文章编号:1007 - 6492(2000)01 - 0083 - 04

代数剪切法:一种曲线与曲面快速求交方法

李 睿,李晶晶,刘德平

(郑州工业大学机械与电子工程学院,河南 郑州 450002)

摘 要:在计算机图形学和几何造型中,参数曲线曲面或代数曲线曲面求交是一基本问题.为提高曲线曲面求交的速度,结合代数法、求交的矩阵模式、幂迭代和 Bézier 曲线曲面的几何性质,提出了一种新的基于代数逼近和特征值理论的代数剪切法.在代数剪切法的结果算法中,仅需相交区域内的特征值参与运算.此算法可剪切区间并快速收敛于交点,与已有的算法相比,代数剪切法有较高的效率和准确性.

关键词:交点;代数剪切法;曲线;曲面;射线跟踪法;消元式;剪切

中图分类号:TP 391.72 **文献标识码**:A

0 引言

曲线曲面求交是计算机图形学和几何造型中的基本问题.本文仅讨论有限维求交,如曲线与曲面求交、曲线与曲线求交及射线与曲面求交.

计算交点的 3 个主要算法是分割算法、计算分割区域大小法及代数方法[1].基础的分割算法运用了曲线曲面表示的几何特性,给出两条 Bézier 曲线,通过比较它们控制多面体的凸包进行求交.若凸包相互不覆盖则曲线不相交;否则,分割曲线,再判断所得的凸包有无相交,每一步迭代中,算法将剪去那些不包含交点的曲线区域.简单的分割算法在区间内线性收敛,运用 Bézier 剪切可提高收敛速度.计算分割区域与分割算法相似.曲线用纵向和横向切线分割成区间并计算其矩形包围盒.分割就相当于计算每一分割区域的中点坐标并确定最终的矩形区域.代数法以求解方程组的形式解决求交问题,对于低阶曲线曲面求交(包括三阶或四阶),求隐式方程是最快的算法之一;而求解高阶多项式会导致数值不稳定,求交中算法也会因此而不准确.此外,展开行列式并存储到计算机很浪费计算机资源.

本文介绍了基于求交的特征值理论的代数剪切法,它的特别之处在于运用了 Bézier 曲线曲面性质、矩阵结构和反幂迭代,在求交过程中仅需计算区域内的结果.总算法收敛于区域内的交点,并同时剪切区域内不包含交点的区间.它的收敛速度和可行性与区域内的交点个数有关,在仅包含少数交点的求交问题中算法运行效率很高,且在实际应用中比其他方法有明显的优越性.

1 曲线与曲面求交

本文仅考虑有理参数曲线曲面及代数曲线曲面的求交,其中包括 Bézier 曲线曲面、NURBS 曲线曲面、二次曲面片等[2].

本文中的许多算法假定以多项式的幂级数表示,而不是伯恩施坦基形式.从伯恩施坦基形式转化为幂级数形式,需要经过一个线性变换,而这个变换会导致数值问题.为解决这一问题,对张量积曲面可按以下形式再进行一次参数化:在结果式中进行 $s' = s/(1-s)$,$t' = t/(1-t)$ 的替换,则最终的参数化是关于 s',t' 的幂级数形式,曲面的限定区间也相应地转化.

1.1 求交问题与代数方程

求交问题可以转化为求解方程组[3,4].

例如,有 2 个有理 Bézier 曲线的齐次方程:$\boldsymbol{P}(s) = (X(s), Y(s), W(s))$ 和 $\bar{\boldsymbol{Q}}(t) = \bar{X}(t), \bar{Y}(t), \bar{W}(t))$,求交问题就相当于在区间内求解方程组

$$\begin{cases} X(s)\bar{W}(t) - \bar{X}(t)W(s) = 0; \\ Y(s)\bar{W}(t) - \bar{Y}(t)W(s) = 0. \end{cases} \tag{1}$$

1.2 转化为特征值问题

在求交问题中,可得到由 2 个或 3 个代数方

收稿日期:1999 - 08 - 30;修订日期:1999 - 11 - 27
基金项目:河南省科技攻关项目(971110313)
作者简介:李 睿(1974 -),女,河南省周口市人,郑州工业大学硕士研究生.

程组成的联立方程组，消元式可表示为矩阵行列式，矩阵的元素是一元多项式. 我们不把行列式展开，而是转化为特征值方程. 一般地，表示为矩阵多项式的消元式形式为

$$\boldsymbol{M}(s) = \boldsymbol{M}_n s^n + \boldsymbol{M}_{n-1} s^{n-1}(1-s) + \boldsymbol{M}_{n-2} s^{n-2} \cdot (1-s)^2 \cdots + \boldsymbol{M}_0 (1-s)^n,$$

式中：$\boldsymbol{M}_i$ 是 $m \times m$ 阶矩阵，其元素是数值；m 和 n 是曲线和曲面的次数.

等式两边同除以 $(1-s)^n$，并用 $u = s/(1-s)$ 替换，可得到矩阵多项式

$$\boldsymbol{L}(u) = \boldsymbol{M}_n u^n + \boldsymbol{M}_{n-1} u^{n-1} + \cdots + \boldsymbol{M}_0. \tag{2}$$

这样，求交算法就相当于求行列式 $|\boldsymbol{L}(u)| = 0$ 的根. 然后通过如下方法把求根问题转化为特征值问题.[5]

定理 1 设有一特征多项式 $\boldsymbol{L}(u)$，其相应行列式的根为标准化系统 $\boldsymbol{C}_1 u + \boldsymbol{C}_2$ 的特征值，其中

$$\boldsymbol{C}_1 = \begin{bmatrix} \boldsymbol{I}_m & 0 & 0 & \cdots & 0 \\ 0 & \boldsymbol{I}_m & 0 & 0 & 0 \\ \square & \square & \cdots & \square & \square \\ 0 & 0 & 0 & \boldsymbol{I}_m & 0 \\ 0 & 0 & 0 & \cdots & \boldsymbol{M}_n \end{bmatrix};$$

$$\boldsymbol{C}_2 = \begin{bmatrix} 0 & -\boldsymbol{I}_m & 0 & \cdots & 0 \\ 0 & 0 & -\boldsymbol{I}_m & \cdots & 0 \\ \square & \square & \cdots & \square & \square \\ 0 & 0 & 0 & \cdots & -\boldsymbol{I}_m \\ \boldsymbol{M}_0 & \boldsymbol{M}_1 & \boldsymbol{M}_2 & \cdots & \boldsymbol{M}_{n-1} \end{bmatrix}. \tag{3}$$

式中：0，$\boldsymbol{I}_m$ 分别是 $m \times m$ 阶零矩阵和单位矩阵.

根据方程和消元式的性质，$\boldsymbol{C}_1 u + \boldsymbol{C}_2$ 的特征值就与式(1)的未知数之一相对应. 其余变量可从相应的特征向量中求出. 对于大多数的求交问题，我们只对计算实数域有限子集中的特征向量感兴趣，如 Bézier 曲线曲面的取值区间为 $s \in [0,1]$，而 $\boldsymbol{L}(u)$ 中的变量 u 在区间 $[0, \infty]$ 中取值. 为解决这一问题，进行反替换 $u = s/(1-s)$，则矩阵束 $\boldsymbol{C}_1 u + \boldsymbol{C}_2$ 转化为 $(\boldsymbol{C}_1 - \boldsymbol{C}_2)s + \boldsymbol{C}_2$.

2　矩阵运算

$n \times n$ 阶矩阵 A 的特征值与特征向量由方程 $\boldsymbol{A}\boldsymbol{x} = \lambda_0 \boldsymbol{x}$ 定义，其中，λ_0 是特征值，$\boldsymbol{x} \neq 0$ 是特征向量. 矩阵的特征值取决于它的特征多项式，即行列式 $|\boldsymbol{A} - \lambda_0 \boldsymbol{I}|$.

2.1　幂迭代

幂迭代是计算矩阵特征值和特征向量的基本算法. 用幂迭代计算矩阵束 $\boldsymbol{C}_1 s' + \boldsymbol{C}_2$ 的最小特征值. $\boldsymbol{C}_1 s' + \boldsymbol{C}_2$ 的最小特征值就是 $(\boldsymbol{C}_1 s' + \boldsymbol{C}_2)^{-1}$ 的最大特征值. 反幂迭代不是显式地计算矩阵的逆，其结果算法如下(给定 q_0)：

for $k = 1, 2, \cdots$

求解 $(\boldsymbol{C}_1 s' + \boldsymbol{C}_2)\boldsymbol{Z}_k = \boldsymbol{C}_1 \boldsymbol{q}_{k-1}$；

$\boldsymbol{q}_k = \boldsymbol{Z}_k / \| \boldsymbol{Z}_k \|_\infty$；($\| \boldsymbol{Z}_k \|_\infty$ 是向量 $\boldsymbol{Z}_k$ 元素中的最大值)

$s_k = -(\boldsymbol{q}_k^{\mathrm{T}} \boldsymbol{C}_2 \boldsymbol{q}_k)/(\boldsymbol{q}_k^{\mathrm{T}} \boldsymbol{C}_1 \boldsymbol{q}_k)$；(用高斯消元法求 $\boldsymbol{C}_1 s' + \boldsymbol{C}_2$ 的 LU 分解)

end

向量 q_0 是随机的，反幂迭代算法的收敛速度与初始估计值 s' 及特征值与 s' 之间的距离有关. 若使 $\boldsymbol{C}_1 s' + \boldsymbol{C}_2$ 的特征值 $\lambda_1, \lambda_2, \cdots, \lambda_n$ 有序：$|s' - \lambda_1| \leqslant |s' - \lambda_2| \leqslant \cdots \leqslant |s' - \lambda_n|$，则收敛速度是 $|s' - \lambda_1| / |s' - \lambda_2|$ 的函数.

3　代数剪切

前面讨论了如何把求交问题转化为求解矩阵束 $\boldsymbol{C}_1 s + \boldsymbol{C}_2$ 的特征值问题. 下面算法的主要思想是用反幂迭代法求出区间内的特征值，并同时剪去区间内不含任何解的部分. 设区间中点 $s' \approx 0.5$ 作为初始估计值，用反幂迭代法找出与 s' 距离最近的特征值，令此特征值为 t(t 有可能是复数，这时就计算出其共轭复数对). 设初始向量 $\boldsymbol{q}_0, \boldsymbol{u}_0$ 是随机选取的，t 是离 s' 最近的特征值，则在以 $s = s'$ 为圆心，$R = |t - s'|$ 为半径的圆域内，再也没有矩阵束的其他特征值.

基于幂迭代的收敛性有以下结论：

(1) 若 $t \in [0,1]$，则 t 是一个交点，其余的未知交点可由相应的特征向量求出.

(2) 在实数域 $(s' - R, s' + R)$ 中没有其他交点.

(3) 递归地求出下列区间内的交点：若 $(s' + R) \geqslant 0$，区间为 $[0, s' - R]$；若 $(s' = R) \leqslant 1$，区间为 $[s' + R, 1]$.

因此，基于反幂迭代法，可进行求交并剪切这一区间，在剪切后的每一区间递归地使用此算法.

3.1　计算多重解

上一节重点讨论了基于反幂迭代的代数剪切算法. 幂迭代算法往往不收敛于实数解 t 或收敛速度很慢. 若不收敛，则最近的特征值是一对共轭复数；若收敛的速度很慢，则有两个或两个以上的实特征值与 s' 的距离大致相等. 本节将证明反幂迭代可同时计算出一个以上的交点，包括共轭复数对. 本文重点讨论同时计算两个解的方法，该方

法可以很容易扩展到求多个解.

设估计值 $s = s'$，令离它最近的两个特征值为 t_1, t_2，相应的特征向量为 $\boldsymbol{x}_1, \boldsymbol{x}_2$，$\boldsymbol{A} = (\boldsymbol{C}_1 s' + \boldsymbol{C}_2)^{-1}$. 根据等式，$1/(s' - t_1)$ 和 $1/(s' - t_2)$ 是 $\boldsymbol{A}$ 的最大特征值. 于是，给定随机初始单位向量 $\boldsymbol{u}_0$，按下式解出 p, q.

$$\lim_{i \to \infty} (\boldsymbol{A}^{i+2} - p\boldsymbol{A}^{i+1} - q\boldsymbol{A}^{i}) \boldsymbol{u}_0 = 0 . \tag{4}$$

则 $1/t_1$ 和 $1/t_2$ 是方程 $\lambda^2 - p\lambda - q = 0$ 的解. 离 s' 最近的两个特征值是实数还是复数就取决于 $(p^2 + 4q)$ 的符号. 令 $D = p^2 + 4q$，若 $D > 0$，则最近特征值为

$$t_1 = s' - 2.0/(p + \sqrt{D}) ;$$

$$t_2 = s' - 2.0/(p + \sqrt{D}),$$

剪切半径 R 为 $|t_1 - s'|$ 和 $|t_2 - s'|$ 中较小的那个. 若 $D < 0$，则最近特征值是一对共轭复数.

$$\mathrm{Real}(t) = s' - \frac{2p}{p^2 - D} ;$$

$$\mathrm{lmag}(t) = \frac{2\sqrt{-D}}{p^2 - D} ,$$

剪切半径 $R = \sqrt{(\mathrm{Real}(t) - s')^2 + (\mathrm{lmag}(t))^2}$.

3.2　求交算法

求交算法是结合了曲线曲面特性的代数剪切法. 设每一曲线和曲面都有一个相应的控制多面体，如 Bézier 曲线曲面的控制多面体是以控制点的形式给出. 最简单的求交就是从区间的中点开始，用幂迭代法找出最近的特征值，并剪切这一区间，对剪切后的每个区间递归地执行算法. 程序流程框图如图 1 所示.

尽管这个算法很有效，仍然可利用曲线曲面的特性将其进一步提高. 幂迭代的收敛速度与接近估计值的特征值个数有关，若区间内有多个交点，总的收敛速度就很慢. 否则，若仅有少数交点，收敛速度相当快. 因此，来用如下方式运用曲线曲面的几何特性：

(1) 计算曲线对或曲线曲面对的控制多面体之间的交点个数，这是对交点实际个数的估计.

(2) 分割曲线和曲面，使分割后的每对多面体之间最多包括一个或两个交点.

在求交算法中，反幂迭代法占有很重要的位置. 用反幂迭代法计算特征值 s_k，直到相邻的特征值相差不到一个给定公差(TOL). 整个算法的可行性和准确性也随 TOL 变化. 所以基于开始收敛的迭代次数，用以下 TOL 值([a, b]是计算特征值的区间，$\boldsymbol{u}_k$ 表示每次迭代的特征向量).

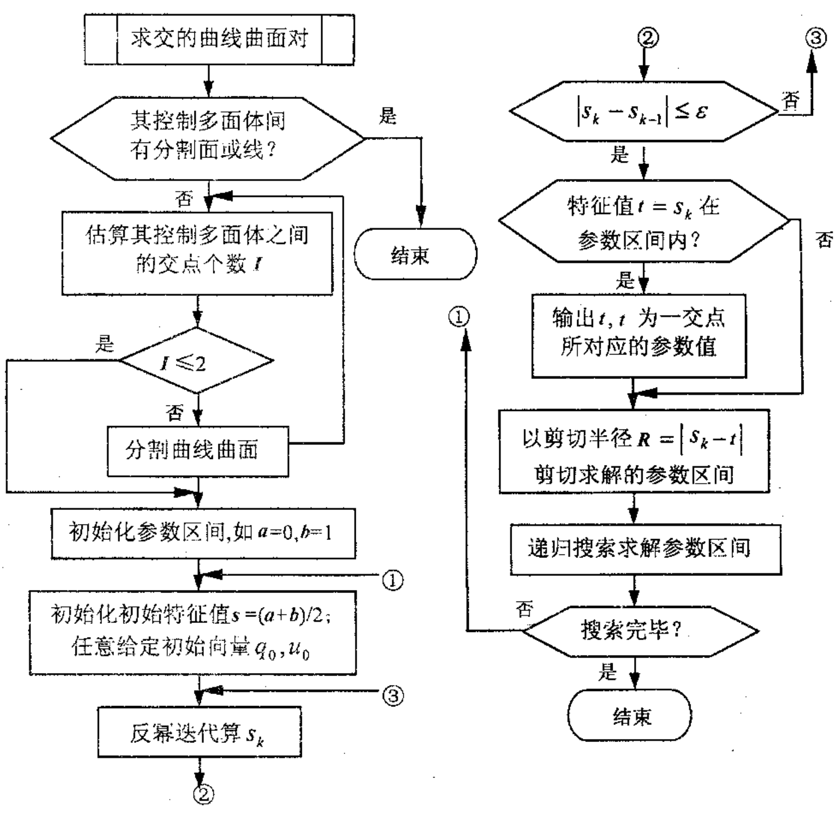

图 1　曲面曲线求交算法程序流程框图

(1) 用幂迭代法计算 $A = C_1 s' + C_2$ 的最近特征值.开始令 TOL = 0.01/($b - a$).3 次迭代后,若 $|s_3 - s_2| <$ TOL,则将 TOL 修改为 TOL = $1.0e^{-6}/(b - a)$.

(2) 若在开始的 3 次迭代中,不收敛于公差,则 TOL 不变,用多重解算法计算 p_k 和 q_k,使幂迭代收敛于复数解 s_k 或一对实数解 t_{1k} 和 t_{2k}(精确到小数点后两位数).

(3) 按相对误差计算的每一特征值 t,令 $s' = t$,TOL = $1.0e^{-6}/(b - a)$,对 $A = C_1 s' + C_2$ 执行幂迭代.

上面用到的公差可由用户自定义,变换公差的迭代次数也可改变.特征向量组 $u_k[1]$ 和 $u_k[2]$ 将用于计算曲线曲面求交中的其它变量.用代数剪切法对一条 Bézier 曲线与一 Bézier 曲面求交的结果如图 2 所示.

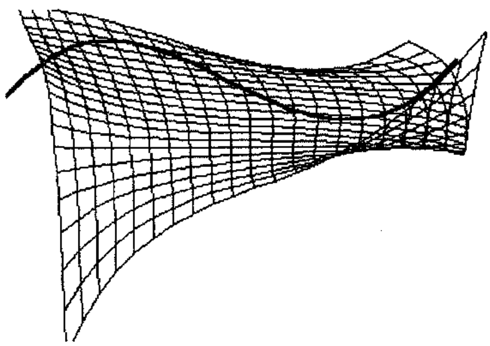

图 2 一条 Bézier 曲线与一 Bézier 曲面求交

4 结论

在代数剪切法中,由于仅计算区域内相关的结果,而不是全部解,算法的效率大大提高了.在求解特征值问题上,代数剪切法比 QR 分解快一个数量级.基于求解隐式方程的代数法中包含行列式展开的运算,且需要计算所得多项式的全部解.而代数剪切法把求解方程组的问题简化为求特征值问题,避免了行列式展开,提高了算法的效率.在与 Bézier 剪切比较的过程中可以看到,区间内仅有少数交点时,代数剪切法很快,它的收敛性比 Bézier 剪切法好,且代数剪切法有很好的空间和时序相关性,算法实现方便,可靠性好,运算速度高,是较为理想的曲线曲面求交方法.

参考文献:

[1] 孙家广.计算机图形学[M].北京:清华大学出版社,1995.

[2] 施法中.计算机辅助几何设计与非均匀有理 B 样条[M].北京:北京航空航天大学出版社,1994.

[3] 张和明,张玉云,熊光楞,等.参数曲面和平面的精确求交[J].机械工程学报,1997,33(5):33-38.

[4] MANOCHA D, DENEL J. Algorithms for intersecting parametric and algebraic curves[J]. ACM Transactions on Graphics, 1994, 13:73-100.

[5] SEDERBEG T W. Algorithms for algebraic curve intersection[J]. Computer Aided Design, 1989, 21:547-555.

Algebraic Clipping: A Fast Technique for Curve and Surface Intersection

LI Rui, LI Jing-jing, LIU De-ping

(College of Mechanical & Electronic Engineering, Zhengzhou University of Technology, Zhengzhou 450002, China)

Abstract: Computing the intersection of parametric and algebraic curves and surfaces is a fundamental problem in computer graphics and geometric modeling. To improve the speed of the intersection, we developed a new technique of a algebraic clipping based on the algebraic approaches and eigenvalue formulation of the problem, which combines with algebraic formulation, matrix formulation of the intersection problem, power iterations and geometric properties. The resulting algorithm corresponds to computing only selected eigenvalues in the domain of intersection. The algorithm proves favourable compared to earlier methods in terms of performance and accuracy.

Key words: intersection; algebraic clipping; curve; surface; ray-tracing