自交有理参数曲面网格生成

曾铮[1)], 贾晓红[1)*], 赵明阳[2,3)], 辛士庆[4)], 严冬明[2,5)]

[1)] (中国科学院数学与系统科学研究院 北京 100190)
[2)] (中国科学院自动化研究所多模态人工智能系统全国重点实验室 北京 100190)
[3)] (北京智源人工智能研究院 北京 100190)
[4)] (山东大学计算机科学与技术学院 青岛 266237)
[5)] (清华大学水沙科学与水利水电工程国家重点实验室 北京 100084)
(xhjia@amss.ac.cn)

摘 要: 针对现有网格生成算法在处理自交曲面时出现的缺少交线表示、误差大以及交线附近三角形质量差的问题, 提出一种针对自交有理参数曲面的网格生成算法. 首先, 利用动平面法计算曲面的奇异因子; 其次, 利用奇异因子和曲面的第一基本形式定位交线上的拓扑关键点; 再次, 基于动平面法设计了一种交线网格点配对生成算法, 以保证网格交线的邻域协调性; 最后, 使用基于粒子的网格生成法生成参数域网格. 在具有不同拓扑的自交曲面上进行网格生成实验, 所提算法可保证网格交线拓扑正确性, 且与未进行交线网格点配对的各类代表性各向同性网格生成算法相比, 网格三角形最小角平均值平均高 0.6%.

关键词: 自交曲面; 网格生成; 动平面法
中图法分类号: TP391.41 **DOI:** 10.3724/SP.J.1089.2023.2023-00006

Mesh Generation for Self-Intersecting Rational Parametric Surfaces

Zeng Zheng[1)], Jia Xiaohong[1)*], Zhao Mingyang[2,3)], Xin Shiqing[4)], and Yan Dongming[2,5)]

[1)] (*Academy of Mathematics and Systems Science, Chinese Academy of Sciences, Beijing* 100190)
[2)] (*State Key Laboratory of Multimodal Artificial Intelligence Systems, Institute of Automation, Chinese Academy of Sciences, Beijing* 100190)
[3)] (*Beijing Academy of Artificial Intelligence, Beijing* 100190)
[4)] (*School of Computer Science and Technology, Shandong University, Qingdao* 266237)
[5)] (*State Key Laboratory of Hydroscience and Engineering, Tsinghua University, Beijing* 100084)

Abstract: When processing self-intersecting surfaces, existing mesh generation methods encounter issues such as lacking intersection representation, large errors, and diminished triangle quality near intersections. In response to these challenges, a novel mesh generation algorithm for self-intersecting rational parametric surfaces is proposed. Firstly, the moving plane method is utilized to calculate the singular factors of the surface. Secondly, the topologically significant points on the intersections is identified by singular factors and the first fundamental form of the surface. Thirdly, a pairing algorithm for mesh vertices on intersections is developed based on the moving plane method to ensure the neighborhood coordination of mesh intersection. The mesh is finally constructed by a particle-based mesh generation method within the parameter domain. Mesh generation tests are conducted on self-intersecting surfaces with different topologies. The proposed algorithm ensures the correctness of mesh in-

收稿日期: 2023-03-31; 修回日期: 2023-09-05. **基金项目**: 国家重点研发计划(2021YFB1715900); 国家自然科学基金面上项目(12022117, 62272277, 62172415); 清华大学水沙科学与水利水电工程国家重点实验室开放基金(sklhse-2022-D-04). **曾铮**(1996—), 女, 博士, 主要研究方向为计算机辅助设计、计算机图形学; **贾晓红**(1981—), 女, 博士, 研究员, 博士生导师, CCF 会员, 论文通信作者, 主要研究方向为计算机辅助设计、计算机图形学、计算代数几何、人工智能; **赵明阳**(1993—), 男, 博士, 主要研究方向为人工智能; **辛士庆**(1979—), 男, 博士, 副教授, 博士生导师, CCF 会员, 主要研究方向为计算机图形学、计算几何; **严冬明**(1979—), 男, 博士, 研究员, 博士生导师, CCF 会员, 主要研究方向为图形/图像处理、虚拟现实、数字几何处理、模式识别.

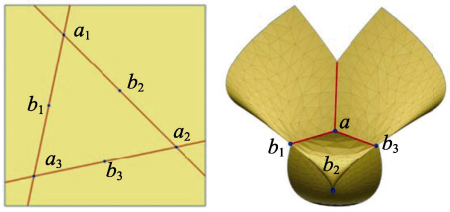
tersection topology. Compared to various typical isotropic mesh generation algorithms that do not pair mesh vertices at intersections, the mean value of minimum angles increased 0.6% on average.

Key words: self-intersecting surfaces; mesh generation; moving plane method

参数表示和网格表示都是表示三维几何形状的重要方法，它们在现代 CAD/CAE/CAM 系统中发挥着至关重要的作用，并且在各种几何建模任务中扮演关键角色[1]. 参数表示能够高效地计算出曲面上的点和线，易于理解且方便曲面形状的操纵，同时也可用于表示具有自交的曲面[2-3]. 网格表示则是进行数值计算和分析曲面性质的常用表示，也是 CAE 系统中大量使用的表示方式[4-5].

参数曲面的网格生成是几何建模和渲染应用中的基本操作[6-7]. 虽然现有的参数曲面网格生成技术种类很多，如椭球密铺法[7]、前沿推进法[8-9]、基于重心 Voronoi 图的网格生成法[10]和基于粒子的网格生成方法[11]，但是这些方法都不能有效地处理自交参数曲面. 然而，自交曲面大量存在于如医学成像以及 3D 打印和建筑设计等实际应用中. 如何有效地处理自交曲面依旧是一个具有挑战性的问题. 对于大部分应用程序而言，在处理自交曲面时保证生成网格的拓扑正确性并移除相交面片非常重要. 此外，还要求生成的三角网格分布满足各向异性或各向同性. 如图 1 所示，图 1b 中点 a 的原像为图 1a 的 a_1,a_2,a_3；图 1b 中点 b 为自交线上的非光滑点，其原像分别为图 1a 的 b_1,b_2,b_3. 点 b_1 又被称为斜切点，它不是曲面的自交点，而是自交曲线的端点. 要保证网格曲面的拓扑正确性，必须精确定位斜切点. 本文算法可以在精确定位这些特殊点的同时，生成图 1b 所示的高质量网格.

目前求解参数曲面自交线的方法主要分为 2 类：数值方法和符号计算方法. 数值方法通过曲面细分操作，将求解自交线的问题转换为求解小面片之间的相交线的问题. 这类方法的精度取决于细分面片的大小，它们无法找到小于细分面片的环状交线. 此外，数值方法也不能找到参数曲面的非光滑点，这可能导致三角化过程中出现病态三角面片. 不同于数值方法，符号计算方法通过分析参数方程生成的代数方程组来求解曲面的自交线，然后通过追踪法求出自交线上的交点. 符号计算方法能够返回自交线在参数域上的精确表示，符号计算方法中的动平面法还能返回曲面尖线在参数域上的精确表示. 然而，当参数域上的多条曲线对应曲面上的同一条交线时，分别追踪参数域上的这些曲线会得到曲面上的不同折线.



a. 参数域　　b. 双二次曲面

图 1　一个自交双二次曲面

为解决上述问题，本文基于动平面法设计网格生成算法，按先定位交线网格点后生成内部网格点的顺序来进行网格生成，算法的创新点如下：

(1) 利用曲面的第一基本形式提供了定位曲面奇异因子上非一般奇异点的算法，保证了交线拓扑的正确性.

(2) 利用动平面法给出了参数域交线上的奇异点配对算法，保证了曲面网格交线的邻域协调性，即曲面网格面片之间的交线为网格边.

(3) 在基于粒子的网格生成法中引入影子点和雅可比项，它们分别解决了边界区域和拓扑关键点处网格不符合度量的问题.

1　相关工作

1.1　网格曲面及参数曲面的交线处理

自交曲面生成的网格需要保证三角面之间的交线为网格边，且网格曲面的拓扑与参数曲面一致. 目前有 2 种途径来实现这样的网格，一种是先生成网格，再对三角面相交处进行修复；另一种是先求解网格自交线，再生成带自交线的网格.

第 1 种方法主要包括 Jung 等[12]修复了偏移网格曲面产生的自交网格，Yamakawa 等[13]通过边操作和面操作来修复自交的网格. 这些方法仅适用于去除网格的自交面，而不适用于生成自交参数曲面的三角网格. Yu 等[14]提出的同步进行网格化

和网格顶点对齐操作的网格生成方法，可以解决边界表示的曲面在边界线处相交的问题，但并不能处理曲面内的交线.

第 2 种方法可以根据求解交线的方法分为使用数值方法和使用符号计算求解交线 2 类.

使用数值方法求解自交线有 2 种方式，一种是先生成三角形网格，然后在三角形网格中检测自交曲线. Volino 等[15-16]提出了一种基于细分的算法来计算三角形网格的自交，O'Rourke 等[17]使用边界体层次结构树来计算多边形网格的自交曲线. 当曲面在交线附近相切时，这些方法会产生错误的交线拓扑. 数值方法的另一种方式是将参数域划分为无自交的小区域，然后通过搜索小区域所定义的面片之间的交线来确定自交线[18]. 特别地，该方法需要首先定位斜切点. Ho 等[19]和 Park 等[20]给出了可判定斜切点的基于细分的自交线求解方法. 数值方法可以运用于各种参数表示的曲面，并且能够快速求交，但交线的精度高度依赖细分区域的大小. 如果曲面上的交线形成一个小环，且小环的直径小于细分区域的大小，数值方法将无法找到这样的小环. 尽管一些自适应和启发式的算法可以降低丢失小环的概率，但它们都无法从本质上解决这个问题. 相比之下，如果参数曲面具有代数表示，如 NURBS 曲面等，可以利用符号计算方法来精准地定位小环，以生成正确拓扑的交线. Jia 等[21]使用动平面法来求解有理参数曲面的奇异曲线，这些曲线不仅包括了自交线，还包括了尖线. 这 2 类曲线都是网格生成过程中希望保留的曲线. 由于该方法精度高且具有数值稳定性，本文将在此工作的基础上生成自交网格，同时将解决该方法不能计算出斜切点的问题.

此外，先求解自交线再对曲面拆分进行网格生成的方法目前仍然存在 2 点缺陷. 一是拆分曲面需要进行曲面裁剪操作，由于计算机存在截断误差，共交线的 2 张曲面经过裁剪后可能会产生缝隙，因此导致生成网格在交线处也存在缝隙；二是拆分曲面后分别进行离散化会导致交线在共交线的 2 片曲面上的离散化不一致，进而导致生成网格在交线处存在重叠面、穿插面以及缝隙等错误.

1.2 参数曲面网格生成

生成给定参数曲面的三角网格是几何模型处理的基本任务之一，代表性网格生成方法可参考文献[22-24]. 现有的三角网格生成算法可以分为直接法和映射法 2 类. 直接法直接在三维空间中生成网格；映射法则在参数域上生成三角网格，然后再通过曲面的参数方程将网格映射到三维空间中.

为了直接在三维空间中生成网格，直接法需要计算三维点到曲面的投影，而投影计算对参数曲面而言比较耗时. 因此直接法需要先生成一个密集的曲面网格，然后对该网格进行重新网格化. 利用直接法生成各向同性网格的工作包括：Hoppe 等[25]通过迭代执行网格的边翻转、边分裂、边合并和顶点位置优化来生成同性网格；Yan 等[26]使用限制在曲面上的 Voronoi 图来生成各向同性网格；Zhong 等[11]使用基于粒子的方法来生成各向同性网格. 利用直接法生成各向异性网格的工作包括：Lévy 等[27]通过六维各向同性网格来生成三维空间的各向异性网格；Zhong 等[11]提出的基于粒子的方法可结合不同的黎曼度量生成不同的各向异性网格. 然而，这些方法只能生成曲面网格，而不能获得对应的参数域网格.

与直接法相比，映射法在低维参数域上进行网格生成，不需要考虑投影计算，且能同时获得参数域网格和曲面网格. 映射法现有的工作可以分为 2 类，一类工作利用共形映射对曲面进行重新参数化，并在参数域上生成各向同性网格[28-29]；这类工作通常针对三角网格曲面生成共形映射，而不是直接处理参数曲面，目前尚无法处理自交曲面. 另一类工作是计算现有参数域的度量，然后生成参数域上的各向异性网格. 二维各向异性网格可通过迭代插入网格点的方法生成[30-31]，但是由于插入网格点后需要执行多次优化，因此其计算效率较低. 基于重心 Voronoi 图的网格生成法也可用于各向异性网格生成. Du 等[10]和 Labelle 等[32]分别给出了不同的各向异性 Voronoi 单元定义，利用这些定义迭代更新顶点位置和 Voronoi 图即可生成各向异性网格. 与基于重心 Voronoi 图的网格生成法相比，基于粒子网格生成法在迭代优化过程中不必计算 Voronoi 图，而是通过分析粒子的受力情况来更新粒子位置. 基于粒子的方法在完成粒子位置优化后，仅需计算一次 Voronoi 图，然后通过其对偶来连接粒子形成三角网格，与其他方法相比效率更高. 因此，本文采用基于粒子的网格生成法生成参数域网格. 与之前工作不同的是，本文在粒子的受力分析中引入了曲面的雅可比矩阵，并利用该矩阵考虑粒子的三维受力情况，而不是估计粒子的二维受力情况，以进一步改善网格在拓扑关键点的质量.

2　预备知识

2.1　动平面法

动平面法可以高效地计算有理参数曲面的奇异点[33-34]，本节将简要介绍动平面法的背景知识.

定义 1. 动平面. 它是指一簇由参数 u,v 给出的平面

$$L(u,v;\boldsymbol{X}) \coloneqq l_0(u,v)x + l_1(u,v)y + l_2(u,v)z + l_3(u,v)w = 0.$$

其中，$l_i(u,v), i=0,1,2,3$ 为双变元多项式；$\boldsymbol{X}=(x,y,z,w)$ 为平面上的点的齐次坐标. 为了符号的简洁性，本文使用向量 $\boldsymbol{L}$ 表示动平面，即

$$\boldsymbol{L}(u,v)=\left(l_0(u,v),l_1(u,v),l_2(u,v),l_3(u,v)\right).$$

如果 2 簇动平面的表示向量线性无关，则称这 2 簇动平面线性无关.

定义 2. 有理参数曲面的齐次表示. 给定曲面的有理参数方程

$$\boldsymbol{S}(u,v)\to\left(\frac{f_0(u,v)}{f_3(u,v)},\frac{f_1(u,v)}{f_3(u,v)},\frac{f_2(u,v)}{f_3(u,v)}\right),$$

其齐次表示形式为

$$\begin{aligned}&\boldsymbol{H}:\mathbb{R}^2\to\mathbb{PR}^3\\&(u,v)\to\left(f_0(u,v),f_1(u,v),f_2(u,v),f_3(u,v)\right).\end{aligned}$$

定义 3. 伴随曲面的动平面. 如果曲面的齐次参数方程 $\boldsymbol{H}(u,v)$ 和一簇动平面 $\boldsymbol{L}(u,v)$ 满足条件

$$\boldsymbol{S}(u,v)\cdot\boldsymbol{H}(u,v)^{\mathrm{T}}=\sum_{i=0}^{3}l_i(u,v)\,f_i(u,v)=0.$$

则称动平面 $\boldsymbol{L}$ 伴随曲面 $\boldsymbol{S}$. 从几何上看，伴随曲面的动平面为一簇与曲面相交的平面，且平面 $\boldsymbol{L}(u,v)$ 过曲面点 $\boldsymbol{S}(u,v)$.

定义 4. 曲面的矩阵表示. 它可通过伴随曲面的动平面来构造. 选取伴随次数为 (d_1,d_2) 的曲面 $\boldsymbol{S}$ 的一组极大线性无关动平面 $\boldsymbol{L}_1,\boldsymbol{L}_2,\cdots,\boldsymbol{L}_k$，且它们的最大次数不超过 $(2d_1-1,d_2-1)$ 或 $(d_1-1,2d_2-1)$. 不失一般性，此处假设 $d_1\leqslant d_2$，则动平面的最大次数不超过 $(2d_1-1,d_2-1)$. 将这些动平面改写为以变元 (u,v) 的单项式为基底的形式，即

$$\left[\boldsymbol{L}_1,\cdots,\boldsymbol{L}_k\right]=\left[1,u,\cdots,u^{2d_1-1},v,uv,\cdots,u^{2d_1-1}v^{d_2-1}\right]\cdot\boldsymbol{R}(x,y,z,w).$$

其中，

$$\boldsymbol{R}=\begin{bmatrix}r_{11}&r_{12}&\cdots&r_{1k}\\\vdots&\vdots&\vdots&\vdots\\r_{q1}&r_{q2}&\cdots&r_{qk}\end{bmatrix}\tag{1}$$

为 $q\times k=2d_1d_2\times k$ 的矩阵，称为曲面 $\boldsymbol{S}$ 的矩阵表示.

引理 1. 矩阵表示的行和列满足关系 $q\leqslant k$，且对于齐次坐标表示的曲面点 $\boldsymbol{q}=(x_0,y_0,z_0,w_0)$，矩阵 $\boldsymbol{R}(\boldsymbol{q})$ 的行秩不满[33].

定义 5. 曲面的奇异点及阶数. 对曲面上的一点 $\boldsymbol{q}=(x_0,y_0,z_0,w_0)$，令 $N_{\boldsymbol{q}}$ 表示满足 $\boldsymbol{S}(u_*,v_*)=k\cdot\boldsymbol{q},k\in\mathbb{R}$ 的所有不同参数 (u_*,v_*) 的个数. 若 $N_{\boldsymbol{q}}=1$，则称 $\boldsymbol{q}$ 为简单点；若 $N_{\boldsymbol{q}}>1$，则称 $\boldsymbol{q}$ 为奇异点. 同时，称 $N_{\boldsymbol{q}}$ 为 $\boldsymbol{q}$ 的阶数，若 $N_{\boldsymbol{q}}=\infty$，称 $\boldsymbol{q}$ 为无穷阶点.

曲面点的阶和曲面矩阵表示的秩有以下关系.

引理 2. 若曲面的矩阵表示 $\boldsymbol{R}$ 为 $q\times k,q\leqslant k$ 的矩阵，则曲面点 $\boldsymbol{q}$ 为奇异点当且仅当

$$\operatorname{rank}\left(\boldsymbol{R}(\boldsymbol{q})\right)<q-1.$$

更进一步，若 $\boldsymbol{q}$ 为非孤立奇异点，则其阶为 $q-\operatorname{rank}\left(\boldsymbol{R}(\boldsymbol{q})\right)$[33].

在有理参数曲面上，曲面的奇异点集由有限个一维奇异曲线(包括自交线和尖线)分支及有限个孤立奇异点构成. 在奇异曲线上，除有限个点(非一般奇异点)外，其他奇异点的阶都等于一有限值；而孤立奇异点的阶都为无穷. 本文将提供定位奇异曲线上的非一般奇异点和孤立奇异点的算法.

定义 6. 曲面的奇异因子. 它通过曲面的矩阵表示求解. 表示曲面交线和尖线的奇异因子计算流程如下.

Step1. 将齐次坐标表示的曲面点 $\boldsymbol{q}(u,v)=\left(f_0(u,v),f_1(u,v),f_2(u,v),f_3(u,v)\right)$ 代入矩阵表示 $\boldsymbol{R}(\cdot)$，得到矩阵 $\boldsymbol{R}(\boldsymbol{q})=\overline{\boldsymbol{R}}(u,v)$.

Step2. 计算 $\overline{\boldsymbol{R}}(u,v)$ 的所有次一阶子式的行列式的最大公因子，并对该公因子进行因式分解，得到因式 $h_1(u,v),\cdots,h_l(u,v)$. 这些因式即为曲面的奇异因子，它们给出了曲面上的所有交线/点、尖线/点在参数域上的隐式方程.

2.2　曲面度量与三角网格

通过曲面度量，可以生成各种分布的三角网格. 其中大部分度量是对光滑参数曲面定义的，在自交曲面上施加这些度量会导致不稳定的网格生成过程. 下面分别介绍曲面上常用的度量种类和对应的三角网格特性. 对于 n 维欧几里得空间 $\mathbb{R}^n$，

其各向异性可由空间的内积给定.

空间中一个向量 $\boldsymbol{v}$ 的长度 $\|\boldsymbol{v}\|$ 可以通过向量内积 $\langle \boldsymbol{v},\boldsymbol{v}\rangle$ 给出, 即 $\|\boldsymbol{v}\|^2=\langle \boldsymbol{v},\boldsymbol{v}\rangle$; 2 个向量 $\boldsymbol{v},\boldsymbol{w}$ 间的角度 θ 可以通过向量的内积 $\langle \boldsymbol{v},\boldsymbol{w}\rangle$ 给出, 即 $\|\boldsymbol{v}\|\|\boldsymbol{w}\|\cos\theta=\langle \boldsymbol{v},\boldsymbol{w}\rangle$. 特别地, 欧几里得空间中的点积 $\langle \boldsymbol{v},\boldsymbol{w}\rangle_{\boldsymbol{I}}=\sum_{i=1}^{n}\boldsymbol{v}_i\boldsymbol{w}_i$ 也是一种内积, 其中 $\boldsymbol{I}$ 为对角线全为 1 的单位矩阵, 给出了各向同性的空间, 相对于点积, 其他的内积形式则给出了各向异性的空间. 在实际计算中, 本文使用 $n\times n$ 的对称正定矩阵 $\boldsymbol{M}$ 来表示和计算内积, 即 $\langle \boldsymbol{v},\boldsymbol{w}\rangle_{\boldsymbol{M}}=\boldsymbol{v}^{\mathrm{T}}\boldsymbol{M}\boldsymbol{w}$.

定义 7. 平面区域上的度量. 给定平面区域 $\boldsymbol{D}\subset\mathbb{R}^2$, $\boldsymbol{D}$ 上的度量 $\boldsymbol{M}(\cdot):\boldsymbol{D}\to S_{++}^2$ 为 $\boldsymbol{D}$ 到对称正定矩阵集合 S_{++}^2 的映射(为了符号的简洁性, 本文用在矩阵符号 $\boldsymbol{M}$ 后标注 $\boldsymbol{M}(\cdot)$ 表示这是一个到实数矩阵的映射, $\boldsymbol{M}$ 在它的像中, 后文符号同理). 由前述介绍可知, $\boldsymbol{M}(\boldsymbol{x})$ 给出了从区域 $\boldsymbol{D}$ 上任意一点 $\boldsymbol{x}$ 出发的所有向量构成的向量空间中的双线性形式.

在实际应用中, 通常还要求度量是一个光滑映射, 光滑的度量可以由平面上的 2 个互相正交的非零光滑向量场给定. 设 $\boldsymbol{U}_1(\cdot),\boldsymbol{U}_2(\cdot):\boldsymbol{D}\to\mathbb{U}^{1\times 2}$ 满足 $\langle \boldsymbol{U}_1(\boldsymbol{x}),\boldsymbol{U}_2(\boldsymbol{x})\rangle_{\boldsymbol{I}}$ 为曲面上 2 个互相正交的非零光滑向量场, 它们给定的光滑度量为

$$\boldsymbol{M}(\boldsymbol{x})=\boldsymbol{U}(\boldsymbol{x})^{\mathrm{T}}\boldsymbol{U}(\boldsymbol{x}),$$
$$\boldsymbol{U}(\boldsymbol{x})=\begin{bmatrix}\boldsymbol{U}_1(\boldsymbol{x})\\ \boldsymbol{U}_2(\boldsymbol{x})\end{bmatrix}.$$

从几何上看, 该度量将以 $\boldsymbol{x}$ 为圆心的圆形小邻域变形为以 $\boldsymbol{U}_1(\boldsymbol{x}),\boldsymbol{U}_2(\boldsymbol{x})$ 为长短轴的椭圆区域. 相反地, 对于度量 $\boldsymbol{M}$, 可以利用奇异值分解得到 $\boldsymbol{M}(\cdot)=\boldsymbol{U}(\cdot)^{\mathrm{T}}\boldsymbol{S}(\cdot)^2\boldsymbol{U}(\cdot)^{\mathrm{T}}$, 其中 $\boldsymbol{U}$ 为正交矩阵, 则 $\boldsymbol{U}(\cdot)=\boldsymbol{S}(\cdot)\boldsymbol{U}(\cdot)$.

定义 8. 符合平面度量的三角网格. 给定 $\boldsymbol{D}$ 上的度量 $\boldsymbol{M}(\cdot)$ 和三角网格 $\mathcal{M}_{\boldsymbol{D}}=\{\boldsymbol{V}_{\boldsymbol{D}},E,F\}$, 其中, $\boldsymbol{V}$, E, F 分别表示网格的顶点、边和面; 如果三角形的边长在该度量下相等, 则称三角网格符合度量. 实际计算中, 使用各边中点的度量来近似计算三角形的边长, 即对于顶点为 $\boldsymbol{V}_a,\boldsymbol{V}_b$ 的边 E_{ab}, 其平方边长为 $l_{ab}^2=\boldsymbol{v}_{ab}^{\mathrm{T}}\boldsymbol{M}_{ab}\boldsymbol{v}_{ab}$, $\boldsymbol{M}_{ab}=\boldsymbol{M}\left(\frac{\boldsymbol{V}_a+\boldsymbol{V}_b}{2}\right)$, $\boldsymbol{v}_{ab}=\boldsymbol{V}_a-\boldsymbol{V}_b$.

定义 9. 曲面的雅可比. 给定平面区域 $\boldsymbol{D}$, 定义在其上的光滑参数曲面为

$$\boldsymbol{S}(u,v)=\left(\boldsymbol{S}_x(u,v),\boldsymbol{S}_y(u,v),\boldsymbol{S}_z(u,v)\right):\boldsymbol{D}\to\mathbb{R}^3.$$

曲面的雅可比为曲面参数域到 3×2 矩阵的映射

$$\boldsymbol{J}(\cdot)=\begin{bmatrix}\frac{\partial \boldsymbol{S}_x(\cdot)}{\partial u} & \frac{\partial \boldsymbol{S}_x(\cdot)}{\partial v}\\ \frac{\partial \boldsymbol{S}_y(\cdot)}{\partial u} & \frac{\partial \boldsymbol{S}_y(\cdot)}{\partial v}\\ \frac{\partial \boldsymbol{S}_z(\cdot)}{\partial u} & \frac{\partial \boldsymbol{S}_z(\cdot)}{\partial v}\end{bmatrix}.$$

对于给定的参数 $\boldsymbol{x}=(\cdot)$, 记从 $\boldsymbol{x}$ 出发的向量空间为 $T_{\boldsymbol{x}}$, 从 $\boldsymbol{S}(\boldsymbol{x})$ 出发的、由曲面切向量

$$\left[\frac{\partial \boldsymbol{S}_x(\cdot)}{\partial u},\frac{\partial \boldsymbol{S}_y(\cdot)}{\partial u},\frac{\partial \boldsymbol{S}_z(\cdot)}{\partial u}\right],\ \left[\frac{\partial \boldsymbol{S}_x(\cdot)}{\partial v},\frac{\partial \boldsymbol{S}_y(\cdot)}{\partial v},\frac{\partial \boldsymbol{S}_z(\cdot)}{\partial v}\right]$$

张成的向量空间为 $T_{\boldsymbol{S}(\boldsymbol{x})}$. 则 $\boldsymbol{x}$ 处的雅可比矩阵 $\boldsymbol{J}(\boldsymbol{x})$ 给出了从 $T_{\boldsymbol{x}}$ 到 $T_{\boldsymbol{S}(\boldsymbol{x})}$ 的映射 $\boldsymbol{J}_{\boldsymbol{x}}(\boldsymbol{u})=\boldsymbol{J}(\boldsymbol{x})\boldsymbol{u}$. 相反地, 雅可比矩阵的伪逆 $\boldsymbol{J}(\boldsymbol{x})^{-1}$ 给出了从 $T_{\boldsymbol{S}(\boldsymbol{x})}$ 到 $T_{\boldsymbol{x}}$ 的映射.

定义 10. 曲面上的度量. 对于光滑参数曲面 $\boldsymbol{S}$, 可以通过参数域上的度量 $\boldsymbol{M}(\cdot)$ 来定义曲面上的度量, 即

$$\boldsymbol{M}_S(\cdot):\boldsymbol{D}\to\mathbb{R}^{3\times 3},$$
$$\boldsymbol{M}_{\boldsymbol{S}}(\cdot)=\left(\boldsymbol{J}(\cdot)^{\mathrm{T}}\right)^{-1}\boldsymbol{M}(\cdot)\left(\boldsymbol{J}(\cdot)^{-1}\right).$$

对曲面上一点 $\boldsymbol{S}(\boldsymbol{x})$, 度量矩阵 $\boldsymbol{M}_{\boldsymbol{S}}(\boldsymbol{x})$ 给出了 $T_{\boldsymbol{S}(\boldsymbol{x})}$ 上的双线性形式, 根据该双线性形式可以计算 $T_{\boldsymbol{S}(\boldsymbol{x})}$ 中向量的夹角和长度.

若表示曲面 $\boldsymbol{S}$ 的三角网格 $\mathcal{M}_{\boldsymbol{S}}=\{\boldsymbol{V}_{\boldsymbol{S}},E,F\}$, 对应的平面网格 $\mathcal{M}_{\boldsymbol{D}}=\{\boldsymbol{V}_{\boldsymbol{D}},E,F\}$ 符合度量曲面参数域上的度量 $\boldsymbol{M}(\cdot)$, 则称 $\mathcal{M}_{\boldsymbol{S}}$ 符合度量 $\boldsymbol{M}_{\boldsymbol{S}}(\cdot)$.

3 算法流程

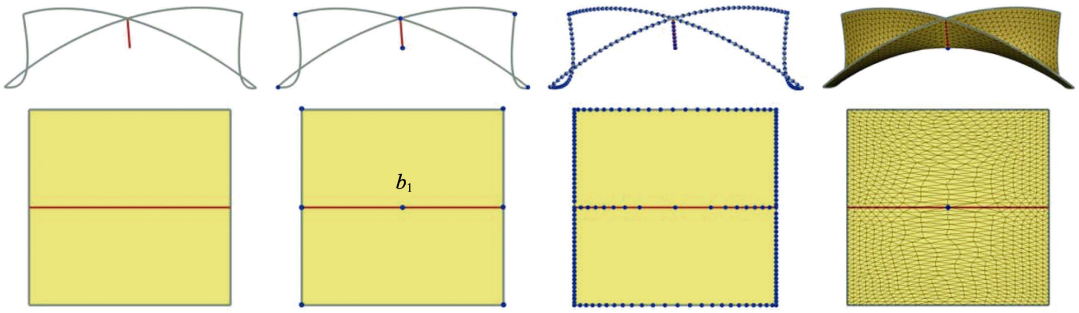
给定参数域为

$$\boldsymbol{D}=\left\{(u,v)\in\mathbb{R}^2\,|\,u_{\min}\leqslant u\leqslant u_{\max},v_{\min}\leqslant v\leqslant v_{\max}\right\}$$

的参数曲面 $\boldsymbol{S}(u,v)$, 以及曲面上的度量 $\boldsymbol{M}_{\boldsymbol{S}}(\cdot)$.

本文算法将生成符合给定度量 $\boldsymbol{M}_{\boldsymbol{S}}(\cdot)$ 的三角网格 $\mathcal{M}$, 及其对应的参数域网格 $\mathcal{M}_{\boldsymbol{D}}$.

本文算法整体流程如图 2 所示. 首先, 利用动平面法计算参数曲面的奇异因子, 它们代表了曲



a. 计算曲面的奇异曲线　b. 计算曲面的拓扑关键点　c. 追踪并匹配边界线和奇异曲线上的网格点　d. 利用基于粒子的网格生成法生成三角网格

图 2　网格生成算法流程

面的自交线和尖线；然后计算曲面上的特征点，包括奇异曲线、边界线之间的交点，以及奇异曲线上的非正则点；接着通过追踪法确定奇异曲线及边界线上的网格点，并利用动平面的矩阵表示求出参数域不同分支上互相匹配的网格点；最后利用粒子法生成曲面的三角网格.

3.1　计算拓扑关键点

曲面上的拓扑关键点包括奇异曲线、边界线之间的交点和奇异曲线上的非一般奇异点. 其中，奇异曲线上的非一般奇异点又包括奇异曲线之间的交点和非正则点. 同时考虑曲面度量在非正则点上无定义，这会导致曲面网格生成算法的不稳定，因此本文算法首先求解这些特征点.

3.2　计算奇异曲线、边界线之间的交点

通过动平面法可以求得曲面的奇异因子 $h_1(u,v),\cdots,h_k(u,v)$，将它们两两联立求解，即可求得所有奇异曲线的交点. 当3个或多个奇异因子两两联立的方程组在定义域内有相同解时，该解对应的交点即为3条或多条曲线的交点. 同理可得奇异曲线和边界线之间的交点.

3.3　计算非正则点

在非正则点处，曲面的切平面没有良定义[35].

定义 11. 非正则点. 若参数曲面 $\boldsymbol{S}(u,v)$ 上一点处 $\dfrac{\partial \boldsymbol{S}}{\partial u}$ 与 $\dfrac{\partial \boldsymbol{S}}{\partial v}$ 线性相关，则该点被称为非正则点.

非正则点可通过曲面的第一基本形式

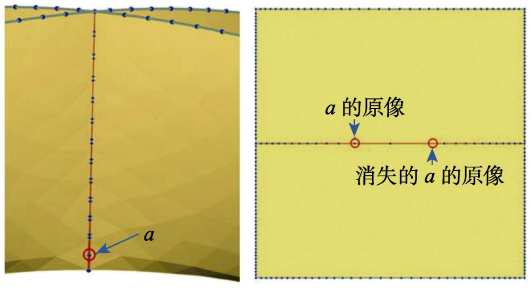
$$\boldsymbol{I}=\begin{bmatrix}\dfrac{\partial \boldsymbol{S}}{\partial u} & \dfrac{\partial \boldsymbol{S}}{\partial v}\end{bmatrix}^{\mathrm{T}}\begin{bmatrix}\dfrac{\partial \boldsymbol{S}}{\partial u} & \dfrac{\partial \boldsymbol{S}}{\partial v}\end{bmatrix}$$

的行列式的零点得到. 按照非正则点的定义，尖线上的点均为非正则点，孤立的奇异点也是非正则点，它们均可在计算奇异曲线时求出. 对于在奇异曲线上的孤立非正则点，本文算法通过联立交线的奇异因子方程和 $|\boldsymbol{I}|=0$ 得到；由此，可计算出所有非正则点，并保证曲面奇异曲线的正确拓扑.

3.4　计算奇异曲线上的网格点

在得到特征点后，对所有的奇异曲线和边界线以特征点为端点进行分割，得到曲线分支 $c_i, i=1,2,\cdots,n$，并记录每个分支 c_i 对应的奇异因子 $h_{i_j}(\cdot)$ 及端点 $\boldsymbol{p}_{i_0}$，$\boldsymbol{p}_{i_1}$；然后通过追踪法[36]得到表示每条分支 c_i 的折线分支 b_i，其折线段表示网格边，折线端点表示网格点. 追踪法得到的折线边长在给定度量下为一给定值 l，该值将用于确定网格的分辨率.

此时得到的折线分支存在2个问题，一是折线端点不一定位于曲线上；二是映射到同一交线的不同折线分支上的端点并不互相匹配. 如图 3 所示，图中蓝点为追踪法所得点，图3a交线上的点 a 在图 3b 参数域上红圈处应有 2 个原像点，追踪法只得到其中一个. 追踪法无法保证找到交线点的所有原像，且生成的交线点分布不均匀. 为解决这2个问题，本文采用下述流程对折线进行矫正、匹配和筛选.



a. 参数曲面　b. 参数域

图 3　追踪法所得交线的问题

Step1.矩阵表示 $\boldsymbol{R}(\cdot)$ 中代入

$$\boldsymbol{q}(u,v)=\left(f_0(u,v),f_1(u,v),f_2(u,v),f_3(u,v)\right),$$

得到矩阵 $\boldsymbol{R}(\boldsymbol{q})=\overline{\boldsymbol{R}}(u,v)$.

Step2. 对追踪所得的所有折线段 $b_i, i=1,2,\cdots,n$，利用该折线段对应的奇异因子 $h_{i_j}(\cdot)$ 可以精确地求解折线

端点附近交线上的点，再将精确交点带入曲面的矩阵表示即可求解交点的所有参数对. 综上，对所有折线段的每个内部端点 (u_0,v_0)，算法需求解方程组

$$\begin{cases} c_i(u_0,\bar{v})=0 \\ \boldsymbol{B}(u,v)\bar{\boldsymbol{R}}(u_0,\bar{v})=\boldsymbol{0} \end{cases}.$$

其中，$\bar{v}$, u, v 为方程组变元；$\boldsymbol{B}(u,v)=\left[1,u,\cdots,u^{2d_1-1},v,uv,\cdots,u^{2d_1-1}v^{d_2-1}\right]$ 为式(1)中的单项式基底. 该方程组关于变元 u, v 的解 $(u_i^*,v_i^*), i=1,2,\cdots,m$ 即为交点对应的所有参数对. 由此，可以求出所有折线段对应交线的原像.

Step3. 依次计算每段分支 b_i 对应奇异曲线的原像，对得到的分支原像，通过判断折线分支之间的 Hausdorff 距离来判断所得折线分支是否与已有分支相同. 若相同，则替换已有折线分支，并跳过该分支的原像计算，最终可得到互相匹配的折线分支.

3.5 基于粒子法生成三角网格

得到奇异曲线上的采样点后，使用传统网格生成算法生成三角网格. 为了同时生成参数域网格和参数曲面网格，并避免耗时的投影计算，本文采用映射法中的基于粒子的网格生成法生成网格. 首先固定前述步骤中得到网格点，使用自适应采样法[37]在定义域内部随机生成采样点，采样点的数目通过 $N=\dfrac{A-N_bS_\Delta}{2S_\Delta}-N_i$ 确定. 其中，A 为曲面在度量下的总面积；$S_\Delta=\dfrac{\sqrt{3}}{4}l^2$，$l$ 为第 3.2 节中追踪法使用的长度；N_b 为边界点数；N_i 为内部奇异曲线上的采样点数. 然后优化所有粒子的总能量，以得到满足度量的粒子分布. 为了保证定义域边界上的粒子分布符合曲面度量，本文在优化目标中引入了边界修正项. 最后使用各向异性三角网格生成法[38]连接采样点，以生成三角网格.

在基于粒子的网格生成法[11]中，粒子的总能量可根据曲面度量采用以下方式计算.

给定曲面度量 $\boldsymbol{M}_S:\boldsymbol{D}\to\mathbb{R}^{3\times3}$，一对粒子 $\boldsymbol{q}_i=(x_i,y_i,z_i)$ 和 $\boldsymbol{q}_j=(x_j,y_j,z_j)$ 之间的能量计算公式为

$$E_{ij}=\mathrm{e}^{-\frac{(\boldsymbol{q}_i-\boldsymbol{q}_j)\boldsymbol{M}_S^{ij}(\boldsymbol{q}_i-\boldsymbol{q}_j)^{\mathrm{T}}}{4\sigma^2}}.$$

其中，$\boldsymbol{M}_S^{ij}$ 为 $\boldsymbol{M}$ 在 $\boldsymbol{q}_i$ 和 $\boldsymbol{q}_j$ 的中点 $\boldsymbol{q}_{ij}$ 处的度量. $\boldsymbol{q}_i$ 和 $\boldsymbol{q}_j$ 之间的相互作用力计算公式为

$$\boldsymbol{F}_{ij}=\frac{\boldsymbol{Q}_{ij}(\boldsymbol{q}_i-\boldsymbol{q}_j)}{2\sigma^2}\mathrm{e}^{-\frac{(\boldsymbol{q}_i-\boldsymbol{q}_j)\boldsymbol{M}_S^{ij}(\boldsymbol{q}_i-\boldsymbol{q}_j)^{\mathrm{T}}}{4\sigma^2}}.$$

其中，$\boldsymbol{Q}_{ij}$ 为 $\boldsymbol{Q}$ 在 $\boldsymbol{q}_{ij}$ 处的矩阵，$\boldsymbol{Q}$ 满足 $\boldsymbol{Q}^{\mathrm{T}}\boldsymbol{Q}=\boldsymbol{M}_S$ 称为曲面度量的雅可比估计项.

利用映射法，本文在参数域上计算粒子之间的能量和相互作用力，参数域的度量 $\boldsymbol{M}$ 可通过 $\boldsymbol{M}=\boldsymbol{J}^{\mathrm{T}}\boldsymbol{M}_S\boldsymbol{J}$ 得到. 其中，$\boldsymbol{J}$ 为曲面的雅可比矩阵. 基于参数域上的度量 $\boldsymbol{M}$，一对粒子 $\boldsymbol{p}_i=(u_i,v_i)$ 和 $\boldsymbol{p}_j=(u_j,v_j)$ 之间的能量计算公式为

$$E_{ij}=\mathrm{e}^{-\frac{(\boldsymbol{p}_i-\boldsymbol{p}_j)\boldsymbol{M}_{ij}(\boldsymbol{p}_i-\boldsymbol{p}_j)^{\mathrm{T}}}{4\sigma^2}}.$$

其中，$\boldsymbol{M}_{ij}$ 为 $\boldsymbol{M}$ 在 $\boldsymbol{p}_i$ 和 $\boldsymbol{p}_j$ 的中点 $\boldsymbol{p}_{ij}$ 处的度量. 每个粒子 $\boldsymbol{p}_i$ 的总能量为 $\hat{E}_i=\sum_{j\in U_i}E_{ij}$. 其中，

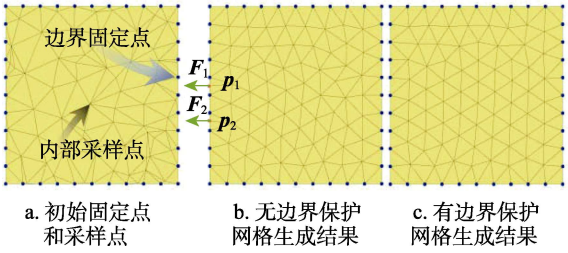
$$U_i=\left\{j\middle|(\boldsymbol{p}_j-\boldsymbol{p}_i)^{\mathrm{T}}\boldsymbol{M}_i(\boldsymbol{p}_j-\boldsymbol{p}_i)<5\sigma\right\}.$$

$\boldsymbol{S}(\boldsymbol{p}_i)$ 和 $\boldsymbol{S}(\boldsymbol{p}_j)$ 间的相互作用力计算公式为

$$\boldsymbol{F}_{ij}=\frac{\boldsymbol{Q}_{ij}\boldsymbol{J}_{ij}(\boldsymbol{p}_i-\boldsymbol{p}_j)}{2\sigma^2}\mathrm{e}^{-\frac{(\boldsymbol{p}_i-\boldsymbol{p}_j)\boldsymbol{M}_{ij}(\boldsymbol{p}_i-\boldsymbol{p}_j)^{\mathrm{T}}}{4\sigma^2}} \tag{2}$$

其中，$\boldsymbol{J}_{ij}$ 和 $\boldsymbol{Q}_{ij}$ 分别表示 $\boldsymbol{J}$ 和 $\boldsymbol{Q}$ 在 $\boldsymbol{p}_{ij}$ 处的矩阵. 每个粒子 $\boldsymbol{p}_i$ 所受合力为 $\hat{\boldsymbol{F}}_i=\boldsymbol{J}_i^{-1}\sum_{j\in U_i}\boldsymbol{F}_{ij}$. 其中，$\boldsymbol{J}_i$ 和 $\boldsymbol{M}_i$ 分别表示 $\boldsymbol{J}$ 和 $\boldsymbol{M}$ 在 $\boldsymbol{p}_i$ 处的矩阵；$U_i=\left\{j\middle|(\boldsymbol{p}_j-\boldsymbol{p}_i)^{\mathrm{T}}\boldsymbol{M}_i(\boldsymbol{p}_j-\boldsymbol{p}_i)<5\sigma\right\}$. 实验中，取 $\sigma=0.3\sqrt{|\Omega|/|P|}$. 其中，$|\Omega|$ 为 $\boldsymbol{D}$ 在度量 $\boldsymbol{M}$ 下的面积；$|P|$ 为总顶点数. 然后，用 L-BFGS 方案[39]优化总能量 $\sum_i\hat{E}_i$ 得到度量 $\boldsymbol{M}$ 下的顶点分布.

然而，在固定边界点的情况下优化总能量会导致图 4b 所示的不均匀边界. 这是因为优化过程无法限制粒子跑到边界上，且对于边界上的粒子，如图 4b 中的 $\boldsymbol{p}_1$, $\boldsymbol{p}_2$，当其所受合力 $\boldsymbol{F}_1$ 和 $\boldsymbol{F}_2$ 垂直边界指向定义域外部时，该粒子被视为优化过程中的稳定点. 为解决该问题，为每个到边界的距离为 $d_b<5\sigma$ 的粒子 $\boldsymbol{p}_i$ 添加 2 个影子点 $\boldsymbol{s}_{i1}$ 和 $\boldsymbol{s}_{i2}$，其位



a. 初始固定点和采样点　b. 无边界保护网格生成结果　c. 有边界保护网格生成结果

图 4　基于粒子的网格生成法

置如图 5 所示，图中影子点所在直线与边界的距离 $d_{\min}$ 为 $\boldsymbol{p}_i$ 与周围最近粒子 $\boldsymbol{p}_j$ 间的距离．添加影子点后，$\boldsymbol{p}_i$ 所受合力因此变为

$$\boldsymbol{F}_i = \boldsymbol{J}_i^{-1}\left(\sum_{j\in U_i}\boldsymbol{F}_{ij} + \boldsymbol{F}_{is1} + \boldsymbol{F}_{is2}\right).$$

其中，$\boldsymbol{F}_{is1}$ 和 $\boldsymbol{F}_{is2}$ 为利用式(2)所计算的粒子 $\boldsymbol{s}_{i1}$ 和 $\boldsymbol{s}_{i2}$ 与 $\boldsymbol{p}_i$ 的相互作用力．类似地，粒子 $\boldsymbol{p}_i$ 的能量为

$$E_i = \sum_{j\in U_i} E_{ij} + E_{is1} + E_{is2}.$$

通过 L-BFGS 优化 $E = \sum_i E_i$，即可得到图 4c 的均匀边界．

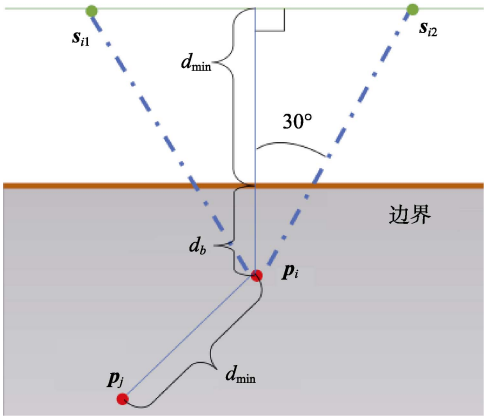


图 5　影子点分布示意图

4　实验结果

本节将使用具有不同拓扑的自交曲面进行网格生成实验并和代表性网格生成算法进行对比以展示本文算法所生成网格的优越性．在与其他算法的对比实验中，本文算法使用映射法在参数域上生成网格，而其他算法则使用直接法对空间中的密集三维网格曲面生成网格，该密集三维网格曲面与原网格面之间的 Hausdorff 距离小于 0.000 5．所有实验均在配备了 Intel i7-7700HQ 处理器的一台笔记本电脑上进行．本文算法采用 Maple 和 C++实现，同时使用了计算几何算法库 CGAL[①]和几何处理库 libigl[②]．

4.1　特征点计算示例

使用本文算法可以精准地定位奇异曲线之间的交点和奇异曲线上的非正则点，从而保障奇异曲线具有正确的拓扑．

图 1 展示了自交参数曲面

$$\boldsymbol{S}(u,v) = \left(u^2 + \frac{1}{2}v, v^2 + \frac{1}{2}u, uv + u + v\right).$$

该曲面上共有 3 条阶为 2 的自交线，和 4 个非一般奇异点 a, b_1, b_2, b_3．其中，a 为奇异曲线之间的交点，其阶为 3；b_1, b_2, b_3 为奇异曲线上的非正则点，其阶为 1．利用动平面法可求出曲面的奇异因子

$$f_1 = 2v + 2u - 1,$$

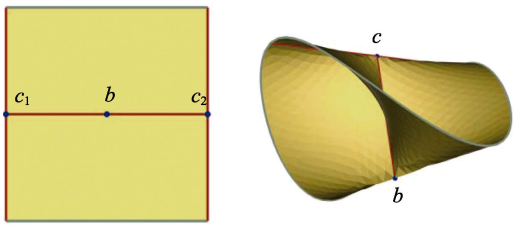
$$f_2 = 4v^2 - 20uv - 18v + 4u^2 - 18u - 27.$$

点 a_1, a_2 可通过求解 $\{f_1 = 0, f_2 = 0\}$ 得到，点 a_3 可通过求解 $\left\{f_2 = 0, \frac{\partial f_2}{\partial u} = 0, \frac{\partial f_2}{\partial v} = 0\right\}$ 得到，点 b_1, b_2 可通过求解 $\{f_2 = 0, \det(\boldsymbol{J}^{\mathrm{T}}\boldsymbol{J}) = 0\}$ 得到，点 b_3 可通过求解 $\{f_1 = 0, \det(\boldsymbol{J}^{\mathrm{T}}\boldsymbol{J}) = 0\}$ 得到．

图 6 为拥有 2 个重合非正则点 c_1 和 c_2 的曲面

$$\boldsymbol{S}(u,v) = \left(\frac{2u^3v + 2uv}{u^4 + 2u^2 + 1}, \frac{-u^4v + v}{u^4 + 2u^2 + 1}, \frac{u^4 - 2u^2 + 1}{u^4 + 2u^2 + 1}\right).$$

其中，c_1 和 c_2 映射到曲面上同一点 c．曲面上有 2 条阶为 2 的自交线，其中一条自交线上有一个阶为 1 的非正则点 b，2 条自交线交于点 c．计算可得曲面的奇异因子为 $f_1 = v$，$f_2 = u - 1$，$f_3 = u + 1$．点 c_1, c_2, b 可通过求解 $\{f_1 = 0, \det(\boldsymbol{J}^{\mathrm{T}}\boldsymbol{J}) = 0\}$ 得到．



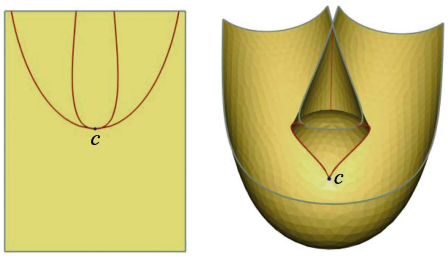
a. 参数域　　b. 曲面

图 6　一个自交 Conoid 曲面

图 7 展示了带有尖线和自交线的有理自交曲面

$$\boldsymbol{S}(u,v) = \left(\frac{uv^2 + v^2 - 12uv + 4u^3 + 4u^2 + u + 1}{v^2 + 4u^2 + 1}, \frac{u^2v^2 + 6v + 4u^4 + u^2}{v^2 + 4u^2 + 1}, \frac{6v^2}{v^2 + 4u^2 + 1}\right).$$

① https://www.cgal.org
② https://libigl.github.io



a. 参数域 b. 有理曲面

图 7 一个自交有理曲面

其奇异因子为 $f_1=4u^2v^2+v^2-12v+16u^4+8u^2+1$，$f_2=v^2-12v+4u^2+1$．2 条奇异曲线交于点 c，同时 c 也是自交线上的非正则点，c 可通过求解 $\{f_1=0,\ f_2=0,\det(\boldsymbol{J}^{\mathrm{T}}\boldsymbol{J})=0\}$ 得到．此外，验证方程组 $\{f_1=0,\det(\boldsymbol{J}^{\mathrm{T}}\boldsymbol{J})=0\}$ 的退化可知，$f_1=0$ 为尖线．

4.2 奇异曲线的精度

表 1、图 8 和图 9 对比了本文算法与 MO[25]，ACVD[40]，VASM[27]算法在 5 个具有不同拓扑的自交曲面上得到的特征曲线，这些特征曲线包括边界线、自交线和尖线．图 8 中，所对比方法的自交线和尖线分别由网格面片的交线和硬边界线给出，而本文算法可自动记录网格上的边是否属于特征线．为了比较特征曲线处的精度，在特征曲线上生成随机采样点集 P，其个数为 $|P|=1\times10^4$，并计算这些采样点到网格上的特征曲线集 C 的单边 Hausdorff 距离

$$d_{\mathrm{H}}(P,C)=\max_{\boldsymbol{p}\in P}\min_{\boldsymbol{c}\in C}\|\boldsymbol{p}-\boldsymbol{c}\|_2$$

和平均距离

$$d_{\mathrm{A}}(P,C)=\frac{1}{|P|}\sum_{\boldsymbol{p}\in P}\min_{\boldsymbol{c}\in C}\|\boldsymbol{p}-\boldsymbol{c}\|_2,$$

并将采样点的距离在图 8 和图 9 中可视化．

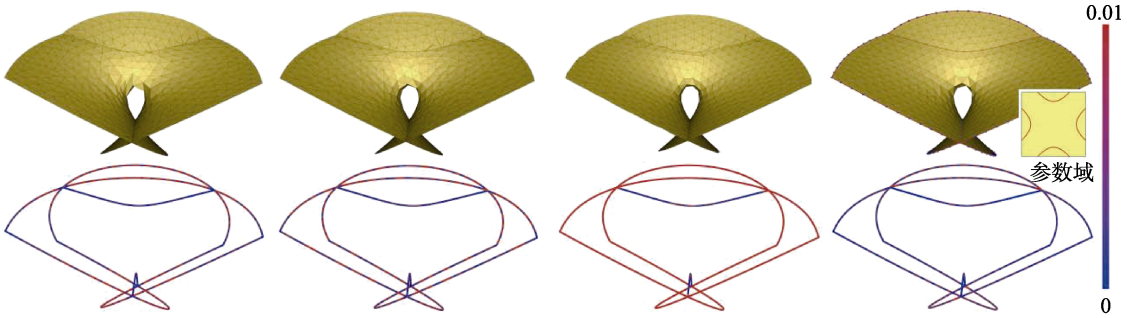
表 1 列出了本文算法与其他算法得到的网格上特征曲线的精度．从表 1 可以看出，生成的网格在顶点数更小的情况下，本文算法还能保证网格与曲面的特征曲线之间有最小的单边 Hausdorff 距离和平均距离．与其他算法相比，特征曲线方程的精确计算和特征曲线点的精准定位，可保障本文算法在特征曲线上的精度．

从图 8 和图 9 可以看出，本文算法得到的奇异曲线更接近原曲线．在边界线处，VASM 方法所得网格的边界线向曲面内部收缩了一段距离，导致边界不准确，而本文算法和其他算法都能生成准确的边界线，特别地，本文算法得到的网格边界线上顶点分布更加均匀．在奇异曲线处，仅有本文算法能保证其拓扑与原参数曲面奇异曲线的拓扑一致且具有较高的精度．对于 Rational 曲面，MO 和 ACVD 算法得到的尖线存在抖动且精度较差，并且 VASM 算法在该曲面的奇异曲线的交点处产生多余分支．对于 Bi-quadric1 曲面，由于曲面在

表 1 不同算法处理奇异曲线和边界线的精度比较

算法	Enneper			Rational			Bi-quadric1			Bi-trinomial			Bézier		
	V	d_{A}	d_{H}	V	d_{A}	d_{H}	V	d_{A}	d_{H}	V	d_{A}	d_{H}	V	d_{A}	d_{H}
MO[25]	961	0.57	1.82	1428	0.86	26.60	841	0.65	7.50	699	1.01	29.43	894	0.34	3.59
ACVD[40]	953	0.56	1.86	1438	1.67	12.77	874	0.79	17.06	656	0.75	17.96	860	0.37	4.37
VASM[27]	945	18.97	34.39	1435	13.50	29.06	927	14.90	60.69	653	30.55	134.39	860	3.84	7.61
本文	**942**	**0.34**	**1.19**	**1426**	**0.27**	**2.59**	**839**	**0.11**	**0.56**	**641**	**0.62**	**7.16**	**854**	**0.10**	**2.49**

注．V 为生成网格顶点数，d_{A} 为平均距离($\times10^{-2}$)，d_{H} 为 Hausdorff 距离($\times10^{-2}$)；粗体表示最优值．



MO[25] ACVD[40] VASM[27] 本文

图 8 Enneper 参数曲面生成的各向同性网格在自交线和边界线上的精度比较

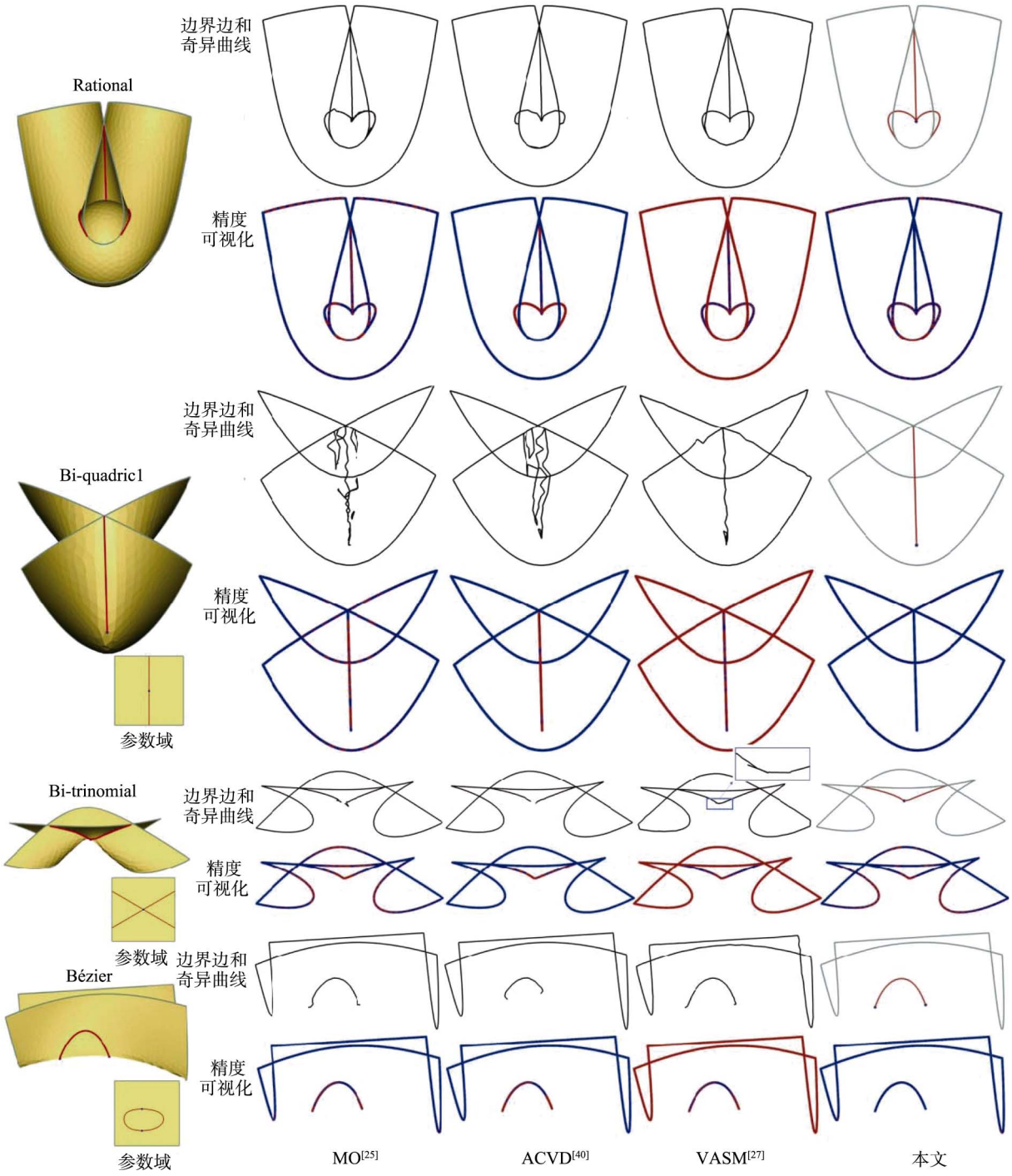


图 9　网格上的奇异曲线和边界线及精度比较

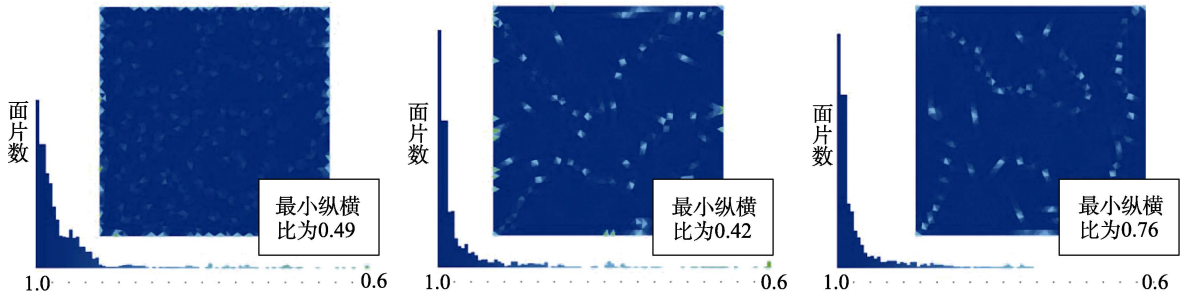
交线处相切，其他算法得到的交线会产生大量多余的环线. 对于 Bi-trinomial 曲面, VASM 算法生成的奇异曲线包含多余分支，MO 算法生成的奇异曲线则在曲线交点处断开. 对于 Bezier 曲面，仅有本文算法能保证奇异曲线在 2 个非正则点附近的精度.

4.3　边界修正项的有效性

图 10 展示了边界修正项的有效性. 图中网格为 3 种网格生成算法迭代 100 次后生成的网格，并用不同颜色编码了三角形纵横比，图片下方为不同纵横比的三角形数量统计图. 不论是其他网格生成算法还是无边界修正项的本文算法，均会在边界处生成较多的低纵横比三角形. 在有边界修正项的情况下，本文算法不仅能够降低边界区域低纵横比三角形的数量，还可以显著地提升三角形的最低纵横比.

4.4　雅可比项的有效性

Zhong 等[11]也给出了通过映射法生成网格的算法，该算法与本文算法的主要不同在于计算粒



a. MO[25]算法的结果 b. 无边界修正项的结果 c. 有边界修正项的结果

图 10 不同算法下的三角形纵横比

子间作用力的公式. Zhong 等[11]使用

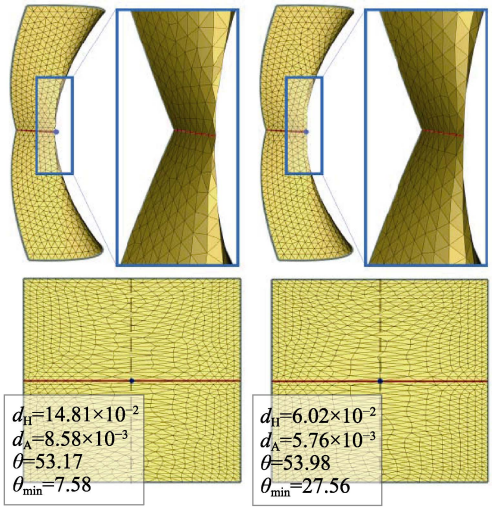
$$\hat{\boldsymbol{F}}_{ij}=\frac{\hat{\boldsymbol{Q}}_{ij}\left(\boldsymbol{p}_i-\boldsymbol{p}_j\right)}{2\sigma^2}\mathrm{e}^{\frac{\left(\boldsymbol{p}_i-\boldsymbol{p}_j\right)\boldsymbol{M}_{ij}\left(\boldsymbol{p}_i-\boldsymbol{p}_j\right)^{\mathrm{T}}}{4\sigma^2}} \tag{3}$$

计算粒子间作用力. 其中, $\hat{\boldsymbol{Q}}_{ij}$ 为 $\hat{\boldsymbol{Q}}$ 在 $\boldsymbol{p}_{ij}$ 处的矩阵, $\hat{\boldsymbol{Q}}$ 满足 $\hat{\boldsymbol{Q}}^{\mathrm{T}}\hat{\boldsymbol{Q}}=\boldsymbol{M}$. 式(2)与式(3)相比有 2 个不同点, 一是式(2)使用曲面度量的雅可比估计项 $\boldsymbol{Q}$ 而不是平面度量的雅可比估计项, 二是式(3)保留了参数曲面映射的雅可比项. 换言之, 式(2)对从参数域到曲面的切向量映射进行了估计, 而式(3)保留了正确的切向量映射.

图 11 比较了 2 种作用力生成的各向同性曲面网格的不同. 图中 θ 表示三角形最小角的平均值, $\theta_{\min}$ 表示三角形网格中的最小角. 图 11a 为式(3)的生成结果, 其非正则点附近存在细长三角形. 图 11b 为式(2)生成结果, 其非正则点附近仍保持各向同性. 从参数域上看, 本文算法在参数域上生成的粒子在非正则点附近能够沿虚线分布, 虚线方向为非正则点附近的最大曲率方向, 让粒子沿虚线分布能够有效地减少该点附近细长三角形的数量.

4.5 各向同性网格对比

表 2 将本文生成的各向同性网格在质量和精度上与其他算法(MO[25], ACVD[40], VASM[27])进行了比



a. 无雅可比项 b. 有雅可比项

图 11 不同粒子作用力生成网格的对比

较. 为了比较网格精度, 在曲面上生成随机采样点集 P, 其大小 $|P|=1\times10^6$, 然后计算这些采样点到网格 $\mathcal{M}$ 上的单边 Hausdorff 距离 $d_{\mathrm{H}}(P,\mathcal{M})=\max\limits_{\boldsymbol{p}\in P}\min\limits_{\boldsymbol{q}\in\mathcal{M}}\|\boldsymbol{p}-\boldsymbol{q}\|_2$ 和平均距离 $d_{\mathrm{A}}(P,\mathcal{M})=\frac{1}{|P|}\cdot\sum\limits_{\boldsymbol{p}\in P}\min\limits_{\boldsymbol{q}\in\mathcal{M}}\|\boldsymbol{p}-\boldsymbol{q}\|_2\sum\limits_{\boldsymbol{p}\in P}\min\limits_{\boldsymbol{q}\in\mathcal{M}}\|\boldsymbol{p}-\boldsymbol{q}\|_2$.

从表 2 可以看出, 本文算法最小角的平均值与其他算法中表现最好的 VASM 水平相当. 注意,

表 2 各向同性网格的网格质量比较

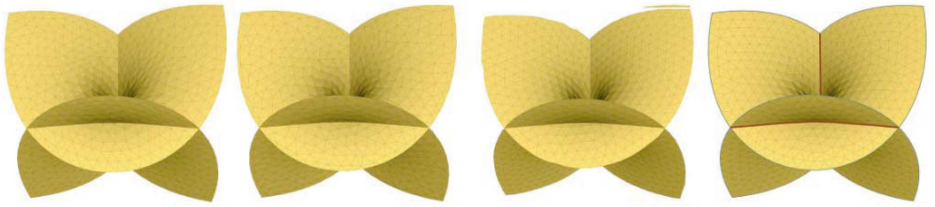
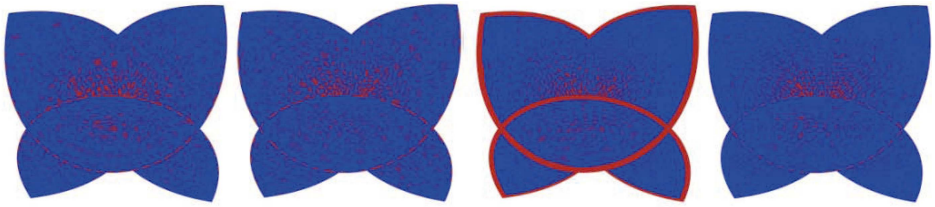
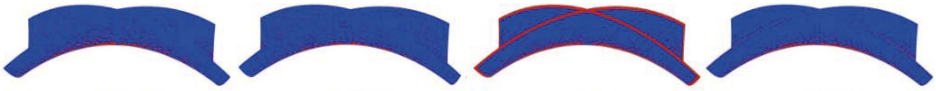
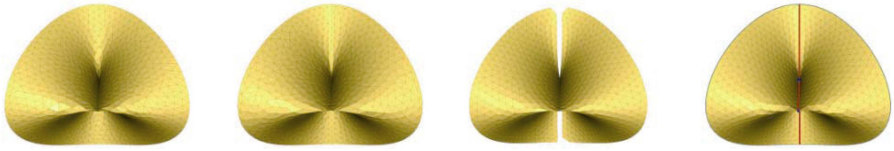
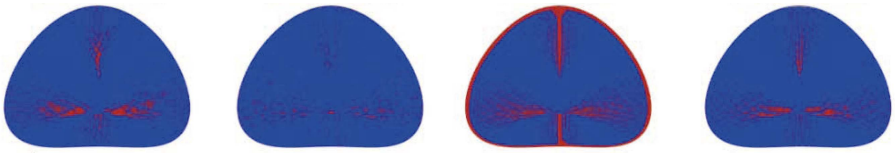
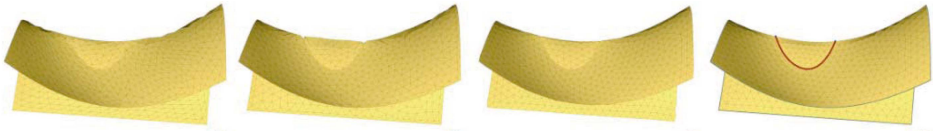
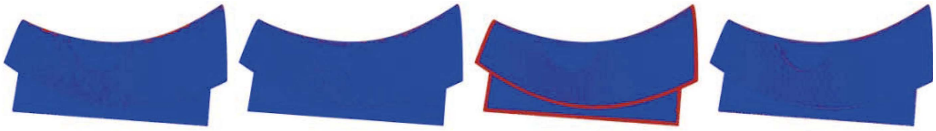
算法	Enneper				Riemann				Conoid				Bézier			
	V	d_{A}	d_{H}	θ	V	d_{A}	d_{H}	θ	V	d_{A}	d_{H}	θ	V	d_{A}	d_{H}	θ
MO[25]	961	0.69	12.65	43.36	1142	1.21	40.41	40.52	1084	0.13	2.48	39.93	894	0.21	7.36	37.87
ACVD[40]	953	**0.48**	**7.80**	45.02	944	**0.39**	8.61	44.64	945	**0.06**	1.91	44.87	860	**0.08**	2.87	44.53
VASM[27]	945	0.91	33.61	**53.57**	950	1.14	29.93	52.40	936	0.31	10.75	53.46	860	0.24	7.45	**51.71**
本文	**942**	0.52	8.20	52.96	**941**	0.58	**6.02**	**53.98**	**935**	0.11	**1.44**	**53.55**	**854**	0.13	**2.67**	51.65

注. V 为生成网格顶点数, d_{A} 为平均距离($\times10^{-2}$), d_{H} 为 Hausdorff 距离($\times10^{-2}$), θ 表示网格中所有三角形最小角的平均值; 粗体表示最优值.

VASM 算法未考虑边界限制，而其他算法在边界上会生成大量质量较差的三角形．同时本文算法生成的网格精度与其他算法中的最优精度相当，且在大部分情况下具有最小的单边 Hausdorff 距离．

表 3 展示了不同算法生成的各向同性网格．本文的网格在相交处为重边而其他算法的网格在相交处无边表示．通过可视化的精度可以看出，Riemann 和 Bézier 曲面上的非正则点会影响 MO 和 ACVD 算法生成网格的精度，而本文算法生成的网格在非正则点处的精度比其他算法更高．

表 3　各向同性网格比较

曲面	生成结果	MO[25]	ACVD[40]	VASM[27]	本文
Enneper	各向同性网格				
	曲面到网格的距离				
Riemann	各向同性网格				
	曲面到网格的距离				
Conoid	各向同性网格				
	曲面到网格的距离				
Bézier	各向同性网格				
	曲面到网格的距离				

此外，表 4 所示为本文算法的运行时间，计算其平均值，仅有 6.72%的时间被用于计算奇异曲线上的网格点．注意对于不包含非正则点的参数曲面，如 Enneper 曲面，本文的算法在基于粒子的网格生成步骤会收敛得更快．

4.6　各向异性网格对比

通过使用不同的各向异性度量，如六维各向异性和基于曲率的各向异性，本文算法可生成不同的各向异性网格．本节将对本文算法生成的 2 种各向异性网格和其他算法生成的各向异性网格

表 4 不同曲面的各向同性网格生成时间

曲面	V	T_1/s	T_2/s
Enneper	945	1.45	3.60
Riemann	950	1.34	30.65
Rational	1 426	1.75	54.08
Bi-quadric1	839	1.71	8.86
Conoid	935	1.62	38.81
Bi-trinomial	641	1.48	18.30
Bézier	854	1.88	24.66

注. V为网格顶点数; T_1为用于生成特征曲线上网格点的时间; T_2 为基于粒子的网格生成法所用时间.

(ACVD[37], EVO[41])进行精度对比.

下面简要解释六维度量. 六维各向异性[27]度量在参数域上可由 $\boldsymbol{M}=\boldsymbol{J}_{6D}^{\mathrm{T}}\boldsymbol{J}_{6D}$给出. 其中,

$$\boldsymbol{J}_{6D}=\begin{bmatrix}\dfrac{\partial \boldsymbol{S}}{\partial u} & \dfrac{\partial \boldsymbol{S}}{\partial v}\\ s\dfrac{\partial \boldsymbol{n}}{\partial u} & s\dfrac{\partial \boldsymbol{n}}{\partial u}\end{bmatrix},$$

$\boldsymbol{n}(u,v)$ 为 $\boldsymbol{S}(u,v)$ 的单位法向量; s为一自定义参数. 在使用六维度量生成网格时, 用于式(2)的矩阵 $\boldsymbol{QJ}$ 需替换为矩阵 $\boldsymbol{J}_{6D}$.

下面简要解释基于曲率的各向异性度量. 通过椭球密铺法[7]可以推导出基于曲率的各向异性在参数域上的度量, 该度量可按以下流程计算.

Step1. 通过曲面的第一基本形式 $\boldsymbol{I}$ 、第二基本形式 $\boldsymbol{\Pi}$ 计算曲面的 Weingarten 矩阵 $\boldsymbol{W}=\boldsymbol{I}^{-1}\boldsymbol{\Pi}$.

Step2. 对 $\boldsymbol{W}$ 进行奇异值分解得到 $\boldsymbol{W}=\boldsymbol{U\Sigma V}^{\mathrm{T}}$. 注意曲面的主曲率方向由矩阵 $\boldsymbol{JU}$ 的列向量给出, 主曲率则由 $\boldsymbol{\Sigma}$ 中的奇异值给出.

Step3. 将 $\boldsymbol{\Sigma}$ 中的奇异值限制在[0.000 01,10]范围内, 并计算曲面的度量 $\boldsymbol{M}_S=(\boldsymbol{U}^{\mathrm{T}}\boldsymbol{J}^{\mathrm{T}})^{-1}\boldsymbol{\Sigma}(\boldsymbol{U}^{\mathrm{T}}\boldsymbol{J}^{\mathrm{T}})$.

Step4. 定义域上的度量由 $\boldsymbol{M}=\boldsymbol{J}^{\mathrm{T}}\boldsymbol{M}_S\boldsymbol{J}$ 给出, 用于式(3)的矩阵表示为

$$\boldsymbol{Q}=\begin{bmatrix}\overline{\boldsymbol{Q}} & \\ 0\ \ 0 & 0\end{bmatrix}.$$

其中, $\overline{\boldsymbol{Q}}=\boldsymbol{U}\sqrt{\left((\boldsymbol{JU})^{\mathrm{T}}(\boldsymbol{JU})\right)^{-1}\boldsymbol{\Sigma}}(\boldsymbol{JU})^{\mathrm{T}}$. 使用度量 $\boldsymbol{M}$ 即可生成与椭球密铺法分布相同的网格. 与椭球密铺法不同的是, 椭球密铺法需要考虑椭球的大小. 因此椭球密铺法常常需要处理多余或缺失椭球, 而度量 $\boldsymbol{M}$ 仅控制了椭球的长短轴比, 椭球的大小由给定的网格点数目自适应地确定.

表 5 给出了本文算法与其他算法生成的各向异性网格在精度和用时上的比较, 对应的网格如表 6 所示.

表 5 不同方法生成的各向异性网格质量比较

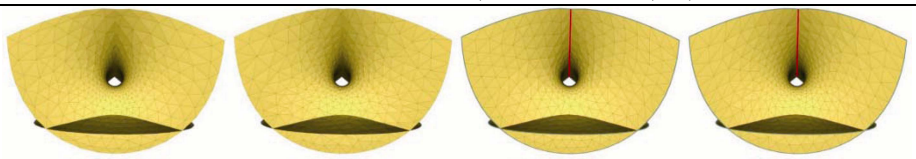
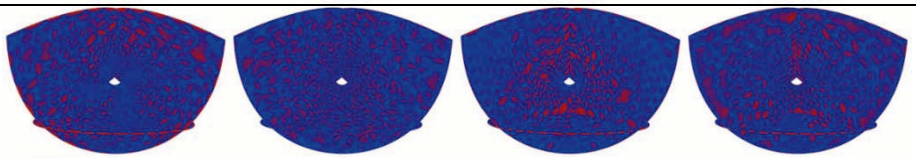
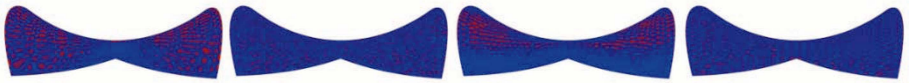
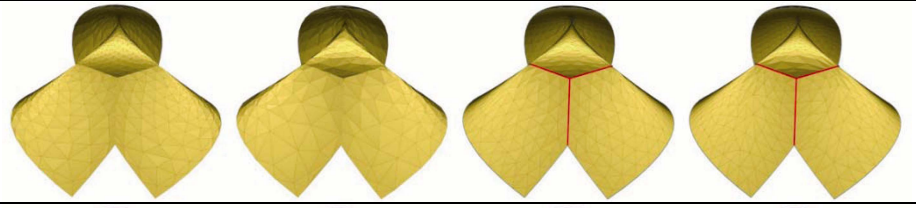
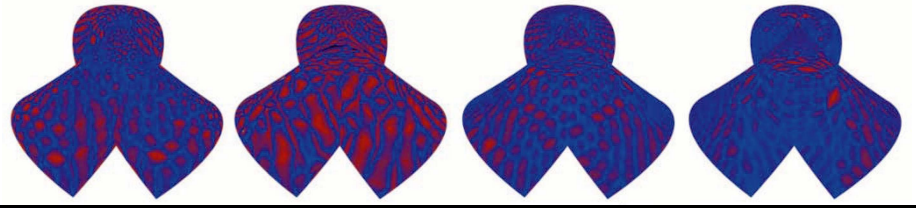
算法	Enneper				Riemann				Bi-quadratic2			
	V	d_A	d_H	T/s	V	d_A	d_H	T/s	V	d_A	d_H	T/s
ACVD[37]	981	2.44	5.07	**0.91**	976	2.29	3.37	**2.73**	868	2.90	4.40	**1.99**
EVO[41]	903	2.35	**0.71**	60.46	1016	1.87	**0.55**	86.95	846	4.35	**1.32**	72.17
本文 1	970	**2.28**	2.58	6.53	1059	1.90	1.74	11.78	845	2.47	1.62	39.94
本文 2	**902**	2.31	1.87	13.50	**937**	**1.04**	0.80	39.18	**840**	**2.18**	1.97	46.68

注. 本文 1 和本文 2 分别表示本文算法生成的六维各向异性网格和基于曲率的各向异性网格; V为生成网格顶点数; d_A为平均距离($\times10^{-3}$); d_A为 Hausdorff 距离($\times10^{-2}$); T表示运行时间. 粗体表示最优值.

本文算法成功地对所有参数曲面生成了 2 种度量下具有复形结构的三角网格, 且保证了网格的拓扑与参数曲面一致. 从表 5 可以看出, 基于曲率度量生成的网格在精度上普遍优于基于六维度量生成的网格, 这个结果与逼近论中的结论相吻合, 即符合曲率各向异性的三角网格给出光滑曲面在 l_2 距离下的最优逼近[42-43]. 与 ACVD 算法相比, 本文算法的单边 Hausdorff 距离和平均距离更小. 与为了降低最大误差而设计的启发式算法 EVO 相比, 尽管本文算法限制了交线上的网格点, 且不针对网格面片到参数曲面的距离进行额外优化, 基于曲率度量生成的网格精度依然可以达到与对比方法相当的水平, 且具有更低的平均误差.

从表 6 可以看出, ACVD 和 EVO 方法生成的网格在边界处和非正则点处均没有保持自然过渡的各向异性. ACVD 在边界处的三角网格比内部更狭长, 而 EVO 也会在曲率较小的边界处生成狭长三角形. 相比之下, 本文算法生成的各向异性网格在边界处仍然保持与内部一致的各向异性.

表 6　不同曲面的各向异性网格生成结果

曲面	生成结果	ACVD[37]	EVO[41]	本文 1 (六维各向异性网格)	本文 2 (基于曲率的各向异性网格)
Enneper	各向异性网格				
	曲面到网格的距离				
Riemann	各向异性网格				
	曲面到网格的距离				
Bi-quadric2	各向异性网格				
	曲面到网格的距离				

5　结　语

本文提出了一种针对有理参数曲面的网格生成算法，该算法可帮助 CAD 系统正确处理自交参数曲面. 本文基于动平面法进行算法设计，利用曲面的第一基本形式来弥补动平面法无法得到非一般奇异点的缺陷. 为了保证网格交线的拓扑正确性和邻域协调性，本文设计了交线的配对追踪算法，实验结果表明该算法还可提升网格交线的精度. 此外，本文在粒子能量中引入了边界能量项，以优化网格点在边界处的分布. 本文算法能保证网格曲面拓扑与原曲面一致，且在精度和效率方面都具有显著的优势. 在各种度量下，本文算法可以保证边界线以及自交线上网格点的分布与度量一致.

在下一步工作中，将针对本文算法的通用性和效率考虑进一步改进. 在通用性上，通过引入有理参数曲面之间的精准交线算法，考虑将本文算法用于生成 NURBS 曲面网格；在效率上，可通过并行方法加速基于粒子的网格生成法中粒子之间的相互作用力和粒子能量计算.

参考文献(References):

[1] Farin G, Hoschek J, Kim M S. Handbook of computer aided geometric design[M]. Amsterdam: Elsevier, 2002: 141-163

[2] Farin G. Curves and surfaces for CAGD: a practical guide[M]. 5th ed. Burlington: Morgan Kaufmann, 2002: 119-146

[3] Goldman R. Pyramid algorithms[M]. San Francisco: Morgan Kaufmann, 2003: 418-422

[4] Alliez P, Ucelli G, Gotsman C, *et al*. Recent advances in remeshing of surfaces[M] //Floriani L, Spagnuolo M. Shape Analysis and Structuring. Heidelberg: Springer, 2008: 53-82

[5] Khan D, Plopski A, Fujimoto Y, *et al*. Surface remeshing: a systematic literature review of methods and research directions[J]. IEEE Transactions on Visualization and Computer Graphics, 2022, 28(3): 1680-1713

[6] Eisenacher C, Meyer Q, Loop C. Real-time view-dependent rendering of parametric surfaces[C] //Proceedings of the Symposium on Interactive 3D Graphics and Games. New York: ACM Press, 2009: 137-143

[7] Shimada K, Gossard D C. Automatic triangular mesh generation of trimmed parametric surfaces for finite element analysis[J]. Computer Aided Geometric Design, 1998, 15(3): 199-222

[8] Cuillière J C. An adaptive method for the automatic triangulation of 3D parametric surfaces[J]. Computer-Aided Design,

1998, 30(2): 139-149

[9] Tristano J R, Owen S J, Canann S A. Advancing front surface mesh generation in parametric space using a Riemannian surface definition[C] //Proceedings of the 7th International Meshing Roundtable. Dearborn: Sandia National Laboratories, 1998: 429-445

[10] Du Q, Faber V, Gunzburger M. Centroidal Voronoi tessellations: applications and algorithms[J]. SIAM Review, 1999, 41(4): 637-676

[11] Zhong Z C, Guo X H, Wang W P, *et al*. Particle-based anisotropic surface meshing[J]. ACM Transactions on Graphics, 2013, 32(4): 1-14

[12] Jung W, Shin H, Choi B K. Self-intersection removal in triangular mesh offsetting[J]. Computer-Aided Design and Applications, 2004, 1(1-4): 477-484

[13] Yamakawa S, Shimada K. Removing self intersections of a triangular mesh by edge swapping, edge hammering, and face lifting[C] //Proceedings of the 18th International Meshing Roundtable. Heidelberg: Springer, 2009: 13-29

[14] Yu F, Cao J, Shan J L, *et al*. PASM: parallel aligned surface meshing[J]. International Journal for Numerical Methods in Engineering, 2021, 122(15): 3705-3732

[15] Volino P, Thalmann N M. Efficient self-collision detection on smoothly discretized surface animations using geometrical shape regularity[J]. Computer Graphics Forum, 1994, 13(3): 155-166

[16] Volino P, Thalmann N M. Collision and self-collision detection: efficient and robust solutions for highly deformable surfaces[C] //Proceedings of the Computer Animation and Simulation. Heidelberg: Springer, 1995: 55-65

[17] O'Rourke J, Goodman J E, Tóth C D. Handbook of discrete and computational geometry[M]. 3rd ed. Boca Raton: CRC Press, 2017: 1029-1056

[18] Busé L, Elkadi M, Galligo A. Intersection and self-intersection of surfaces by means of bezoutian matrices[J]. Computer Aided Geometric Design, 2008, 25(2): 53-68

[19] Ho C C, Cohen E. Surface self-intersection[M] //Lyche T, Schumaker L L. Mathematical Methods for Curves and Surfaces: Oslo 2000. Nashville: Vanderbilt University, 2001: 183-194

[20] Park Y, Hong Q Y, Kim M S, *et al*. Self-intersection computation for freeform surfaces based on a regional representation scheme for miter points[J]. Computer Aided Geometric Design, 2021, 86: Article No.101979

[21] Jia X H, Chen F L, Yao S S. Singularity computation for rational parametric surfaces using moving planes[J]. ACM Transactions on Graphics, 2022, 42(1): Article No.12

[22] Baker T J. Mesh generation: art or science?[J]. Progress in Aerospace Sciences, 2005, 41(1): 29-63

[23] Ho-Le K. Finite element mesh generation methods: a review and classification[J]. Computer-Aided Design, 1988, 20(1): 27-38

[24] Botsch M, Kobbelt L, Pauly M, *et al*. Polygon mesh processing[M]. Natick: A K Peters/CRC Press, 2010: 63-150

[25] Hoppe H, DeRose T, Duchamp T, *et al*. Mesh optimization[C] //Proceedings of the 20th Annual Conference on Computer Graphics and Interact. New York: ACM Press, 1993: 19-26

[26] Yan D M, Lévy B, Liu Y, *et al*. Isotropic remeshing with fast and exact computation of restricted Voronoi diagram[J]. Computer Graphics Forum, 2009, 28(5): 1445-1454

[27] Lévy B, Bonneel N. Variational anisotropic surface meshing with Voronoi parallel linear enumeration[C] //Proceedings of the 21st International Meshing Roundtable. Heidelberg: Springer, 2013: 349-366

[28] Su K H, Lei N, Chen W, *et al*. Curvature adaptive surface remeshing by sampling normal cycle[J]. Computer-Aided Design, 2019, 111: 1-12

[29] Lei N, Zheng X P, Si H, *et al*. Generalized regular quadrilateral mesh generation based on surface foliation[J]. Procedia Engineering, 2017, 203: 336-348

[30] Borouchaki H, George P L, Hecht F, *et al*. Delaunay mesh generation governed by metric specifications. part i. algorithms[J]. Finite Elements in Analysis and Design, 1997, 25(1-2): 61-83

[31] Klein R, Straber W. Large mesh generation from boundary models with parametric face representation[C] //Proceedings of the Third ACM Symposium on Solid Modeling and Applications. New York: ACM Press, 1995: 431-440

[32] Labelle F, Shewchuk J R. Anisotropic Voronoi diagrams and guaranteed-quality anisotropic mesh generation[C] //Proceedings of the Nineteenth Annual Symposium on Computational Geometry. New York: ACM Press, 2003: 191-200

[33] Laurent B. Implicit matrix representations of rational Bézier curves and surfaces[J]. Computer-Aided Design, 2014, 46: 14-24

[34] Shen L Y, Goldman R. Implicitizing rational tensor product surfaces using the resultant of three moving planes[J]. ACM Transactions on Graphics, 2017, 36(5): Article No.167

[35] Do Carmo M P. Differential geometry of curves and surfaces[M]. New York: Dover Publications, 2016: 51-68

[36] Falai C, Yuyu F, Kozak J. Tracing a planar algebraic curve[J]. Applied Mathematics-A Journal of Chinese Universities, 1997, 12(1): 15-24

[37] Valette S, Chassery J M, Prost R. Generic remeshing of 3D triangular meshes with metric-dependent discrete Voronoi diagrams[J]. IEEE Transactions on Visualization and Computer Graphics, 2008, 14(2): 369-381

[38] Du Q, Wang D S. Anisotropic centroidal Voronoi tessellations and their applications[J]. SIAM Journal on Scientific Computing, 2005, 26(3): 737-761

[39] Liu D C, Nocedal J. On the limited memory BFGS method for large scale optimization[J]. Mathematical Programming, 1989, 45(1): 503-528

[40] Valette S, Chassery J M. Approximated centroidal Voronoi diagrams for uniform polygonal mesh coarsening[J]. Computer Graphics Forum, 2004, 23(3): 381-389

[41] Zhang W X, Wang Q, Guo J P, *et al*. Constrained remeshing using evolutionary vertex optimization[J]. Computer Graphics Forum, 2022, 41(2): 237-247

[42] Simpson R B. Anisotropic mesh transformations and optimal error control[J]. Applied Numerical Mathematics, 1994, 14(1-3): 183-198

[43] Heckbert P S, Garland M. Optimal triangulation and quadric-based surface simplification[J]. Computational Geometry, 1999, 14(1-3): 49-65