曲线上 G^2 数据的误差

迭代次数	均匀 B 样条曲线	非均匀 B 样条曲线
1	8.8818e-4	2.2204e-5
5	8.0491e-7	3.3307e-9
10	7.7716e-10	3.9345e-11
20	4.4409e-12	1.8874e-15

表 8 相同误差下插值心形曲线的两种算法比较

误差	迭代	均匀B样条曲线	非均匀B样条曲线
1e-5	次数（次）	9	8
	时间（秒）	0.625115	0.544247
1e-10	次数（次）	19	18
	时间（秒）	0.997687	0.783717
1e-15	次数（次）	29	29
	时间（秒）	1.158335	1.086315

（1）非闭合螺旋线

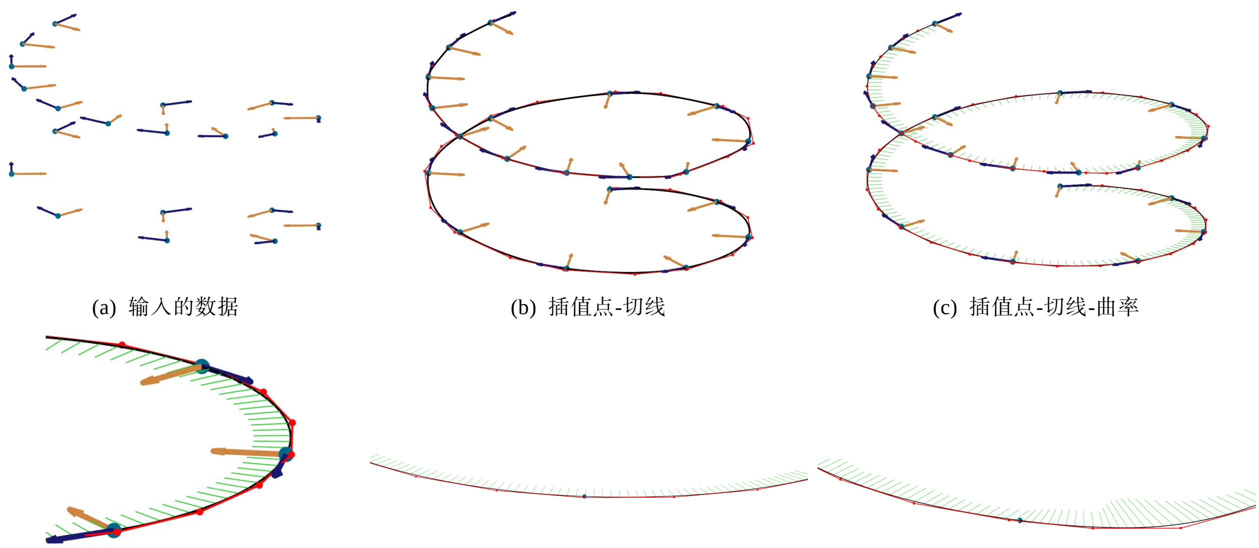
$$r(t)=\left[x(t),y(t),z(t)\right],\quad t\in\left[0,4\pi\right].$$

其中

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

图 6 中的(e)、(f)曲率图对比说明非均匀三次 B 样条曲线插值算法得到的曲线更光顺.

在相同迭代次数时，表 9 至表 11 对比了非闭合螺旋线的均匀 B 样条曲线与非均匀 B 样条曲线分别插值 G^0 数据、G^1 数据和 G^2 数据的误差. 表中数据说明，当迭代次数相同时，非均匀 B 样条曲线的插值误差更小.



(a) 输入的数据 (b) 插值点-切线 (c) 插值点-切线-曲率

(d) 局部图 (e) 非均匀三次 B 样条曲线 (f) 均匀三次 B 样条曲线

图 6 非闭合螺旋线

表 9 两种算法插值非闭合螺旋线上相同采样点的误差

迭代次数	均匀 B 样条曲线	非均匀 B 样条曲线
1	0.0213	0.0175
5	6.3971e-7	1.4823e-7
10	6.6775e-12	1.2038e-12
20	2.8641e-14	3.7717e-15

表 10 两种算法插值非闭合螺旋线上 G^1 数据的误差

迭代次数	均匀 B 样条曲线	非均匀 B 样条曲线
1	7.4685e-4	2.6100e-4
5	7.6214e-6	2.6697e-6
10	1.0386e-6	5.6330e-7
20	1.2323e-11	7.3781e-13

相同误差条件下，表 12 对比了非闭合螺旋线的均匀 B 样条曲线与非均匀 B 样条曲线插值 G^2 数据

表 11 两种算法插值非闭合螺旋线上 G^2 数据的误差

迭代次数	均匀 B 样条曲线	非均匀 B 样条曲线
1	4.1802e-9	1.7837e-10
5	7.7216e-12	1.6844e-13
10	5.9289e-14	1.2401e-14
20	7.5000e-17	1.3295e-19

表 12 相同误差下插值非闭合螺旋线的两种算法比较

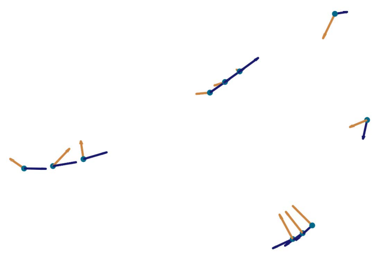
误差	迭代	均匀B样条曲线	非均匀B样条曲线
1e-5	次数（次）	10	9
	时间（秒）	0.237130	0.194085
1e-10	次数（次）	13	12
	时间（秒）	0.434989	0.431937
1e-15	次数（次）	28	27
	时间（秒）	2.674380	2.083651

所运行的时间以及迭代次数．对比数据表明，非均匀 B 样条曲线的运行时间更短，迭代次数更少．

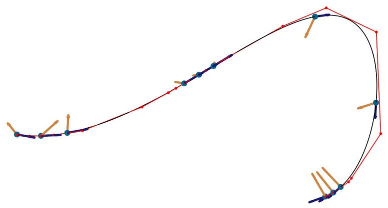
（2）不规则曲线

$$r(t)=\left[x(t),y(t),z(t)\right],\quad t\in\left[0,\frac{\pi}{2}\right].$$

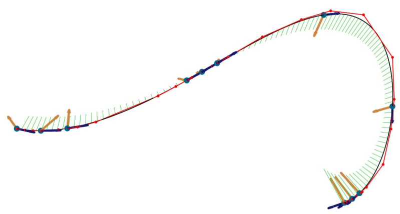
其中，



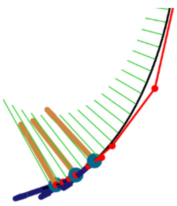
(a) 输入的数据



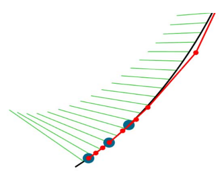
(b) 插值点-切线



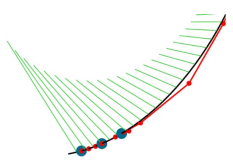
(c) 插值点-切线-曲率



(d) 局部图



(e) 非均匀三次 B 样条曲线



(f) 均匀三次 B 样条曲线

图 7　不规则曲线

$$\begin{cases}x(t)=\cos(2t)+\cos(4t)+10\left[\cos(t)+\cos(3t)\right]\\ y(t)=6\sin(t)+10\sin(3t)\\ z(t)=4\sin(4t)-2\left[\sin(6t)+\cos(6t)\right]+2\end{cases}.$$

图 7 中的(e)、(f)曲率图对比说明非均匀三次 B 样条曲线插值算法得到的曲线更光顺．

表 13　两种算法插值不规则曲线上相同采样点的误差

迭代次数	均匀 B 样条曲线	非均匀 B 样条曲线
1	1.4955	2.5305e-10
5	0.0060	1.1890e-15
10	1.2770e-05	4.0678e-17
20	6.9379e-11	9.6724e-19

表 14　两种算法插值不规则曲线上 G^1 数据的误差

迭代次数	均匀 B 样条曲线	非均匀 B 样条曲线
1	0.3632	7.5430e-10
5	4.9226e-4	4.4501e-16
10	3.8406e-6	1.4640e-17
20	2.5729e-11	2.8669e-19

表 15　两种算法插值不规则曲线上 G^2 数据的误差

迭代次数	均匀 B 样条曲线	非均匀 B 样条曲线
1	0.0213	3.5510e-10
5	1.3753e-4	1.7933e-15
10	1.1313e-6	5.1792e-17
20	9.2937e-11	8.4469e-18

在相同迭代次数时，表 13 至表 15 对比了不规则曲线的均匀 B 样条曲线与非均匀 B 样条曲线分别插值 G^0 数据、G^1 数据和 G^2 数据的误差．表中数据说明，当迭代次数相同时，非均匀 B 样条曲线的插值误差更小．

表 16　相同误差下不规则曲线的两种算法比较

误差	迭代	均匀 B 样条曲线	非均匀 B 样条曲线
1e-5	次数（次）	11	2
	时间（秒）	0.200175	0.066268
1e-10	次数（次）	20	2
	时间（秒）	0.352067	0.080213
1e-15	次数（次）	29	2
	时间（秒）	0.526435	0.063208

相同误差条件下，表 16 对比了空间上不规则曲线的均匀 B 样条曲线与非均匀 B 样条曲线插值 G^2 数据所运行的时间以及迭代次数．对比数据表明，非均匀 B 样条曲线的运行时间更短，迭代次数更少．

7 结 语

本文提出了非均匀三次 B 样条曲线插值 G^0 Hermite 数据、G^1 Hermite 数据和 G^2 Hermite 数据的PIA 算法,并证明了算法的收敛性. 第 6 节数值实例结果表明，相比于均匀三次 B 样条曲线插值算法，当相邻数据点间距离变化程度较大时，本文算法的收敛速度越快. 在相同误差条件下迭代次数更少，迭代时间也更短. 下一步工作将讨论基于 PIA 的非均匀三次 B 样条曲面 Hermite 插值.

致 谢 感谢向本文提出宝贵建议的审稿专家!

参 考 文 献

[1] Qi DongXu, Tian ZiXian, Zhang YuXin, et al. The method of numeric polish in curve fitting. Acta Mathematica Sinica, 1975, 18(3): 173-184(inChinese)
(齐东旭，田自贤，张玉心等. 曲线拟合的数值磨光法.数学学报, 1975, 18(3): 173-184)

[2] de Boor C. How does Agee's smoothing method work// Proceedings of the 1979 Army Numerical Analysis and Computers Conference. Army Research Office Report, 1979, 79(3): 299-302

[3] Qi DongXu. Some notes on mathematical methods in computer aided geometric modeling. Journal of the Northern Industrial University, 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)

[4] Lin HongWei, Wang GuoJin, Dong ChenShi. Constructing iterative non-uniform B-spline curve and surface to fit data points. Science in China Series: Information Sciences, 2004, 47(3): 315-331

[5] Liu XiaoYan, Deng ChongYang. JacobiPIA algorithm for non-uniform cubic B-spline curve interpolation. Journal of Computer-Aid Design & Computer Graphics, 2015, 27(3): 485-491(in Chinese)
(刘晓艳，邓重阳. 非均匀三次 B 样条曲线插值的 Jacobi-PIA 算法. 计算机辅助设计与图形学学报, 2015, 27(3): 485-491)

[6] Wang ZhiHao, Li YaJuan, Deng ChongYang. Convergence proof of GS-PIA algorithm. Journal of Computer-Aid Design & Computer Graphics, 2018, 30(11): 2035-2041(in Chinese)
(王志好，李亚娟，邓重阳. GS-PIA 算法的收敛性证明. 计算机辅助设计与图形学报, 2018, 30(11): 2035-2041)

[7] Chen Jun, Wang GuoJin. Progressive iterative approximation for triangular Bezier surface. Computer-Aided Design, 2011, 43(8): 889-895

[8] Lin HongWei. Survey on geometric iterative methods with applications. Journal of Computer-Aid Design & Computer Graphics, 2015, 27(4): 582-589(in Chinese)
(蔺宏伟. 几何迭代法及其应用综述. 计算机辅助设计与图形学报, 2015, 27(4): 582-589)

[9] Cheng FuHua, Fan FengTao, Lai ShuHua, et al. Loop subdivision surface based progressive interpolation. Journal of Computer Science and Technology, 2009, 24(1):39-46

[10] Chen ZhongXian, Luo XiaoNan, Tan Le, et al. Progressive interpolation based on Catmull-Clark subdivision surfaces. Computer Graphics Forum, 2008, 27(7): 1823-1827

[11] Zhang Meng, Li YaJuan, Deng ChongYang. Optimizing NURBS curves fitting by least squares progressive and iterative approximation. Journal of Computer-Aid Design & Computer Graphics, 2020, 32(4): 568-574(in Chinese)
(张蒙，李亚娟，邓重阳. NURBS 曲线拟合的最小二乘渐进迭代逼近优化算法. 计算机辅助设计与图形学报，2020，32(4): 568-574)

[12] Lin HongWei, Bao HuJun, Wang GuoJin. Totally positive bases and progressive iteration approximation. Computers & Mathematics with Applications, 2005, 50(3): 575-586

[13] Lin HongWei, Zhang ZhiYu. An extended iterative format for the progressive-iteration approximation. Computers & Graphics, 2011, 35(5): 967-975

[14] Deng ChongYang, Lin HongWei. Progressive and iterative approximation for least squares B-spline curve and surface fitting. Computer-Aided Design, 2014, 47(1): 32-44

[15] Lin HongWei, Maekawa T, Deng ChongYang. Survey on geometric iterative methods and their applications. Computer-Aided Design, 2018, 95: 40-51

[16] Lin HongWei, Cao Qi, Zhang XiaoTing. The convergence of Least-Squares progressive iterative approximation for singular Least-Squares fitting system. Journal of Systems Science & Complexity, 2018, 31(06): 1618-1632

[17] Li ShaSha, Xu HuiXia, Deng ChongYang. Data-Weighted least square progressive and iterative approximation and related B-Spline curve fitting. Journal of Computer-Aid Design & Computer Graphics, 2019, 31(9): 1574-1580(in Chinese)
(李莎莎，徐惠霞，邓重阳. 数据点加权最小二乘渐进迭代逼近及其 B 样条曲线拟合. 计算机辅助设计与图形学学报，2019, 31(9): 1574-1580)

[18] Hamza Y F, Lin HongWei, Li ZiHao. Implicit progressive-iterative approximation for curve surface reconstruction. Computer Aided Geometric Design, 2020, 77: 1-15

[19] Gofuku S, Tamura S, Maekawa T. Point-tangent/point- normal B-spline curve interpolation by geometric algorithms. Computer-Aided Design, 2009, 41(6): 412-422

[20] Abbas A, Nasri A, Maekawa T. Generating B-spline curves with points, normals and curvature: A constructive approach. The Visual Computer, 2010, 26: 823-829

[21] Okaniwa S, Nasri A, Lin Hong-Wei, Abbas A, Kineri Y, Maekawa T. Uniform B-spline curve interpolation with prescribed tangent and curvature vectors. IEEE Transactions on Visualization and Computer Graphics, 2012: 18(9): 1474-1487

[22] Lin Feng, Wang Di. Research on interpolation algorithm of cubic non-uniform B-spline curve. Modular Machine Tool and Automated Processing Technology, 2012: 8(8): 32-35

[23] Yamaguchi F. A method of designing free form surfaces by

computer display (1st Report) (in Japanese). Precision Machinery, 1977: 43(2): 168-173

[24] Yamaguchi F. Curves and Surfaces in Computer Aided Geometric Design. Springer-Verlag, 1988

[25] Li QingYang, Wang NengChao, Yi DaYi. Numerical Analysis. 2008. Beijing Tsinghua University: Tsinghua University Press, 2014(in Chinese)
(李庆扬, 王能超, 易大义. 数值分析. 北京清华大学: 清华大学出版社, 2014)

WU Shuo-Lin, M. S. candidate. Her main research interests include computer aided geometric design and computer graphics.

LI Ya-Juan, Ph. D., associate professor. Her main research interests include computer aided geometric design and computer graphics.

DENG Chong-Yang, Ph. D., professor. His main research interests include computer aided geometric design and computer graphics.

Background

The problem studied in this paper belongs to the field of computer aided geometric design (CAGD) and computer graphics (CG). The generation of B-spline curves and surfaces from given data is an important problem in CAGD and CG, and has been widely used in automobile, aircraft, shipbuilding, clothing and other industries. Progressive and iteration approximation (PIA) is a method for data interpolation with the advantages, such convexity preserving, having intuitional geometric meaning, etc. Owing to its desirable properties, PIA has been used in CAGD and related areas PIA has wide applications in academic studies and engineering practices.

PIA for Hermite interpolation by uniform cubic B-spline has been proposed by Okaniwa et al. Considered that non-uniform B-spline is more flexible than uniform B-spline when the data points are non-uniform, we generalize the PIA for Hermite interpolation from uniform cubic B-spline to non-uniform cubic B-spline. Numerical examples confirm that for non-uniform data, PIA based on non-uniformcubic B-spline curves perform better than uniform cubic B-spline curves.

This work was supported by the National Natural Science Foundation of China (NSFC) under the project numbers 61872121 and the Zhejiang Provincial Science and Technology Program in China under Grant 2021C01108.