

闭合 B 样条曲线控制点的快速求解算法及应用

任 菲, 王家润, 仇建伟, 聂 芸

(华北计算技术研究所, 北京 100083)

摘 要: 为了提高求解闭合 B 样条曲线控制点的速度, 提出了一种基于专用 LU 分解的求解算法。根据控制点方程组系数矩阵的特点, 参照追赶法的 LU 分解, 构造了解析后的 L 、 U 矩阵的结构。基于这两个矩阵的结构特征设计了专用的 LU 分解方法, 具有较少的存储空间和计算量。在此基础上, 根据追赶法的原理, 设计了闭合 B 样条曲线控制点的快速求解算法。通过数值实验和在等值线光滑中的实际应用, 表明了该算法的可靠性和有效性。

关键词: B 样条曲线; 闭合曲线; 控制点; LU 分解; 追赶法; 等值线光滑

中图分类号: TP391 **文献标识码:** A **文章编号:** 1000-7024 (2011) 04-1374-05

Quick algorithm of solving control points of closed B-Spline curve and application

REN Fei, WANG Jia-run, QIU Jian-wei, NIE Yun

(North China Institute of Computing Technology, Beijing 100083, China)

Abstract: To improve speed of solving closed B-Spline curve control points, a quick solving algorithm based on appropriate LU decomposition is presented. Based on the characteristic of the control points equations, decomposed L and U matrix structures are constructed according to the LU decomposition of pursuit method. An appropriate LU decomposition method is designed based on the structural characteristics of these two matrices, with less storage and computation. Then, a quick algorithm of solving control points of closed B-Spline curve is designed according to the principle of pursuit method. The reliability and efficiency of the algorithm is proved by numerical tests and practical application in contours smoothing.

Key words: B-Spline curve; closed curve; control points; LU decomposition; pursuit method; contour smoothing

0 引 言

B 样条曲线在计算机辅助设计与制造中应用广泛。构造 B 样条曲线一般有正算和反算两种方式, 反算即插值方式是已知曲线上的型值点 $\{V_i\}$, 反解出控制点 $\{P_i\}$, 再由控制点 $\{P_i\}$ 构造 B 样条曲线, 这种方法能够有效地解决计算机辅助几何设计中几何造型的问题。

在反算过程中求解控制点时, 非闭合曲线和闭合曲线的控制点方程组有不同的形式。非闭合曲线的求解较容易, 谢志鹏^[1]通过矩阵的 LU 分解求解控制点方程组, 快速地生成了 B 样条曲线, 吴光亚^[2]通过对 A^{-1} 的研究提出一种快速求解算法, 柴本成^[3]提出了一种通用的求解算法, 根据给定型值点和端点处的切向量构造反算矩阵。现有的非闭合曲线控制点的求解算法能够满足实时性的要求, 闭合曲线的方程组则较复杂, 现有文献在处理光滑封闭曲线时, 绝大多数都是采用开曲线造型方法, 仅保证了曲线在封闭点的 G_1 、 G_2 连续, 不能构造真正意义上的封闭曲线^[4]。为了构造真正的闭合曲线, 需要

对闭合 B 样条曲线的控制点方程组直接求解, 运算量比较大。李学艺^[4]提出了一种双列带阵线性方程组求解算法, 效果比较好但过程比较复杂。本文提出了一种专用的 LU 分解方法, 算法过程简单易实现, 计算效率高, 存储空间小。基于该方法推导了解析闭合 B 样条曲线控制点的快速算法, 算法可靠有效, 可用于大数据量情形下的闭合曲线生成。

1 闭合 B 样条曲线的构造

本文研究 B 样条曲线的一种最简单而且最有用的情况, 即三次均匀 B 样条曲线。求解 B 样条曲线控制点的步骤为: 确定曲线的型值点集; 列出满足约束的方程; 根据需要, 确定端点的边界条件; 求解方程, 得到控制点集。

本文使用确定的型值点集 $\{V_i\}$, 对于三次均匀 B 样条曲线, 型值点 $\{V_i\}$ 和控制点 $\{P_i\}$ 的关系为

$$s(0) = (P_{i-1} + 4P_i + P_{i+1})/6 = V_i \quad (i=1, 2, \dots, n) \quad (1)$$

该方程组有 n 个方程, $n+2$ 个未知数, 因此需补充端点的边界条件。对于闭合曲线, 为了保证整条曲线首尾拼接的光滑

收稿日期: 2010-05-10; 修订日期: 2010-07-21。

作者简介: 任菲 (1985—), 女, 河北石家庄人, 硕士研究生, 研究方向为空间天气数据可视化; 王家润 (1968—), 男, 山东莒南人, 硕士, 研究方向为计算机图形学、科学计算可视化; 仇建伟 (1960—), 男, 江苏南京人, 硕士, 研究员, 研究方向为系统体系结构、信息协同; 聂芸 (1973—), 女, 山东济南人, 高级工程师, 研究方向为指挥自动化。E-mail: fr.good@163.com

性, 补充的控制点 $P_0 = P_n, P_{n+1} = P_1$, 根据式(1)可得反求控制点的方程组如下

$$\begin{pmatrix} 1 & 4 & 1 & & & 1 \\ & 1 & 4 & 1 & & \\ & & \dots & & & \\ & & & 1 & 4 & 1 \\ 1 & & & & 1 & 4 \\ 4 & 1 & & & & 1 \end{pmatrix} \begin{pmatrix} P_0 \\ P_1 \\ \vdots \\ P_{n-3} \\ P_{n-2} \\ P_{n-1} \end{pmatrix} = 6 \begin{pmatrix} V_1 \\ V_2 \\ \vdots \\ V_{n-2} \\ V_{n-1} \\ V_n \end{pmatrix} \quad (2)$$

方程组有 n 个方程, n 个未知数, 因此方程组有唯一解。根据输入的类型值 V_1, V_2, \dots, V_n 可求得控制点 P_0, P_1, \dots, P_{n-1} , 再由 $P_0 = P_n, P_1 = P_{n+1}$, 可得全部的控制点 P_0, P_1, \dots, P_{n+1} 。

2 反解控制点算法优化

为了构造闭合曲线, 需要对闭合 B 样条曲线的控制点方程组直接求解, 即求解上面的方程组(2)。方程组的形式并不规律, 根据已有的研究可知该求解过程较复杂, 需要大量的时间和空间。为了能够更好地满足实时性的要求, 需要对求解过程进行改进和优化, 首先从分析矩阵的结构特征入手, 设计可行的优化算法。

2.1 L、U 矩阵结构推导

观察控制点方程组的特点, 如果把最后一行的方程调整到第一行, 则其系数矩阵会成为对称矩阵, 具有较好的结构, 调整后结果如下

$$\begin{pmatrix} 4 & 1 & & & & 1 \\ 1 & 4 & 1 & & & \\ & 1 & 4 & & & \\ & & \dots & & & \\ & & & 1 & 4 & 1 \\ & & & & 1 & 4 \\ 1 & & & & & 1 \end{pmatrix} \begin{pmatrix} P_0 \\ P_1 \\ \vdots \\ P_{n-3} \\ P_{n-2} \\ P_{n-1} \end{pmatrix} = 6 \begin{pmatrix} V_n \\ V_{n-1} \\ \vdots \\ V_2 \\ V_1 \\ V_0 \end{pmatrix}$$

该矩阵除左下角和右上角各有一个非零元素外, 其余非零元素均位于 3 条对角线上。这种矩阵形式在计算机中存储时, 只需存 $\{a_i\}, \{b_i\}, \{c_i\}$ 3 个数组和 d, e 两个变量即可。对于 n 比较大的方程组, 这种存储方式能够极大地降低存储量。方程组的右边可将 $6V$ 用一个数组 r 存储, 此时的方程组可表示为

$$AP = R \Rightarrow \begin{pmatrix} b_1 & c_1 & & & & e \\ a_1 & b_2 & c_2 & & & \\ & 1 & & & & \\ & & \dots & & & \\ & & & a_{n-3} & b_{n-2} & c_{n-2} \\ & & & & a_{n-2} & b_{n-1} & c_{n-1} \\ d & & & & & a_{n-1} & b_n \end{pmatrix} \begin{pmatrix} P_0 \\ P_1 \\ \vdots \\ P_{n-1} \end{pmatrix} = \begin{pmatrix} r_n \\ r_1 \\ \vdots \\ r_{n-1} \end{pmatrix} \quad (3)$$

解线性方程组一般使用 LU 分解的方法, 该方法在 n 比较大的情况下, 比传统的 Gauss 消去法要省去一半的时间。常见的专用 LU 分解为追赶法, 针对的是形式如下的主对角线占优的三带状矩阵

$$A = \begin{pmatrix} b_1 & c_1 & & & & e \\ a_1 & b_2 & c_2 & & & \\ & \dots & & & & \\ & & a_{n-3} & b_{n-2} & c_{n-2} & \\ & & & a_{n-2} & b_{n-1} & c_{n-1} \\ & & & & a_{n-1} & b_n \end{pmatrix} \quad (4)$$

该矩阵使用追赶法进行 LU 分解, 分解后的 L, U 矩阵有如下的形式

$$L = \begin{pmatrix} 1 & & & & & \\ f_1 & 1 & & & & \\ & f_2 & 1 & & & \\ & & \dots & & & \\ & & & f_{n-2} & 1 & \\ & & & & f_{n-1} & 1 \end{pmatrix}$$
$$U = \begin{pmatrix} h_1 & c_1 & & & & \\ & h_2 & c_2 & & & \\ & & \dots & & & \\ & & & h_{n-1} & c_{n-1} & \\ & & & & h_n & \end{pmatrix}$$

这种专用的 LU 分解运算效率高, 存储空间小。观察方程组(3)的系数矩阵, 只比三带状矩阵(4)多了 d, e 两个变量, 因此尝试推导类似的分解方式。考虑 L, U 的形式大体不变, 两个矩阵对角线上的两行均不变, 则通过计算对矩阵进行修正调整。根据矩阵特点判断, 此时的 L, U 矩阵应比追赶法的 L, U 矩阵的非零元素多, 因此通过添加非零元的方式进行矩阵调整。首先由 L 的第一行乘 U 的最后一列得到 e , 推知需要在 L 的第一行或者 U 的最后一列添加非零元素, 考虑 L 为下三角矩阵, U 为上三角矩阵, 因此在 U 的最后一列添加非零元素, 矩阵 U 变形为

$$U = \begin{pmatrix} h_1 & c_1 & & & & \\ & h_2 & c_2 & & & \\ & & \dots & & & \\ & & & h_{n-1} & c_{n-1} & \\ & & & & h_n & \end{pmatrix}$$

然后由 L 的第 2 行乘 U 的最后一列得到 0。由于 f_i, k 均不为 0, 所以 $f_k \neq 0$, 因此需要继续添加变量。仍然按照之前的原则在 U 的最后一列添加非零元素, 矩阵 U 变形为

$$U = \begin{pmatrix} h_1 & c_1 & & & & k_1 \\ & h_2 & c_2 & & & k_2 \\ & & \dots & & & \\ & & & h_{n-1} & c_{n-1} & \\ & & & & h_n & \end{pmatrix}$$

再由 L 第 i 行 ($2 < i < n-1$) 乘 U 的最后一列都有类似的形式, 均得到 0, 可推得 U 的最后一列均有非零元素, 矩阵 U 的最终形式为

$$U = \begin{pmatrix} h_1 & c_1 & & & k_1 \\ & h_2 & c_2 & & k_2 \\ & & & \dots & \\ & & & c_{n-2} & k_{n-2} \\ & & h_{n-2} & h_{n-1} & k_{n-1} \\ & & & & h_n \end{pmatrix} \quad (5)$$

同理,由 L 的最后一行乘 U 的每一列,根据矩阵特点进行推导,在 L 的最后一行添加非零元素,最终得到 L 的矩阵形式为

$$L = \begin{pmatrix} 1 & & & & \\ f_1 & 1 & & & \\ & f_2 & 1 & & \\ & & & \dots & \\ & & & & f_{n-2} & 1 \\ g_1 & g_2 & & g_{n-2} & f_{n-1} & 1 \end{pmatrix} \quad (6)$$

矩阵 $U(5)$ 比之前多了一列,矩阵 $L(6)$ 比之前多了一行,结果接近追赶法的分解形式,求解过程应该会比较简单。此时方程组 $AP=R$ 变换为 $LUP=R$,令 $UP=Y$,则 $LY=R$ 。求解控制点方程组的过程为先将矩阵 A 分解为 L 、 U ,由 $LY=R$ 解得 Y ,再由 $UP=Y$ 解得控制点矩阵 P 即可。

下面基于该 L 、 U 结构,参照追赶法,进行专用LU分解算法的推导。

2.2 专用的LU分解

对闭合B样条曲线控制点方程(3)系数矩阵的LU分解过程进行推导,推导步骤如下:

(1)建立矩阵 L 、 U 存储。 L 由数组 $\{f_i\}(i=1,2,\dots,n-1)$ 、 $\{g_i\}(i=1,2,\dots,n-2)$ 存储, U 由数组 $\{h_i\}(i=1,2,\dots,n)$ 、 $\{k_i\}(i=1,2,\dots,n-1)$ 存储, $\{c_i\}(i=1,2,\dots,n-2)$ 仍为 A 中的值。

(2)求解得到 $\{h_i\}(i=1,2,\dots,n-1)$ 、 $\{f_i\}(i=1,2,\dots,n-2)$ 。由 L 的第一行乘 U 的第一列得 $h_1=b_1$ 。由 L 的第二行乘 U 的第一列得 $f_1h_1=a_1$,因此 $f_1=a_1/h_1$ 。由 L 的第二行乘 U 的第二列得 $f_1c_1+h_2=b_2$,因此 $h_2=b_2-f_1c_1$ 。之后几行类似,推导求解过程如下:

$$\begin{aligned} h_1 &= b_1 \\ \text{for } i &= 1, 2, \dots, n-2 \\ \{ \end{aligned}$$

$$\begin{aligned} f_i &= a_i/h_i \\ h_{i+1} &= b_{i+1}-f_i c_i \end{aligned}$$

$\}$

这段算法的运算复杂度约为 $O(2n)$ 。

(3)求解得到 $\{g_i\}(i=1,2,\dots,n-2)$ 、 f_{n-1} 。由 L 的最后一行乘 U 的第1列得 $g_1h_1=d$,因此 $g_1=d/h_1$ 。由 L 的最后一行乘 U 的第2列得 $g_1c_1+g_2h_2=0$,因此 $g_2=-g_1c_1/h_2$ 。之后几行类似,推导求解过程如下:

$$\begin{aligned} g_1 &= d/h_1 \\ \text{for } i &= 1, 2, \dots, n-3 \\ \{ \\ g_{i+1} &= -g_i c_i / h_{i+1} \end{aligned}$$

$\{g_i\}$ 数组求解完毕。接下来由 L 的最后一行乘 U 的倒数第2行,得 $g_{n-2}c_{n-2}+f_{n-1}h_{n-1}=a_{n-1}$,因此 $f_{n-1}=(a_{n-1}-g_{n-2}c_{n-2})/h_{n-1}$ 。至此 $\{f_i\}$ 数组求解完毕。这段算法的运算复杂度约为 $O(n)$ 。

(4)求解得到 $\{k_i\}(i=1,2,\dots,n-1)$ 、 h_n 。由 L 的第一行乘 U 的最后一列得 $k_1=e$ 。由 L 的第二行乘 U 的最后一列得 $f_1k_1+k_2=0$,因此 $k_2=-f_1k_1$ 。之后几行类似,推导求解过程如下:

$$\begin{aligned} k_1 &= e \\ \text{for } i &= 1, 2, \dots, n-3 \\ \{ \\ k_{i+1} &= -f_i k_i \end{aligned}$$

由 L 的倒数第二行乘 U 的最后一列得 $f_{n-2}k_{n-2}+k_{n-1}=c_{n-1}$,因此 $k_{n-1}=c_{n-1}-f_{n-2}k_{n-2}$ 。至此 $\{k_i\}$ 数组求解完毕。

由 L 的最后一行乘 U 的最后一列得

$$\sum_{i=1}^{n-2} g_i k_i + f_{n-1} c_{n-1} + h_n = b_n$$

因此, $h_n = b_n - \sum_{i=1}^{n-2} g_i k_i - f_{n-1} c_{n-1}$ 。

至此 $\{h_i\}$ 数组求解完毕,这段算法的运算复杂度约为 $O(n)$ 。

此时系数矩阵的LU分解完毕,接下来代入 L 、 U 矩阵的分解结果,进行方程组的优化求解。

2.3 方程组的优化求解

输入:型值点 $V_1, V_2, \dots, V_n (V_i \neq V_j, i \neq j)$ 。

(1)由 $r_i = 6V_i (i=1,2,\dots,n)$ 得到数组 $\{r_i\}$ 。建立矩阵 A 由3个数组两个变量存储,数组 $\{a_i\}$ 初始化全为1,数组 $\{b_i\}$ 初始化全为4,数组 $\{c_i\}$ 初始化全为1,变量 d 、 e 均为1。

(2)将求得的矩阵 L 代入方程 $LY=R$,求解 Y 。 Y 由数组 $\{y_i\}(i=0,1,\dots,n-1)$ 存储。由 L 的第1行乘 Y 得 $y_0=r_n$ 。由 L 的第2行乘 Y 得 $f_1y_1+y_2=r_2$,因此 $y_2=r_2-f_1y_1$ 。之后几行类似,推导求解过程如下:

$$\begin{aligned} y_0 &= r_n \\ \text{for } i &= 0, 1, \dots, n-1 \\ \{ \\ y_{i+1} &= r_{i+1} - f_{i+1} y_i \end{aligned}$$

由 L 的最后一行乘 Y ,解得

$$y_{n-1} = r_{n-1} - \sum_{i=0}^{n-2} g_{i+1} y_i - f_{n-1} y_{n-2}$$

至此 Y 求解完毕,这段算法的运算复杂度约为 $O(n)$ 。

(3)将求得的 Y 代入 $UP=Y$,求解 P 。由 U 的最后一行乘 P 得 $h_n p_{n-1} = y_{n-1}$,因此 $p_{n-1} = y_{n-1}/h_n$ 。由 U 的倒数第2行乘 P 得 $h_{n-1} p_{n-1} + k_{n-1} p_{n-2} = y_{n-2}$,因此 $p_{n-2} = (y_{n-2} - k_{n-1} p_{n-1})/h_{n-1}$ 。由 U 的倒数第3行乘 P 得 $h_{n-2} p_{n-2} + c_{n-2} p_{n-3} + k_{n-2} p_{n-1} = y_{n-3}$,因此 $p_{n-3} = (y_{n-3} - c_{n-2} p_{n-2} - k_{n-2} p_{n-1})/h_{n-2}$ 。之后几行类似,推导求解过程如下:

$$\begin{aligned} p_{n-1} &= y_{n-1}/h_n \\ p_{n-2} &= (y_{n-2} - k_{n-1} p_{n-1})/h_{n-1} \\ \text{for } i &= n-3, n-2, \dots, 0 \\ \{ \\ p_i &= (y_i - c_{i+1} p_{i+1} - k_{i+1} p_{i+2})/h_{i+1} \end{aligned}$$

得到控制点集 P_0, P_1, \dots, P_{n-1} ,这段算法的运算复杂度为 $O(n)$ 。根据 $P_n = P_0, P_{n+1} = P_1$,得到整个控制点集。

输出: 控制点集 $P_0, P_1, \cdots, P_n, P_{n+1}$.

3 算法的存储效率及运算复杂度

3.1 存储空间分析

普通的 n 维方程组的存储空间为 n^2+2n , 优化后的 LU 分解由 $\{a_i\}, \{b_i\}, \{c_i\}, \{f_i\}, \{g_i\}, \{h_i\}, \{k_i\}, \{r_i\}, \{y_i\}$ 9 个数组和 d, e 两个变量即可, 总的存储空间为 $9n$. 如果再考虑 a 和 c 中存的全是 1, b 中存的全是 4, 因此 a, b, c 这 3 个数组可以直接用 3 个变量来存储, 这样该算法的存储空间就降到了 $6n$, 可见优化后的存储效率得到了很大提高.

3.2 运算复杂度分析

专用的 LU 分解的运算复杂度主要由文中的 (2)~(4) 步和文中的 (2)~(3) 步的运算复杂度决定, 这几部分的算法复杂度之和为 $O(2n+n+n+n+n) = O(n)$, 普通的 n 维方程组求解计算量为 $O(n^2)$, 可见专用的 LU 分解能够降低计算量, 适用于实时性要求比较高, 数据量比较大的情况.

使用 VS2005 编程, 对专用的 LU 分解过程与普通的 LU 分解的效率进行测试. 对 n 取不同的数值时的 n 维方程组进行求解, 分别记录普通的 LU 分解和专用的 LU 分解的计算时间, 试验结果如表 1 所示.

表 1 普通 LU 分解和优化的 LU 分解的时间比较

n	8	40	80	200
普通 LU 分解	0.0031ms	0.1968ms	1.275ms	17.2ms
专用的 LU 分解	0.000219ms	0.001547ms	0.003062ms	0.046ms

可见专用的 LU 分解速度得到了较大的提高. 随着 n 的增大专用的 LU 分解的优势就更加明显, 当绘制等值线时, 面对大量数据, 该优化的方法更能满足实时性的要求.

4 数值试验

试验环境使用三维图形引擎 OSG(open scene graph)2.8, 用 VS2005 的 VC++ 编程. 配置为 Intel Core 2 Quad CPU, 3GB 内存. 使用本文的算法对闭合曲线进行插值, 通过实验验证该算法的可靠性和有效性.

4.1 试验 1

我们选择三维空间中不共面的 5 点做实验, 输入的原始数据 V_i 如表 2 所示, 试验结果如图 1 所示. 图 1 中的粗线为 P_i 控制点的连线, 细线为生成的闭合曲线.

4.2 试验 2

接下来我们选择位于一个平面上的 12 个点, 输入的原始数据 V_i 如表 3 所示. 试验结果如图 2 所示, 图 2 中的粗线为生成的闭合曲线, 细线为原始数据 V_i 的连线.

通过这两个试验证明, 对于非共面和共面的型值点集都能生成较好的闭合曲线, 曲线的显示效果比较令人满意.

5 应用实例

以电离层总电子含量等值线图为例, 使用本文的算法进行闭合曲线的光滑. 原始数据使用中国地区的电离层 TEC 数据, 经度范围: 70~140 度, 纬度范围: 0~60 度, 经纬度的间隔均为 2.5 度的标准网格数据.

表 2 第 1 组试验数据

	V_1	V_2	V_3	V_4	V_5
x	-1.0	-1.0	1.0	1.0	0.2
y	1.0	-0.5	-1.0	1.0	0.8
z	0.0	0.0	1.0	0.0	0.0

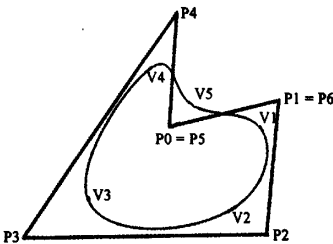


图 1 第 1 组数据插值的结果

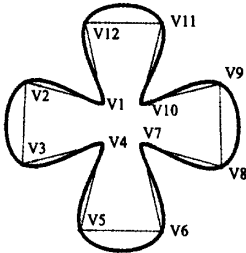


图 2 第 2 组数据插值的结果

5.1 光滑过程如下

(1) 输入跟踪得到的等值线列表, 对每条等值线分别进行光滑. 判断是否为闭合等值线, 若是则进入 (2), 否则进行非闭合曲线的光滑.

(2) 把等值线上的点作为输入点 $input$, 建立控制点集 $control$, 曲线点集 $curve$. 设置每段曲线上的细分点数 m 为 10, m 值可根据绘制要求进行设置. 判断等值线点数是否小于 3, 若是则直接绘制, 否则进入 (3) 进行光滑处理.

(3) 使用本文提出的快速算法求解控制点, 分别输入 $input$ 中的 X, Y, Z 分量, 求解得到控制点集 $control$ 的 X, Y, Z 分量.

(4) 由控制点插值得到光滑后的曲线. 构造 $0, 1, \cdots, n-1$ 段曲线, 每段曲线 $V_i V_{i+1}$ 由 4 个控制点 $P_{i-1}, P_i, P_{i+1}, P_{i+2}$ 插值获得, 构造最后一段曲线, 由 4 个控制点 $P_{n-1}, P_n(P_0), P_{n+1}(P_1), P_2$ 插值获得, 生成的点存入 $curve$.

(5) 连接 $curve$ 中的点得到光滑后的闭合曲线.

5.2 光滑前后效果对比

电离层总电子含量图抽取后直线连接的效果如图 3 所示, 使用三次均匀 B 样条曲线光滑后的效果如图 4 所示.

经过 B 样条曲线光滑插值后的等值线, 曲线变得光滑, 能够更好的体现原数据变化, 细节部分可以更好地看出光滑前后的效果, 如图 5 和图 6 所示.

6 实验结果分析

通过分析和试验可看出, 专用的 LU 分解比普通的 LU 分

表3 第2组试验数据

	V_1	V_2	V_3	V_4	V_5	V_6	V_7	V_8	V_9	V_{10}	V_{11}	V_{12}
x	-0.1	-0.5	-0.5	-0.1	-0.2	0.2	0.1	0.5	0.5	0.1	0.2	-0.2
y	0.1	0.2	-0.2	-0.1	-0.5	-0.5	-0.1	-0.2	0.2	0.1	0.5	0.5
z	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0

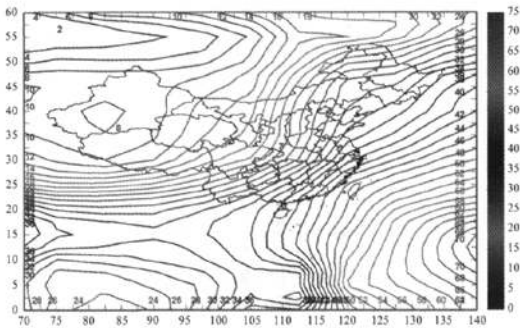


图3 电离层总电子含量图光滑前

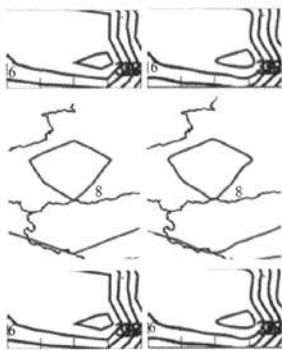


图5 电离层总电子含量图光滑前后细节图(1)

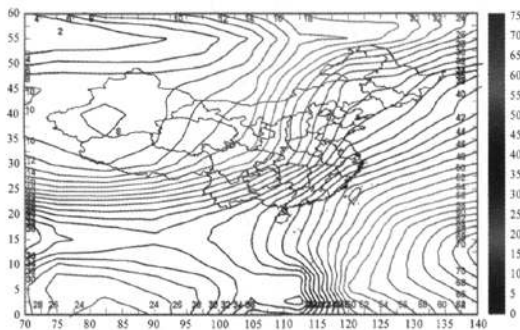


图4 电离层总电子含量图光滑后

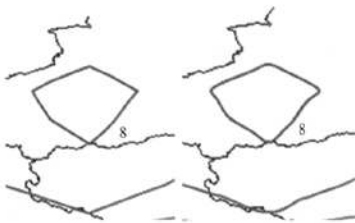


图6 电离层总电子含量图光滑前后细节图(2)

解降低了存储量,提高了计算效率,更能满足实时性的要求。基于该LU分解推导了求解闭合B样条曲线控制点的快速算法,通过数值试验,验证了该算法的可靠性和有效性。应用实例中,针对电离层总电子含量等值线图对实时性要求高的特点,使用该快速求解控制点的算法进行等值线光滑,通过光滑结果可看出该算法效果较好。

7 结束语

本文对闭合B样条曲线反求控制点的矩阵进行分析,给出了专用的LU分解算法。该算法适合于具有该结构的矩阵的LU分解,算法的效率相比于普通的LU分解有很大提高。基于该LU分解算法推导了求解闭合B样条曲线控制点的快速算法,适合点数大于3的闭合曲线的构造,计算过程简单。通过数值试验和应用实例验证,该算法计算效率高,存储空间小,更能满足实时性的要求。可用于等值线光滑等大数据量情形下的闭合曲线生成。

参考文献:

[1] 谢志鹏,施建文.一种基于矩阵LU分解的分段B样条插值法[J].计算机与数字工程,2006,34(3):62-72.

[2] 吴光亚,王小华.反求三次B样条曲线控制顶点的一种快速算法[J].杭州电子科技大学学报,2005,25(3):64-66.

[3] 柴本成,李继芳,张怡芳.开放均匀B样条曲线反算的一种通用算法[J].计算机工程与设计,2007,28(18):4429-4474.

[4] 李学艺,王钊,连小珉,等.基于周期性延伸的三次B样条闭曲线插值[J].农业机械学报,2009,40(12):251-254.

[5] 李志明,檀结庆.有理三次样条的误差分析及空间闭曲线插值[J].计算机辅助设计与图形学学报,2008,20(7):876-881.

[6] Schaefer S, Ju T, Warren J. A unified, integral construction for coordinates over closed curves[J]. Computer Aided Geometric Design, 2007, 24(8-9): 481-493.

[7] Sarfraz M, Hussain M Z. Data visualization using rational spline interpolation[J]. Journal of Computational and Applied Mathematics, 2006, 189(1/2): 513-525.

[8] 李志明.有理三次样条及空间闭曲线插值问题的研究[D].合肥工业大学,2008.

[9] 谭德松.基函数法构造插值样条曲线曲面[D].湖南师范大学,2007.

[10] OpenSceneGraph [EB/OL]. <http://www.openscenegraph.org>, 2010-01-05.

[11] Samuel R Buss. 3D 计算机图形学[M]. 唐龙,译.北京:清华大学出版,2006:182-211.