



有限元网格生成方法综述

雷娜¹, 冯宇豪¹, 段君毅¹, 张萍¹, 郑晓朋^{1*}, 顾险峰²

(1. 大连理工大学 软件学院, 大连 116600; 2. 纽约州立大学石溪分校 计算机科学学院, 纽约 NY11794)

摘要 有限元网格是链接几何模型和数值仿真的关键纽带, 其自动化自适应生成仍是须重点研究的问题。文章综述了三角形网格、四面体网格、四边形网格和六面体网格的研究现状; 对非结构化网格生成方法进行了总结, 详细介绍了三角形网格和四面体网格生成的主流方法与工程技巧; 进一步对结构化网格生成的技术和发展进行了分析, 论述了四边形和六面体网格的各种生成方法的原理、优缺点以及研究趋势。

关键词 三角形面网格; 四面体网格; 四边形面网格; 六面体网格

中图分类号 TP31 **文献标志码** A **DOI** 10.12178/1001-0548.2024246

A Survey of Finite Element Mesh Generation Methods

LEI Na¹, FENG Yuhao¹, DUAN Junyi¹, ZHANG Ping¹, ZHENG Xiaopeng^{1*}, and GU Xianfeng²

(1. School of Software Technology, Dalian University of Technology, Dalian 116600, China;

2. Department of Computer Science, Stony Brook University, The State University of New York, New York NY11794, USA)

Abstract The finite element mesh is the key link between the geometry model and the numerical simulation, and its automatic adaptive generation is still a key problem to be studied. In this paper, the research of triangular surface meshes, tetrahedral meshes, quadrilateral surface meshes and hexahedral meshes is reviewed. This paper summarizes the methods of unstructured mesh generation, and introduces the mainstream methods and engineering techniques of triangular mesh and tetrahedral mesh generation in detail. Furthermore, the technology and development of structured mesh generation are analyzed, and the principles, advantages and disadvantages of various generation methods of quadrilateral and hexahedral meshes are discussed.

Key words triangular surface meshes; tetrahedral meshes; quadrilateral surface meshes; hexahedral meshes

工业软件在现代工程设计和生产中扮演着至关重要的角色, 是现代工程行业不可或缺的一部分, 对于提高生产效率、加快产品上市速度以及推动技术创新具有重要意义。然而, 根据中商产业研究院《2022 年中国工业软件行业市场前景及投资研究报告》, 管理经营类的工业软件 7 成实现国产化, 研发设计类工业软件国产化率较低, 仅为 5%, 尚有 95% 的国产替代空间, 凸显了我国自主研发工业软件的紧迫性和重要性。近年来, 中国工业软件行业呈现快速发展态势, 成为国家经济转型升级的重要推动力量之一。

在工程领域, 计算机辅助工程 (Computer Aided Engineering, CAE) 软件能够基于产品的几何模型进行仿真分析, 预测产品在不同工况下的力

学、热力学、流体动力学等性能, 评估其结构强度、稳定性、振动特性等, 帮助用户快速评估不同设计方案的性能差异, 找出潜在的设计缺陷和改进空间, 并进行参数优化和灵敏度分析, 以实现产品设计的最优化和性能提升。在 CAE 仿真分析中, 各种物理场的方程通常被离散化为有限元方程组, 然后通过数值方法进行求解。相应地, 计算区域也需要被离散化为有限元网格, 具体而言, 是将复杂的几何形状分解为简单的几何单元, 如三角形、四边形、四面体、六面体等, 每个单元内的物理属性可以通过方程进行计算和求解。

在实际应用中, 最常用的曲面网格是三角形网格, 这是因为三角形网格更容易从原始数据 (如 CAD 模型或点云) 生成且生成算法较为成熟。生

收稿日期: 2024-09-06; 修回日期: 2024-10-15

基金项目: 国家自然科学基金 (T2225012, 62272080)

作者简介: 雷娜, 博士, 教授, 主要从事计算共形几何、有限元网格生成、计算机图形学等方面的研究。

*通信作者 E-mail: zhengxp@dlut.edu.cn

成三角形网格往往是网格生成领域的第一步, 三角形网格既可以直接用于仿真分析求解, 也可以作为其他网格生成算法的输入。三角形网格生成算法往往追求所生成网格的质量、速度和鲁棒性, 同时期望能够尽可能地减少网格的数量。对于特征线保持、自适应密度分布以及不同尺寸网格之间的平滑过渡是三角形网格生成算法研究的重点。对于工业界中的许多复杂模型, 往往只有三角形网格生成方式才能全自动生成有效、贴合模型的离散网格。在实际应用中, 三角形网格常应用于仿真分析、游戏建模等领域, 对于复杂形状的表达与模拟, 有着其他网格不可替代的作用。

相较于三角形网格, 四边形网格单元具有自然张量积结构, 适合进行样条拟合, 这使得它的高阶曲面建模方面更具优势。在计算机辅助设计和计算机辅助制造领域中, 四边形网格常用于绘制样条和非均匀有理 B 样条。这种网格结构可以更好地刻画曲面表面的形状, 提供更精确细致的建模结果, 具有较高的计算精度。此外, 四边形网格相比于三角形网格能够更好地表达模型的局部几何特征, 如曲面上的主曲率方向等, 这也是为什么在许多应用, 特别是需要保留细节和需要贴合精确几何信息的领域, 如动画制作、电影制作等领域中, 四边形网格被广泛使用的原因。同时, 由于四边形网格能够在不同的网格尺寸之间更迅速地过渡, 因此特别适用于处理各向异性问题, 如应用于物理仿真、模拟边界层流动等, 对于高精度的仿真计算来讲, 采用结构化的四边形网格, 往往带来更快的收敛速度和更高的计算精度。

对于体网格问题, 四面体网格是最常用也是最容易的剖分方式。四面体网格生成灵活性强, 对于复杂几何形状模型也能良好地适应。四面体网格生成效率高, 可对不规则几何模型进行快速填充, 在地形模型和有限元分析中起到中流砥柱的作用。四面体网格能够自由控制网格密度, 在狭小特征附近可灵活提升密度, 在较为平坦的区域可以降低密度, 因此在提供良好的数值精度的同时, 能有效控制网格整体数量。四面体网格是工业领域中应用最广泛的网格, 对于复杂模型的高精度拟合的场合, 它是工业领域中首选的网格类型。

另一种重要的体网格是六面体网格。六面体网格化是将空间中的区域剖分, 每个胞腔都是六面体, 结构化程度更高, 并且在数值计算精度、收敛速度、抗畸变程度及存储空间等方面相较于四面体

网格具有显著优势, 被认为是最理想的有限元体网格。对于模拟大形变的物理现象, 六面体网格更加灵活。六面体网格通常具有更好的计算效率和精度, 四面体网格则相对较差。此外, 相比于四面体网格, 六面体网格的单元之间具有更多的接触面 (六面体为 6 个, 四面体为 4 个), 因此能得到更多的插值信息, 具有更快的收敛速度, 在数值模拟上具有明显的优势。数值模拟的另一个关键问题是计算的效率, 对于同等级的精度要求, 所需六面体网格的体单元数目远远小于所需的四面体单元个数, 这使得存储以及读取的效率更高, 计算速度也比使用四面体网格高出许多。因此, 六面体网格在网格领域中具有“holy grid”之称, 其自动生成问题受到学术界和工业界的广泛关注。

1 非结构化网格

1.1 三角形网格

三角形网格生成算法是比较成熟的网格生成算法, 近年来三角形网格生成算法的研究往往聚焦于网格算法的密度控制效果、平均网格质量提升、算法生成效率加速。在不同的应用场景中, 有时需对固定点集生成三角形网格, 有时仅给定边界, 需在内部插点生成网格, 有时还需要保持内部特征线段和特征点。按照生成三角形网格的方法进行分类, 可分为基于 Delaunay 的方法^[1]、基于二叉树的方法^[2]、基于推波的方法^[3]、基于粒子的方法^[4]、基于深度学习的方法^[5]和其他方法^[6], 如图 1 所示。每种方法都有其优势和局限性, 需要根据具体问题和应用场景进行选择 and 调整。

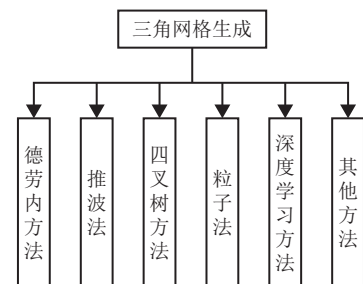


图 1 三角形网格生成方法分类

1.1.1 Delaunay 方法

Delaunay 方法的本质是保持 Delaunay 性质, 即每个三角形的外接圆中不包括任何其他点, 满足此性质的三角形网格是相同点集下最小角最大的三角形网格。关于固定点集生成 Delaunay 三角剖分的算法较多, 如文献 [7-9] 将二维点集提升到三

维然后来构造凸包，其连接关系给出了二维点集的 Delaunay 三角形网格的拓扑，如图 2 所示。Delaunay 三角形网格的对偶是 Voronoi 图，故可通过生成点集的 Voronoi 图来获取 Delaunay 三角剖分^[10-11]，如图 3 所示。

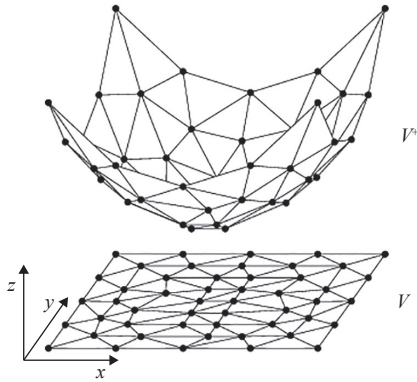


图 2 lifting 算法的凸包构造^[9]

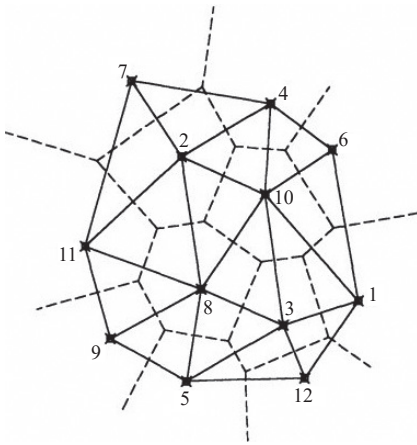


图 3 Delaunay 三角剖分与 Voronoi 图的关系^[10]

在实际应用中，常常只给定部分特征点和特征线，需要额外添加顶点生成高质量网格，研究者们为此提出了各种增点 Delaunay 算法。通常的增点 Delaunay 算法可分为以下 4 步：

- 1) 构造包含所有给定特征点集 P 的超三角形，依次插入点集 P 中顶点 V ，以 V 为新顶点细分所在三角形，并保持局部的 Delaunay 性质；
- 2) 通过边翻转等方式在 Delaunay 三角形网格中保持特征边；
- 3) 根据设置的约束条件和插点方式在特征边限定的内部区域不断增加顶点，直至所有三角形满足设定条件；
- 4) 移除外边界外部和内边界内部的三角形。

在增点算法生成三角形网格的算法中，算法将所有产生的坏三角形排序放入队列，坏三角形的判断条件和排序方式通常为角度或边长等，然后按序

向坏三角形中插入顶点。插入的顶点一般选择为坏三角形的重心^[12-14]、外心^[15-17]、最长边中点^[18-19]或 Voronoi 边中点^[20]，不同的插点方式和顺序直接决定了三角形网格的质量。

为了保持 Delaunay 性质，常用方法有 Lawson 算法（边翻转）和 Bowyer 算法（去除外接圆包含顶点的三角形），其中 Bowyer 算法略快于 Lawson 算法。为保持输入特征边，常用方法有对特征边相交线段进行翻转，直到无相交特征边^[21]，以及删除相交三角形，对特征边两侧分别生成 Delaunay 三角形网格。

文献 [18] 尝试在坏三角形的最长边中点进行插点，并引入 3 种确保收敛的约束条件，确保内部特征边和边界边附近构成小角度时收敛，该方法很容易保持特征边及控制边长，最终生成的三角形网格在实践中最小角通常可大于 27° 。

文献 [17] 对坏三角形的外心插点，对于靠近特征边的顶点，将其转为分割特征边，该方法在特征边不构成小夹角的时候，经证明必然收敛于 20.7° ，在实际算法中对大部分点线段集输入通常可收敛至 33.8° 。

文献 [22] 将多种插点方式结合，在坏三角形最短有向边一侧的一定角度的外接圆内，寻找离附近三角形最远的顶点，即在 Voronoi 顶点或 Voronoi 边与外接圆的交点进行插入。该方法相较于外心等方法，可用更少的插入顶点实现相同的最小角度，大部分情况下最小角可收敛至 40° 的同时，最大角小于 90° 。

通过尺寸场、密度函数、引入黎曼度量等方式，可以控制不同区域的坏三角形的条件，从而实现网格密度控制、生成各项异性网格等需求。

增点 Delaunay 方法通过与尺寸场结合，对不同区域的三角形设置不同的密度约束，可实现网络的密度控制^[23]，自适应地与尺寸密度结合进行插点，如图 4 所示，对不同区域设置不同的密度函数，从而自适应生成三角形网格。文献 [24] 的研究提出了根据物理仿真结果设置密度场，在解变化较大的区域生成密集的三角形网格，从而根据物理仿真结果生成自适应的三角形网格。

文献 [25] 通过修正方程控制网格密度，使用 Delaunay 方法作为点集的凸包三角形网格生成器，通过将边界外顶点向边界投影，生成二维约束边界下的满足修正方程的三角形网格剖分。文献 [26] 用水深作为密度函数的约束条件，结合 Delaunay

方法实现网格密度设置。文献 [27] 通过对样条先使用 Delaunay 方法生成粗糙三角形网格, 随后在粗糙三角形网格上设置密度函数, 再使用 Delaunay 方法在相同区域下重新生成质量较高的三角形网格。

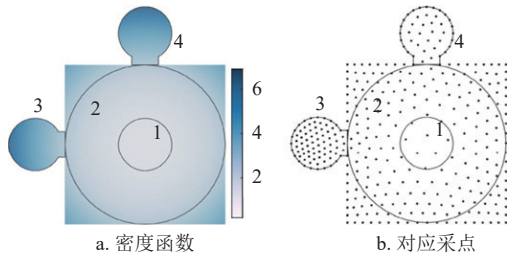


图 4 密度函数影响 Delaunay 方法采点^[23]

Delaunay 方法与黎曼度量相结合, 可在指定区域生成各向异性网格。文献 [28-29] 提出通过设置度量的方式也可生成各向异性网格, 从而与仿真的激波方向形成良好的对齐, 自适应地生成高质量网格, 如图 5 所示。

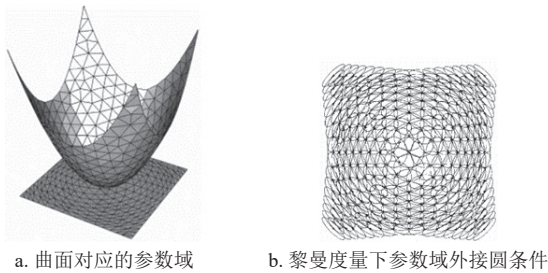


图 5 度量影响 Delaunay 参数域外接圆的判定条件^[28-29]

在样条曲面上使用 Delaunay 方法生成三角形网格, 可通过黎曼度量控制参数域局部三角形网格形状, 也可先生成粗糙贴体背景网格 M_1 , 再计算 M_1 的参数化生成畸变较小的参数域 D , 在 D 中重新生成最终三角形网格 M_2 。

文献 [30] 对于每个样片面片, 通过 Delaunay 方法生成的稀疏网格作为背景网格, 再将其作为重网格化的基础。文献 [31] 先生成整体的粗糙 Delaunay 网格, 随后进行分区参数化, 通过 Delaunay 方法重网格化, 生成均匀的三角形网格。

1.1.2 四叉树方法

在三角形网格生成领域, 输入往往是顶点集合、线段、简单多边形和平面直线图的组合。对于此类输入域 Ω , 四叉树方法通常是将域 Ω 封装到一个正方形边界框 $B(\Omega)$ 中, 以作为空间分解的根。对于每个根, 可分为 4 个相同大小的子域, 直到每个子域符合局部几何或者用户设定的最大细化准则。

如果相邻四叉树节点大小差异大, 将在局部诱

导低质量三角形网格, 为避免此类情况, 通常会相邻四叉树节点做出平衡性限制, 即四叉树中任意节点不应与小于其大小一半的节点相邻, 如图 6 所示。这样平衡的四叉树是生成高质量平面三角形网格的关键中间步骤。在诸多基于四叉树生成三角形网格的算法中, 主要区别在于四叉树的细分条件和三角形网格连接关系的构建方式。与曲面参数化方法相结合, 可将四叉树方法扩展到曲面三角形网格生成。

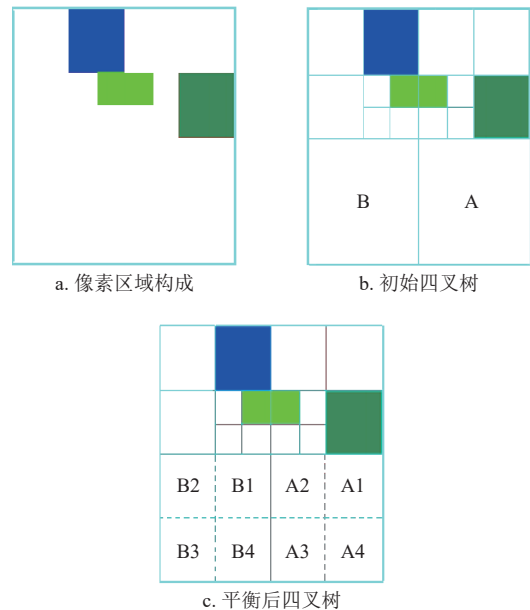


图 6 四叉树平衡过程

一类四叉树方法依赖模板来连接四叉树节点, 如文献 [2] 提出了一种对于输入图片进行平面三角剖分的算法, 根据像素点构建平衡四叉树, 并证明了对于该平衡四叉树的点集做 Delaunay 三角剖分, 最终只会存在 16 种三角形构型, 对应于 6 个模板, 其中三角形的 Aspect Ratio (长边与高的比值) 最大为 2.5, 最小角为 26.565° , 如图 7 所示。

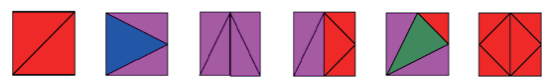


图 7 平衡四叉树顶点生成三角形的可能情况^[2]

另一类四叉树方法使用 Delaunay 方法构建三角形网格, 如文献 [32] 对样条模型进行三角化, 由样条模型给出边界集, 随后通过构建四叉树来区分外部节点、内部节点和边界节点, 用四叉树与曲面的逼近程度 ρ 和最大边长 l 作为划分四叉树的根据, 同时维持相邻节点的深度差不大于 1。最终, 对所有的内部节点、边界顶点与边界线段进行 Delaunay 三角剖分, 最后通过样条的映射函数 ϕ 映射回原始

样条曲面上, 从而实现基于四叉树的曲面三角形网格生成。与 Delaunay 方法相结合的四叉树方法可避免边界处的模板判断, 在边界处自然生成质量合理的三角形网格。

四叉树方法常用于地形网格的三角形网格生成, 如在山脉网格的生成中, 通过高度场可控制四叉树节点大小, 从而生成自适应高度变化的三角形网格, 文献 [33] 提出了一种使用希尔伯特曲线表示的四叉树用于山脉网格的生成, 相比指针表示的四叉树可大大减少内存消耗。

1.1.3 推波法

推波法是从待剖分区域的边界出发, 以边界作为前沿边 E , 不断向内寻找推波前进方向和最优插点位置 P , 使得点 P 与前沿边 E 构成高质量三角形; 随后更新前沿边集合, 当前前沿边集合成为空集时完成整个区域的三角形网格生成。各类推波法的差别主要在于前沿点的选取方式。推波法通常与 Delaunay 方法结合使用, 在最终所有因自相交而终止的前沿边构成的区域, 通过 Delaunay 方法构造内部三角形网格, 部分方法将推波法嵌入增点 Delaunay 的流程, 同步更新 Delaunay 网格和前沿边信息。推波法非常适用于由样条曲面直接生成三角形网格, 在这个过程中, 第一步是离散样条曲面的边界曲线, 该曲线的离散线段恰好可作为推波法的前沿输入, 如图 8 所示, 因此在大部分由样条生成三角形网格的商业软件中, 推波法都是不可或缺的方法。

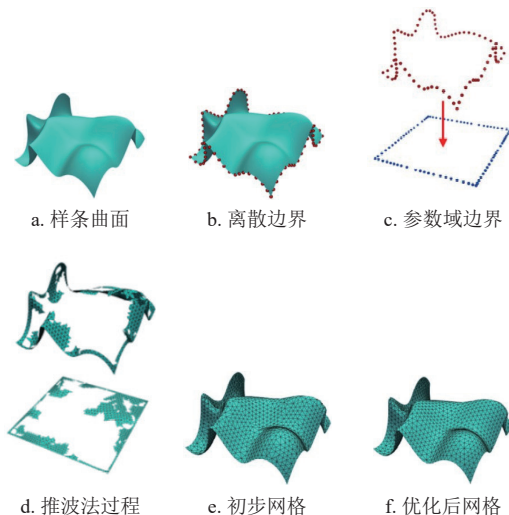
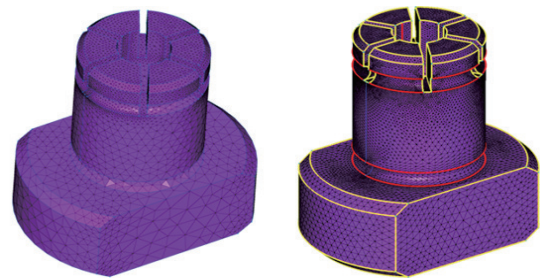


图 8 推波法从样条曲面生成三角形网格流程^[34]

文献 [35] 利用切向投影的方式在复杂样条曲面上生成各项同性三角形网格。在样条边界构建初始网格边界, 沿空间的切方向寻找构成正三角形的最佳顶点位置投影回样条曲面, 从而找到样条参数

域的最佳顶点位置。在此过程中根据局部曲率决定三角形大小, 从而自适应地根据曲率控制网格密度, 最终通过平滑化操作来提升网格质量。该类方法即便在映射畸变较大的曲面上, 也能生成合理分布的三角形网格。

文献 [3] 在 Pointwise 推波过程中通过限制特征边沿垂直方向的曲率来控制三角形网格过渡, 当前沿方向沿垂直方向的曲率大于用户指定角度, 则将特征边沿曲面法向挤压, 相较于 Delaunay 方法, 该方法在红色区域可生成高度一致且平缓过渡的三角形网格, 如图 9 所示。



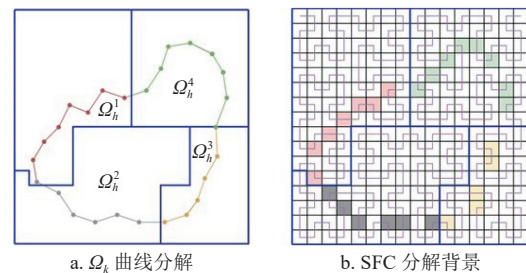
a. Delaunay 方法

b. 10° 转角推波

图 9 转角约束推波生成更贴体的网格^[3]

文献 [36] 根据黎曼度量决定推波的参数域采点位置, 从而在带参数域的曲面上生成密度与质量优良的三角形网格。

文献 [37] 对推波法的并行算法进行了优化, 首先将输入边界线段分区, 进而在不同区域并行推波, 最终生成全局一致的三角形网格, 如图 10 所示。



a. Ω_k 曲线分解

b. SFC 分解背景

图 10 空间曲线分解推波法区域^[37]

文献 [38] 给出了另一种并行的方式, 对不相邻的前沿边同步进行推波, 并检测相交情况, 从而实现算法的并行化加速。

文献 [34] 从样条边界出发, 设计了将 3D 点快速正交投影回样条曲面的投影算子, 并将正交投影操作与推波法结合, 在参数域寻找到的顶点与理想顶点偏差较大的情况下, 自适应地在空间中寻找理想顶点, 再投影到样条曲面, 如图 11 所示。结合 Delaunay 方法构造网格、AD 树加速定位、多面片

并行加速的方法, 实现了对复杂 CAD 模型的快速三角形网格生成。

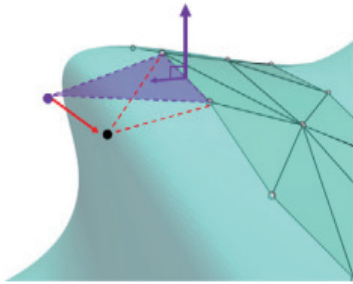


图 11 推波法在样条模型中使用投影确定点的理想位置^[34]

1.1.4 粒子法

粒子法的思想近似于对输入点集构建大小均匀的 Voronoi 图, 通常是把边界内部的顶点视为粒子, 通过粒子间的相互作用力等方式, 来最大化点集中所有顶点间的间距, 最终得到该点集下的三角剖分。此类方法的拓扑关系依赖于 Delaunay 方法, 不同算法的区别在于点之间排斥的表达方式 (如斥力、相切球) 和边界处顶点的处理方式 (预先固定边界上顶点, 或者内部点逐步靠近并被投影至边界上)。

文献 [4] 通过向容器内部不断注射流体粒子的方式填充点, 点之间的作用力符合牛顿定律, 通过设置核函数和流体颗粒的速度, 注入任意数量的顶点, 在边界区域内优化顶点间距, 靠近边界时可投影至边界曲线, 最后使用 Delaunay 方法构建三角形网格, 其算法流程如图 12 所示。文献 [39] 用 MATLAB 实现了一个开源的粒子三角形网格生成器, 对于给定区域内随机设定点集, 先通过 Delaunay 方法构造初始三角形网格, 利用网格的连接关系来传播粒子间的作用力, 将顶点都优化到理想的位置。

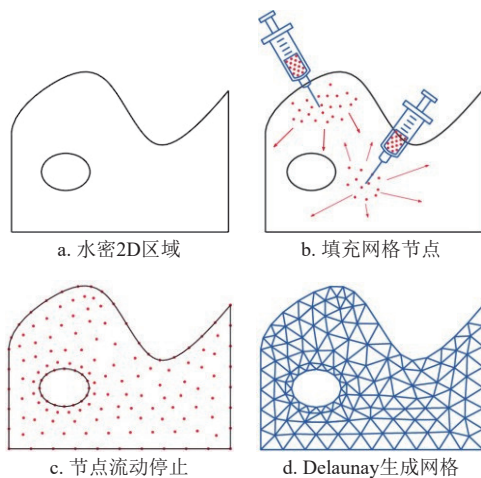


图 12 流体粒子方法生成三角形网格的流程^[4]

将粒子法应用于样条曲面网格生成时, 通常预

先固定边界顶点, 如文献 [40] 给出了一种在样条曲面的参数域上生成三角形网格的粒子方法: 在每个样条曲面边界均匀采点, 内部随机生成顶点, 随后将每个顶点视为一个均匀的气泡, 使得不同气泡尽量相切, 最终使用 Delaunay 方法进行三角形网格生成, 构建均匀相切气泡的过程, 等价于生成一个均匀的 Voronoi 图, 其对偶图等价于一个各向同性的 Delaunay 三角剖分。

此外, 文献 [41] 探索了在多个面片的样条模型上, 通过平衡气泡、构建 3D Voronoi 图, 恢复顶点回到样条曲面表面上, 最后再通过对偶 Voronoi 图得到跨越不同样条面片边界的曲面各向同性三角形网格, 如图 13 所示。

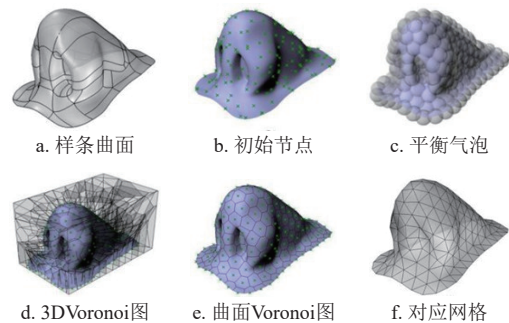


图 13 通过气泡构建跨样条面的三角形网格^[41]

通过控制力的大小, 或控制不同粒子的相切球大小, 可控制粒子法生成的网格密度。文献 [42] 对球填充算法进行改进, 对三角形网格表面构造不同大小的相切球, 得到变密度的三角形网格。文献 [43-44] 将一个均匀的三角形网格覆盖在样条曲面参数域上, 通过设置不同的局部虚拟力影响参数域中不同地方的网格密度, 随后根据样条曲线的边界裁剪整个平面三角形网格, 将裁剪后的三角形网格映射到样条曲面得到曲面三角形网格。文献 [45] 提出了一种 2D 区域改变密度的快速粒子法网格生成算法, 通过 Delaunay 方法生成较密的初始网格, 对每个三角形网格顶点设置一个较大的气泡, 固定边界气泡, 设计了三级邻接气泡来快速查找邻接关系, 能够以较高的效率计算不同斥力, 从而快速优化粒子位置与气泡大小, 生成高质量的 2D 三角形网格。

1.1.5 基于深度学习的方法

随着几何深度学习的发展, 深度学习已经被广泛地运用到了三角形网格生成任务中^[46]。与提到的传统方法相比, 深度学习方法扩展了数据的输入类型, 为三角形网格生成任务提供了新的角度和解决

方案。总的来说,深度学习下的网格生成问题被看作给定具有形状约束下的条件生成问题,可用如下公式^[47]表示:

$$p(M, S | C) = p(M | S, C)p(S | C) \quad (1)$$

式中, M 表示网格; S 表示形状; C 表示蕴含形状信息的条件。

具体来说,深度学习算法除了可以在点云^[48-59], 体素^[5, 60-61]等显式形状表示的数据生成形状约束下的三角形网格外,在符号距离场(SDF)^[59, 61-62]、占有(Occupancy)^[61, 63]和神经辐射场(NeRF)^[64-65]等隐式形状表示的数据生成形状约束下的三角形网格。除此之外,得益于深度学习强大的特征提取和拟合能力,一些算法可以图像视频^[66-71]、笔画轮廓^[72-73],甚至是文本信息^[74-75]等丢失了或遮挡了部分形状信息的数据为条件生成对应的三角形网格模型。

在显式形状约束下的三角形网格生成中,点云为主要代表。文献[50]提出 Point2mesh 利用点云作为形状监督条件,网络提供自先验,将模板网格变形到对应的形状上,如图 14 所示,显示了 Point2mesh 的主要流程。

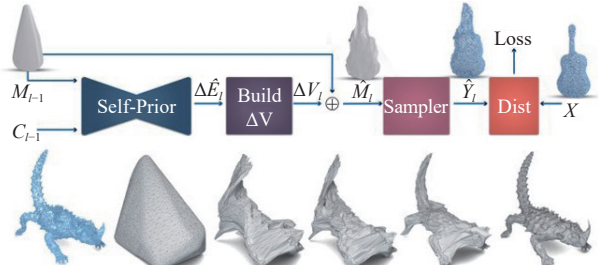


图 14 Point2mesh^[50] 的流程图

在隐式形状约束下的三角形网格生成中,往往将目标曲面看作隐函数的零水平集,从而利用神经网络拟合隐函数,并结合 Marching Cubes 或 Voronoi 图等算法构建对应曲面的三角形网格。文献[46-47, 61]将 Marching Cubes 等算法进行可微化处理,从而实现端到端的训练。值得注意的是,相比于点云体素等离散化表示,隐式表示往往是连续的,使得网络模型更易拟合,所以文献[76-81]将点云和体素转换为 SDF 或 Occupancy 进行表面三角形网格的生成。如图 15 所示,文献[62]的研究利用神经网络将 SDF 转换为 PoNQ (Points, Normals, QEM) 表示,并构建三角形网格。

而对于丢失了部分形状信息的数据,相比于传统方法,深度学习具有一定的优势。深度学习算法

利用网络从缺失大部分形状和结构信息的数据中提取深度特征,从而刻画出相应的形状信息,指导三角形网格的生成。文献[69]从单张图像中提取深度特征,并将特征映射到三维欧式空间的曲面网格变形算法中,从而实现目标形状的三角形网格生成。Pixel2mesh 的流程图如图 16 所示。

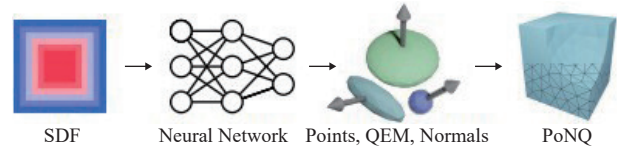


图 15 PoNQ^[62] 的算法流程

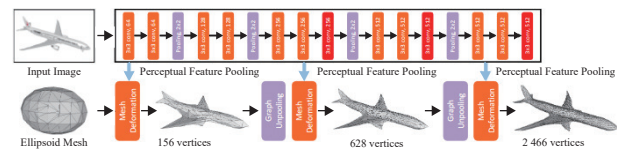


图 16 Pixel2mesh^[69] 流程图

除了上述结合几何深度学习构建三角形网格生成算法外,也有研究者将自然语言处理中的生成架构运用三角形网格生成中,从而为多边形生成任务提供了新的角度和方法。其中最具代表的是采用 Trans-former^[82]架构的 MeshGPT^[83]和 MeshAnything^[47],他们将三角形网格通过排序看作三角形单元序列,利用序列化生成方法,在给定形状条件下生成对应的三角形网格。MeshAnything 针对不同的数据输入生成对应形状的三角形网格结果,如图 17 所示。

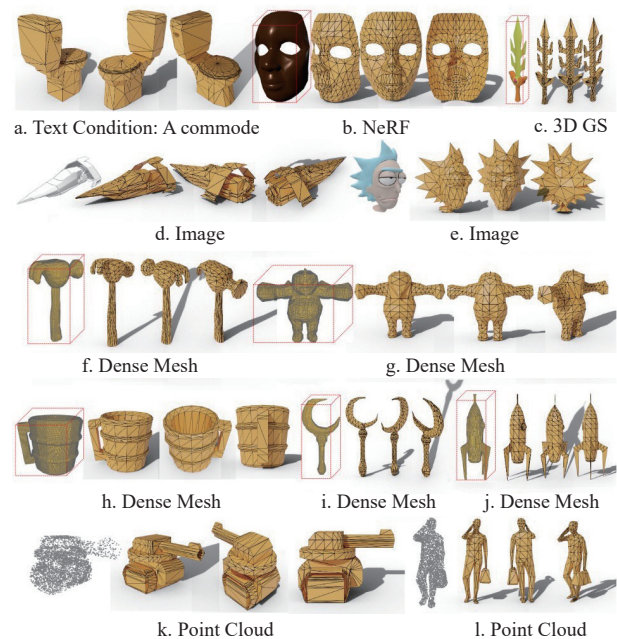


图 17 MeshAnything^[47] 对不同输入生成的三角形网格

1.1.6 其他方法

除了以上常见的三角形网格生成方法, 还有一些特定需求下, 较为少见的非结构化三角形网格生成方法, 如多边形生成与分割、变形等方法。

文献 [6] 构建了一个从文本到三角形网格的生成器, 文本指定了微结构的几何结构, 文中将文本描述的微结构转化为近似的圆, 再转化为多边形网格, 最后将多边形网格分解成三角形网格, 如图 18 所示。

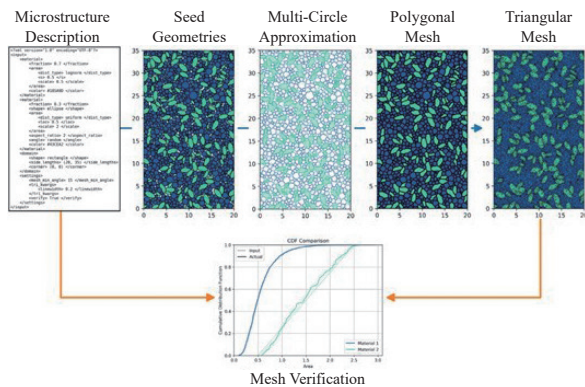


图 18 微结构网格生成流程^[6]

文献 [84] 对于给定的样条曲面, 首先构造一个与给定曲面同伦的粗糙的三角形网格, 随后指定各三角形公共边界上的细分点数, 并行地对每个大三角形进行细分、将顶点投影回样条曲面, 从而实现快速生成, 不同细化程度的多分辨率网格有利于对模型进行区域分解, 如图 19 所示。

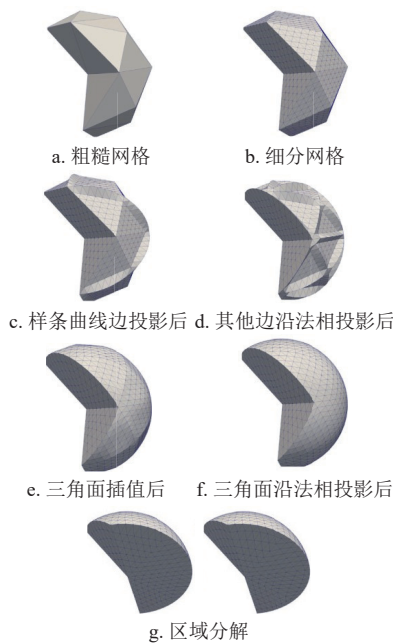


图 19 并行网格生成算法流程^[84]

部分方法对生成的均匀三角形网格给定约束,

进行变形, 生成适应特殊需求的三角形网格, 如文献 [85] 使用 line-SIAC 过滤器将三角形网格转换为适应二维标量场的各向异性网格, 结果适用于流体流动的模拟可视化。文献 [86] 使用均匀三角形网格作为基础, 以 Stokes 方程为状态约束, 在受流体约束的情况下, 以最大功耗为优化目标用梯度法求解优化问题, 从而将均匀的三角形网格优化为沿流动区域密度合理过渡的三角形网格。

1.2 四面体网格

四面体网格生成从算法上大多数是对三角形面网格生成算法的三维推广。按照生成四面体的方法进行分类, 可分为基于 Delaunay 的方法^[3]、基于八叉树的方法^[87]、基于推波的方法^[88]、基于粒子的方法^[4]、基于深度学习的方法^[89] 和其他方法^[90]。根据文献 [91] 的调查研究, 四面体网格生成中, 只有基于 Delaunay 的方法能处理本质上任意复杂的对象, 故四面体网格生成方法中 Delaunay 方法占主导地位, 其他方法也通常依赖 Delaunay 方法来构建拓扑, 通过为 Delaunay 算法提供更好的顶点位置的选取来生成高质量的四面体网格。

1.2.1 基于 Delaunay 的方法

将 Delaunay 性质向三维推广, 保证生成的四面体单元的每个四面体单元的外接球内不包含其他任意顶点, 可用于点集生成四面体网格。Delaunay 方法的输入通常为三维空间的点、线段、复形 (Piecewise Linear Complexes, PLC), 大部分算法仅支持非自相交 PLC, 并将在多边形面上引入额外的顶点以保持形状, 这种顶点称为 Steiner 点。部分方法尽可能去保持表面三角形网格^[92], 但保持任意三角形表面生成四面体且不引入 Steiner 点仍然是一个困难的问题。增点算法的思想可推广至三维, 在不断保持 Delaunay 性质的同时逐点插入生成四面体网格, 在保持 Delaunay 性质时, 通常使用 Bowyer 算法的三维扩展, 将所有外接球包含新加入顶点的四面体删除, 并构造空腔与新顶点的拓扑。增点算法的流程通常如下:

- 1) 对 PLC 的顶点生成四面体网格凸包;
- 2) 通过引入 Steiner 点或其他方式保持 PLC 的面区域约束;
- 3) 考虑四面体网格质量, 使用不同插点方式对四面体网格单元进行细化, 允许修改 PLC 表面的算法, 在此过程中也将允许新顶点插入 PLC 表面上;
- 4) 移除 PLC 外部的四面体单元。

部分方法接受生成四面体网格表面与输入的表

面三角形网格不一致, 允许引入额外表面顶点甚至修改表面拓扑以提高算法的鲁棒性和生成四面体网格的质量, 文献 [93] 从样条边界生成 PLC, 在样条曲面上的不同区域采样不同密度的点集作为输入, 生成密度缓慢过渡的 Delaunay 四面体网格。

文献 [94] 证明了当所有输入边都是 Delaunay 时, 保证不添加 Steiner 点的约束四面体网格存在, 并基于此提出了一系列鲁棒高效的 CDT 算法^[92, 95-96]。文献 [96] 将外心插入的 Delaunay 网格生成算法推广至三维, 通过对空间排序进行快速有效地顶点定位, 通过边翻转、边移除、顶点插入的方式来保持输入三角形的约束边界, 相较于之前的方法, 可维持绝大部分输入曲面三角形的边界和边, 如图 20 所示。此外, 文中还设计了一套几何算子, 可自适应地使用不同精度的算法快速计算四面体外接球是否包含某点及 4 个点的有向体积, 从而有效地提升了算法效率和鲁棒性。在内部顶点插入后, 通过局部拓扑操作和平滑的后处理, 最终实现了可鲁棒高效地生成保边界的高质量四面体网格软件 Tetgen。Tetgen 被广泛用作复杂任务的四面体网格生成器, 如文献 [97] 将 3D 医学图像基于体素生成粗糙表面三角形网格, 重采样后利用 Tetgen 生成四面体网格, 从而实现了医学图像到四面体网格的生成。

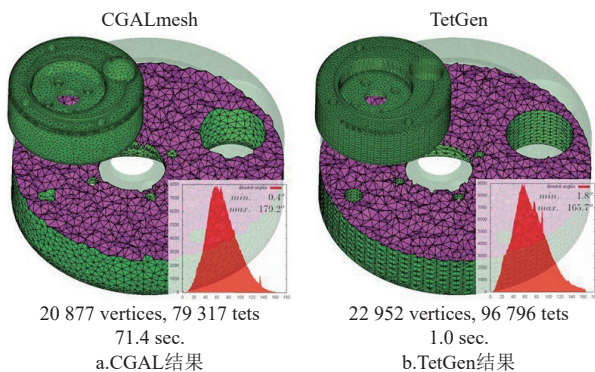


图 20 TetGen 相比 CGAL 维持了大部分边界面和边^[96]

文献 [13] 对 5 种穿透 PLC 的四面体情形设计了不同解决方案, 并通过算子尽可能减少保持 PLC 过程中引入的 Steiner 点。

在四面体 Delaunay 算法细化的过程中, 将算法与尺寸场结合, 可实现对四面体网格密度的有效控制, 如 Pointwise^[3] 中的四面体生成过程中, 用户指定尺寸场可用于控制特定区域的四面体网格密度。文献 [24] 将 Delaunay 算法与二阶拉格朗日-欧拉流动求解器相耦合, 自适应地控制生成沿流动方

向分布的内部四面体网格, 从而更适用于可压缩流动与激波仿真方面的应用。

PLC 的质量将极大影响最终生成四面体网格的质量, 优化四面体网格质量可对算法输入进行优化, 如文献 [30] 通过优化表面三角形网格, 生成自适应重网格化的三角形网格, 以用于 Delaunay 算法生成更高质量的四面体网格, 如图 21 所示。

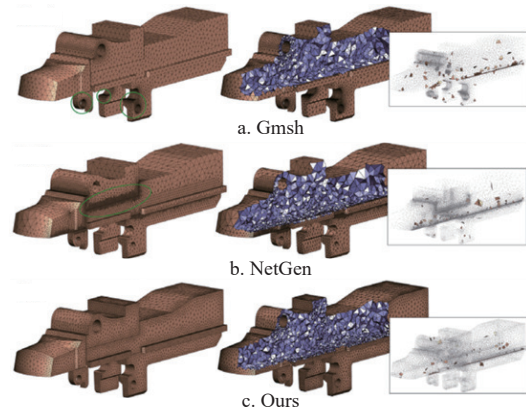


图 21 不同表面网格对生成四面体质量的影响^[30]

为加速四面体网格生成的效率, 近年来大量研究者通过不同分区的方式进行并行化算法加速。文献 [98] 将飞机模型沿轴向进行区域划分, 随后对不同区域并行地进行 Delaunay 三角形网格生成。文献 [99] 对装配体模型的不同面之间首先生成一致的曲面网格, 随后递归地分解不同的区域, 直至每个区域的曲面域小于给定阈值后, 递归分解结束, 随后并行使用 Delaunay 方法生成四面体网格。

文献 [84] 通过 Schur 域的分解方法, 将初始样条模型分解为多个类六面体单元, 随后在每个单元内部并行生成四面体网格。文献 [100] 通过对区域进行编号, 不同区域间先用约束 Delaunay 方法生成壁网格, 随后使用并行方法生成四面体网格, 以加速网格生成。

1.2.2 基于八叉树的方法

八叉树方法是一种简便实用的四面体网格生成方法, 是在待剖分区域构建八叉树, 对于处于目标内部的每个立方体, 从中心分割为一致的四面体。而对于与边界相交的立方体, 与边界相截成为多面体^[87], 通常会用模板法划分多面体为四面体, 并通过优化八叉树节点的方式提升四面体网格质量, 八叉树方法对于内部四面体网格的质量有数值保证。

对于八叉树与边界相交的区域, 文献 [101] 使用插入质心的方式分割多面体单元, 在边界处则需要处理表面与立方体单元相交的各种情况, 如图 22

所示, 该方法最终通过对多边形与质心进行连接, 进而生成全局一致的四面体单元。类似地, 文献 [102-103] 提出了一种仿造晶体结构的四面体网格生成方法, 对于目标区域使用与质心相连的晶体状立方体生成初始网格, 随后通过变形和光滑优化的方式逼近曲面, 生成自适应密度的四面体网格。

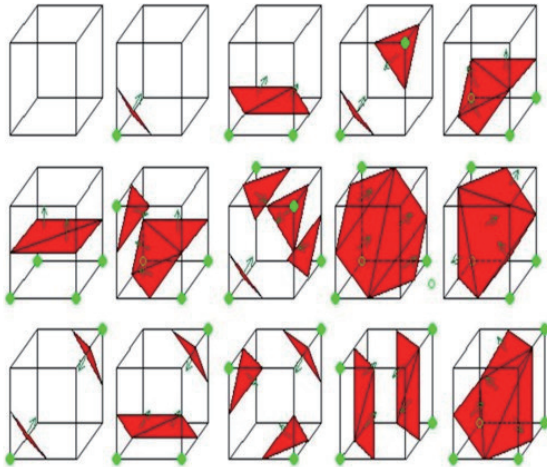


图 22 边界面与立方体单元的相交情况^[101]

类似于二叉树方法, 大部分八叉树方法会构造平衡八叉树, 使相邻八叉树节点大小差距不超过一倍, 以利于使用模板, 控制生成的四面体网格单元质量。文献 [104] 在构造了平衡八叉树后, 将八叉树顶点分为 3 种类型: 内部节点 (图中+号表示)、外部节点 (图中-号表示) 和边界节点, 如果八叉树单元所有节点具有相同的符号类型, 则最小化四面体二面角的分割点在单元中心, 否则选择单元中心到边界曲线的投影, 如图 23 所示。此外, 相较于传统方法, 仅允许一个最小点存在, 改进的算法将在单元格中心的 1/4 半径内寻找额外的最优顶点, 从而避免了最小点靠近边缘时产生的小角度问题, 如图 24 所示。经证明该算法对任意光滑闭曲面网格生成的四面体网格二面角范围为 $12.04^{\circ} \sim 129.25^{\circ}$ 。

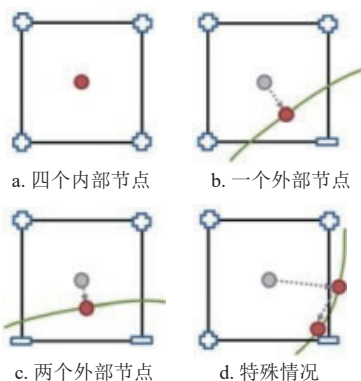


图 23 最小化计算, 红色点为重定位的结果^[104]

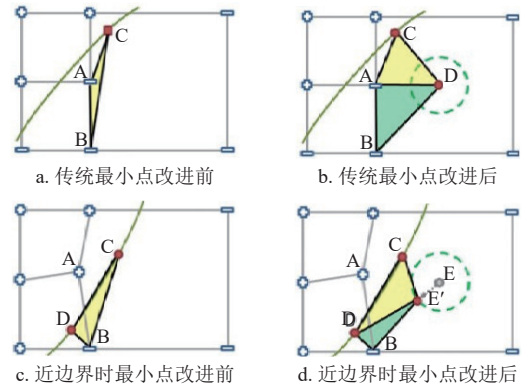


图 24 调整最小点位置以提升最小角度^[104]

1.2.3 基于推波的方法

基于推波的四面体生成方法是三角形网格生成推波法的三维扩展, 主要是在前沿点附近寻找其他顶点构造连接关系。但是目前大部分算法仅将推波的思想用于寻找最优点 P , 随后使用 Delaunay 方法将该 P 点加入四面体网格中, 同时更新推波前沿的锋面, 当锋面碰撞时将锋面移除队列, 最终当锋面为空时收敛。推波法相较于纯 Delaunay 方法, 在局部提供了相对更理想可控的顶点位置, 从而生成更高质量的四面体网格。

文献 [105] 使用传统的推波算法思想, 不断对最小前沿面寻找局部最优位置并探索半径 R 范围内的所有其他顶点, 进行几何检查后决定如何连接局部四面体, 并更新前沿面, 当前沿面为空时, 算法结束。该算法的绝大部分时间耗费在几何检查中, 并且有时会产生裂纹状狭小四面体, 故文中通过研究局部顶点选择、生成体元素的有效性、病态形状面的移除等方式来提升生成四面体网格质量。

文献 [106] 将推波法与 Delaunay 方法结合起来, 从而避免了推波法繁复的计算碰撞、分割多边形等耗时的操作, 仅需控制推波生成点在待划分区域内部即可, 这极大地减少了计算时间。

文献 [88] 先对分割的区域中间层生成四面体网格, 随后并行地在不同区域进行四面体网格生成、对每一个子区域使用推波法寻找理想点, 使用 Delaunay 方法将其插入每个子区域, 从而快速生成大型模型的高质量四面体网格, 其中间层的示意图如图 25 所示。

文献 [37] 通过对空间曲线的区域分解算法来进行多曲线的并行推波, 将分布式自相交八叉树森林算法用于同一区域的自相交判断加速, 使用并行一致的最大独立集来保持最终生成网格的一致性, 该算法最多可有效扩展到 2 000 个处理器并行

执行，在凹形区域和非均匀场景都表现出优异的性能。

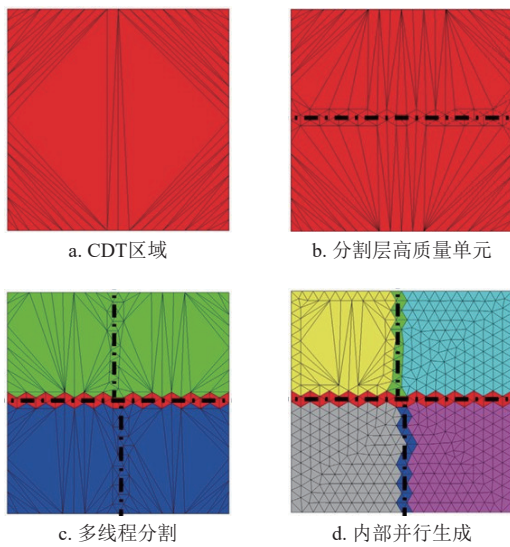


图 25 2D 区域分解流程^[88]

1.2.4 基于粒子的方法

基于粒子的方法在四面体网格生成流程相较于三角形面网格的流程类似，对于输入边界区域包裹的顶点，通过求解力学方程、粒子间互斥等力学方式，以此构造空间中的保持合理间距的点集 P ，随后使用 Delaunay 方法生成给定点集的四面体网格。

为确保生成网格表面与曲面一致，基于粒子的算法通常固定边界区域，或者将靠近边界的粒子投影回边界曲面。文献 [4] 通过在曲面空间内部填入点集，利用顶点间的互斥来保持顶点之间的间距，同时通过投影来保持边界面上的顶点，最终通过 Delaunay 方法对给定点集在曲面内部生成四面体网格，但该方法有时会创建质量较差的四面体，随后通过顶点的轻微挪动和去除坏四面体的后处理来提升四面体网格的生成质量。文献 [39] 在 MATLAB 中通过 Delaunay 方法构造初始四面体网格，同时利用网格拓扑作为力的作用条件，调整优化网格质量，在优化过程中，边界点始终保持不变。

文献 [107] 定义了 4 种不同的粒子：奇异粒子、特征曲线粒子、特征表面粒子、正常粒子。通过对不同粒子的定义，区分不同区域，使算法能够对相邻多个不同区域的粒子并行地进行位置优化，最终在不同分区快速生成一致的四面体网格，如图 26 所示。



图 26 多区域并行的粒子优化^[107]

1.2.5 基于深度学习的方法

基于深度学习的四面体网格生成相对于三角形面网格生成更具挑战性，智能四面体网格生成算法发展较晚，但由于体网格生成具有根本的重要性，因此也逐渐成为了研究热点。

2020 年，文献 [108] 首次将深度学习算法运用到了四面体网格生成问题中。算法主要分为两个阶段：

1) 将空间进行体素划分，结合图像点云等形状约束，预测体素是否被占有（网格元素的几何位置是否在该体素内），并提取初始四面体网；

2) 通过已提取的形状约束，采用变形算法对初始网格的顶点位置进行调优，从而生成具有良好形状拟合的最终四面体网格。算法流程如图 27 所示。

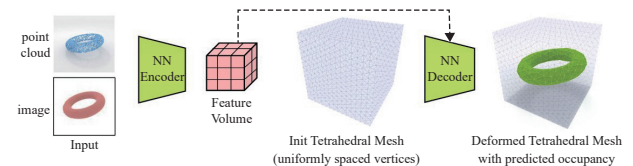


图 27 从点云和图像中利用变形生成四面体网格^[108]

此后，文献 [109] 提出 TetGan 算法，在占有预测和变形的基础上，通过结合四面体细分和简化操作设计四面体上的卷积和上下采样算子，从而提取多尺度特征信息，以达到更优的四面体生成效果。图 28 展示了与细分和简化对应的上下采样算子。文献 [110] 在两阶段之前结合扩散模型，构建噪声到体素数据之间的映射，实现从噪声数据到四面体网格数据的生成。

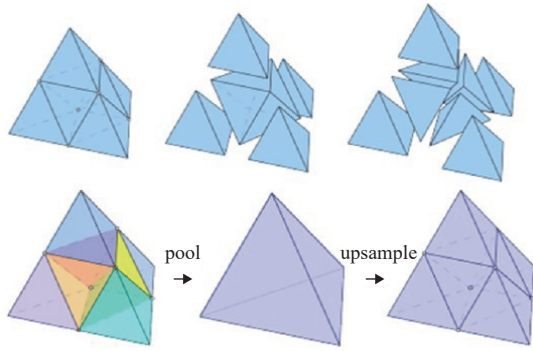


图 28 利用细分做池化和上采样操作^[109]

除了上述由形状约束的顶点位置调优外, 文献 [89] 在给定节点和单元数量下, 利用神经网络去寻找目标偏微分方程系统约束下的误差尽可能小的解, 从而实现非均匀网络的生成。

此外, 文献 [111] 针对于四面体模型设计的神经网络卷积算子也一定程度上推动了基于深度学习的四面体网格生成算法的发展。

1.2.6 其他方法

除了以上方法, 还有部分研究者通过拓扑操作、变形等方式对粗糙四面体网格进行修改, 生成高质量的非结构化网格。如文献 [90] 利用包括边分割、边塌缩和边交换的拓扑操作修改需要生成各项异性网格的区域, 最终生成结果如图 29 所示。类似地, 文献 [112] 使用度量控制和拓扑操作的方式生成了理想的各向异性网格, 可以用于处理薄壁和复杂几何形状的细化问题。

文献 [113] 通过对区域进行分解, 并行地对每个区域使用过滤器并行地进行优化, 减少微小单元, 从而快速提升四面体网格质量。

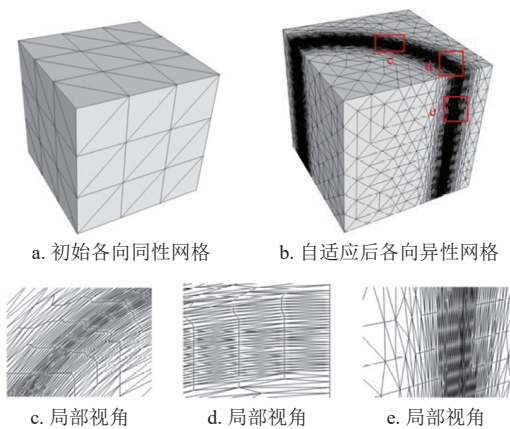


图 29 利用边分割生成指定区域的各项异性网格^[90]

文献 [114] 使用长方体对目标网格进行形状逼近, 将长方体细分成六面体单元, 再细分为四面体

单元, 随后通过自适应方式细分部分四面体, 将表面投影回曲面, 内部节点通过后序平滑优化生成逼近原始形状的高质量四面体网格, 如图 30 所示。

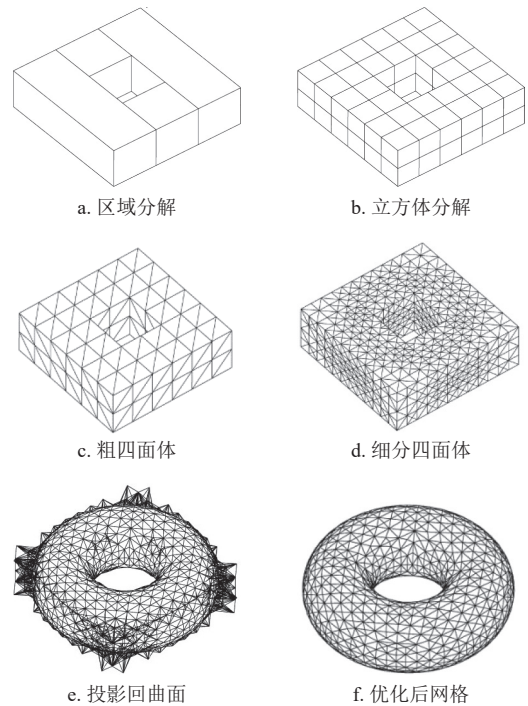


图 30 四面体网格生成流程^[114]

2 结构化网格

2.1 四边形网格

近年来, 自动化的非结构化网格算法层出不穷, 而自动化的高质量四边形及六面体网格生成算法还不够成熟。随着工业界对产品分析仿真的要求日益提高, 这一领域逐渐成为网格生成领域的研究热点, 研究者们提出了许多创新算法和技术, 并取得了一系列令人瞩目的成果。这些方法包括基于度量的方法^[115]、铺砌法^[116]、映射法^[117]、Q-Morph 方法^[118-119]、基于 Morse 理论的方法^[120]、结合交叉场的方法^[121]、代数几何方法^[122-123]。每种方法都有其优势和局限性, 需要根据具体问题和应用场景进行选择 and 调整。

2.1.1 三角形网格转化为四边形网格

目前, 四边形网格生成方法多以三角形网格作为输入, 由三角形网格生成四边形网格的许多方法非常直观且简单, 总的来说分为两类: 分裂方法与合并方法。

分裂方法如 Catmull-Clark 细分法^[124], 以三角形网格为例, 该方法通过在每个面上添加一个新点 O , 同时在每条边上添加一个新点 P_1, P_2, P_3 ,

然后分别连接点 O 与边上的新点 P_1, P_2, P_3 。如此，每个三角形将被分裂为 3 个四边形，如图 31a 所示。显然这种方法对多边形网格也同样适用，可以有效地将各种多边形网格转换为四边形网格，但需要保证多边形网格上的面都是凸多边形，否则需要应用割耳法^[125]将所有网格上的面处理为多个凸多边形的合集。此外，这种分裂方法为网格引入大量的不规则节点（原三角形网格的顶点），生成四边形网格的质量相对较差，需要进行进一步的几何优化和拓扑优化。

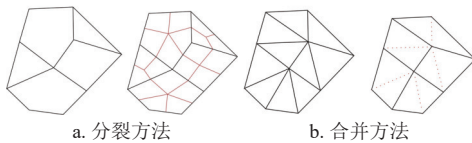


图 31 直接转换三角网格为四边形网格

另一种直观的方法是将网格中两个相邻的三角形按照一定的规则合并成一个四边形，如图 31b 所示，从而形成四边形网格^[126-128]。合并方法的关键步骤之一是选择合适的合并顺序和方向，这决定了生成网格的质量，也关系着三角形是否能全部合并。文献^[129]提出了一种通过选择性地去除三角形之间的对角线来生成三角形单元网格上的四边形单元的算法。四边形单元的质量（形状）可以通过无量纲的 α 值（ $\alpha = \text{面积}/\text{边的平方和}$ ）来测量。根据所生成四边形网格的质量来自动判定、控制三角形面片的合并顺序，可以得到只含有少量三角形的网格。合并方法简单直观，但正如前文提到的，这种两两合并的过程难以确保每个三角形单元都能合并为四边形单元，可能得到四边形和三角形的混合网格，这要求待处理的封闭曲面网格中的三角形数量必须是偶数。不过，如果是带边界的曲面则简单许多，只需将边界处剩余的额外一个三角形分裂为两个三角形即可继续合并。文献^[126-127]采用了贪婪算法，优先识别出最合理的三角形对，将它们合并，然后局部进行边交换操作将剩余的三角形重新组合再融合。模型的特征线则通过移动顶点的位置来保持。文献^[130]提出了一个最优匹配算法，利用图论中著名的 Blossom 算法，在多项式时间内计算图中的最小代价完美匹配。通过一系列局部操作得到最佳的四边形布局，利用组合优化方法实现全局最优的匹配效果。提出的 Bloom-Quad 算法生成的网格在单元形状和效率方面都有较好的效果。然而，该方法的时间复杂度达到 n^2 （ n 为网格的元

素数量），对大模型不友好。

2.1.2 铺砌方法

铺砌方法基于前沿推进法（Advancing Front Technique, AFT），前沿推进算法对模型的复杂几何边界适应能力强，生成的最终网格质量也较好，是网格生成的主流算法之一。该算法思想适用于多种网格生成问题，不仅用于生成四边形网格^[116, 131]，也被应用于六面体网格生成^[132]。文献^[133-134]的研究还将其用于黏性边界层网格生成。

前沿推进算法由文献^[135]最早应用在平面二维区域三角形网格的自动生成，之后将其由三角形网格生成推广到四边形网格生成领域，得到了广泛应用。该方法从区域边界开始，按行推进，每次向所选区域内部推进一层单元，进而生成四边形网格。文献^[116]提出了一种称为“保存”的直接前沿推进方法，直接生成完整的四边形单元行。从外部或内部边界开始，向前推进到域的内部，生成的网格具有良好的贴边性，并且边界附近的元素通常形状良好，能应对复杂几何的边界问题，生成的网格质量高。此外，通过控制过程和各种清理及平滑措施，可以只保留很少的不规则节点，对于给定的区域进行旋转或平移也不会改变由此产生的网格拓扑。

铺砌法的经典方法是递归操作的，每次递归包括 3 个操作：

- 1) 寻找适于当前边界或所设约束的前沿；
- 2) 在区域内插入新的节点，或者连接已有的节点，应用局部贪婪算法确定新的单元元素；
- 3) 移去新单元，并更新前沿。

图 32^[136] 示例了铺砌方法。铺砌方法能应对复杂几何的边界问题，但其理论基础薄弱，无法保证算法的收敛性，且算法的细节繁琐，算法的鲁棒性不高。文献^[137]基于铺路法做了改进，从一排一排向前推进转变为一个单元一个单元向前推进。该方法在插入单个元素后立即对碰撞的网格前沿进行交叉和接近测试。新方法采用了单元面删除，提高了鲁棒性。铺砌法在推进过程中网格可能发生碰撞，碰撞的检测和解析通常需要花费大量时间，同时如何控制单个元素的尺寸以平滑整个网格的尺寸转换也是铺砌方法的另一个关键问题，尤其是当边界包含尺寸大不相同的元素时。文献^[138]根据现有的元素来控制尺寸分布，实现了整个网格的平滑尺寸转换，而无须事先生成背景网格。此外还通过

采用加快碰撞检查过程以及优化铺砌边界合并与分割的措施, 有效地检测和解决了碰撞问题。文献 [139] 提出了一种根据约束条件将区域分解为多个子区域的方法, 通过在每个满足约束条件的子区域中生成四边形网格, 从而实现具有复杂特征约束的几何模型的全自动四边形网格生成。文献 [140] 提出了一种从表面网格提取必要的几何特征, 应用超限插值的方式推波生成结构化四边形附面层网格。

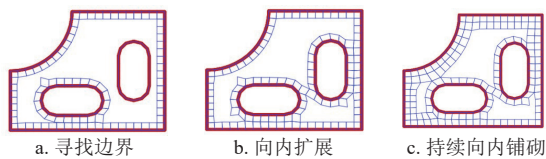


图 32 铺砌方法^[136]

2.1.3 映射法

映射法又称为参数变换法, 其基本思想是通过计算合理的映射函数, 将形状不规则的待剖分模型与形状规则的参数区域之间建立起对应关系, 然后在规则的参数空间上合理布置所需类型的网格, 将规则参数域进行网格剖分, 最后通过逆映射函数将参数域上生成的网格反向映射回实际的物理模型, 从而完成原模型的网格生成, 如图 33 所示。

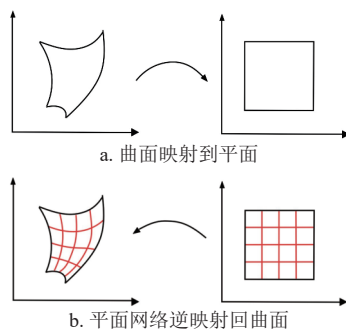


图 33 映射法

映射法的关键问题之一在于几何区域分解的合理性和自动化程度, 也即如何将复杂的模型自动化分解为可映射的简单子区域。文献 [141] 首先根据曲面几何信息 (法向量) 对网格中的面进行划分, 然后将输入的网格曲面分为多个近似平面的区域, 通过分析这些区域的几何形状来判断这些区域是否满足预设的形状标准 (第一阶段), 若满足标准, 则可以进行网格重划分, 将这些近似平面的区域边界作为新的网格边得到粗糙的四边形网格布局; 若不满足标准 (第二阶段), 由于初始划分的区域非常粗糙, 这种情况时有发生, 此时需要在高曲率区

域放置非规则顶点, 然后继续进行细化和清理, 优化区域划分, 进而将上一步的结果分解成为具有更规则形状的区域。最后, 将区域边界作为新的网格边, 以所在区域的近似面作为新的网格面片以得到粗糙的四边形网格布局, 这时再应用 Catmull-Clark 细分方法将网格细分成一个全四边形网格。类似地, 文献 [142] 根据曲面的主曲率、文献 [117] 根据曲面的曲线骨架对网格中的面进行划分, 最终将其细化为半规则四边形网格。

在一些复杂情况中, 如果不进行第二阶段的处理, 在第一阶段中得到的布局顶点集合甚至有可能无法提取有效的四边形布局。文献 [143] 提出了一种构建四边形网格布局的算法, 在第二阶段首先调整初始布局顶点。只要有利于布局质量, 这些顶点就可以在规定的半径内移动, 甚至可以移除。算法基于稳健的量化策略, 将连续的 T 网格结构转化为离散四边形布局。文献 [144] 提出一种基于 T 网络的量化算法, 首先从给定的无缝参数化中构建 T 网格, 以保证结果的质量。然后将整数长度分配给曲面的矩形分区的边。这种表示能够以有限的方式对无限多个布局连接选项进行编码, 从而保证了始终可以找到有效的四边形布局。同时, 用户可以指定布局边缘与规定方向的角度偏差的界限。最后通过求解一个整数线性规划 (ILP), 以找到符合该最大偏差的粗略四边形布局。该算法可以保证得到没有 T 节点且角度失真有限的、全局一致的四边形布局。

映射法的关键问题之二在于需要保证在相邻的两个规则参数域的公共边界上兼容, 这样边界两端的网格剖分面片才能无缝连接。文献 [145] 首先提出了一种模板法, 实现了网格在两个方向均能过渡的全四边形单元生成, 在一定程度上解决了物理问题中网格疏密自然过渡的要求, 并在文献 [146] 中简化了亲单元 (即参数域中相对应的区域) 上模板的分类方法, 利用单元密度要求, 自动计算亲单元的每条边上的单元分割数量, 可以有效地在两个方向上实现网格疏密过渡, 而对没有网格疏密过渡要求的区域可以生成一致的结构化网格, 以保证边界两端的网格剖分面片无缝连接。

从映射法的流程可以看出, 通过求解合理的映射, 将形状复杂的模型的网格生成问题转化为相对简单的规则区域内的网格生成问题, 从而降低了形状复杂模型网格生成的难度。使用此类方法生成的四边形网格可以保证较好的质量, 也可以控制参数区域内生成网格的密度, 通常可以生成半正则四边

形网格, 适用于结构化网格和非结构化网格的生成。但这类方法在生成的网格中引入更多角度畸变和不规则顶点。另外, 如何求解合适的映射, 如何将复杂的几何模型分解为可映射的简单子区域也是映射法的难点。尤其是对复杂模型, 复杂模型的分解一般离不开人工干预, 否则无法保证模型分解的质量和正确性, 因此自动化程度相对较低。除此之外, 由于方法的核心是区域分解, 因此无法保证参数域之间的网格密度连续变化。

2.1.4 基于场的方法

交叉场引导的四边形网格生成方法是近年来的研究热点, 许多基于场论的方法被相继提出^[147-149], 尤其是用场的分布来表示几何特征的方法得到了广泛研究。交叉场可以用于描述模型的几何特征, 并且以直接的方式在表面上生成高质量的四边形网格。场方法的基本思想是:

- 1) 在每个点定义初始交叉场, 并用合适的方法表示场;
- 2) 指定要对齐的边界条件(若有边界), 并通过最小化能量找到光滑场;
- 3) 通过跟踪场的流线将区域划分为四边形布局或使用其他方法得到参数化结果;
- 4) 将规则网格映射到每个区域。

基于场的每种方法首先都需要选择一种表示交叉场的方式, 如 N-RoSy 表示^[150]、周期跳跃技术^[151]和复值表示^[152]等。文献^[153]采用角度场表示交叉场, 使用主曲率来测量局部相对各向异性, 从而可以检测特征(尖锐)边缘。然后, 计算与这些特征边对齐的交叉场。由于角度是实数, 而角度的跳跃是整数值, 因此这本质上是一个混合整数问题。使用迭代求解器来求解与角度跳跃的偏导数相对应的线性方程, 该迭代求解器在每次迭代后对其结果进行舍入。全局参数化是根据交叉场构建的, 由于交叉场的对称性, 角度场被设计为具有 90° 的整数倍的跳跃, 最终生成一个完整的四边形网格。文献^[154]提出了新的场表示方法。他们将交叉场建模为复数场, 对应标架给出的 4 个方向。相关的能量是狄利克雷能量, 由以下两项组成: 复数的范数和相位。事实上, 交叉场应与单位范数的复数相对应。然而, 这种约束会使得奇异点的能量定义不清。最终导出: 对于单位范数的缩放版本的场, 该能量的最小化是指找到与相应拉普拉斯矩阵的最小特征值相关的特征向量。通过对特征向量进行归一化, 最终获得目标交叉场。

在场方法中, 模型表面的初始交叉场的生成是一项非常重要的基础工作, 一般情况下, 初始场是通过计算模型表面的主曲率来得到^[155], 文献^[156]也是首先给定一个由用户定义或由主曲率诱导的交叉场, 由角度场表示, 然后从交叉场构建参数化。然后, 这些方法通常通过能量最小化技术生成平滑的交叉场。场光滑度的典型度量是狄利克雷能量的离散形式^[157]。

最后, 基于获得的交叉场, 通过分析场的奇异性和奇异点分布, 使用流线跟踪技术^[142, 147]或参数化^[158]方法, 实现区域的自动分解继而生成模型的表面四边形网格, 如图 34 所示^[147]。文献^[159]在平面上实现了流线跟踪算法, 从奇点开始持续跟踪流线到达域的边界或另一个奇点。这种方法在大多数领域都能很好地工作, 但在具有极限环的几何结构上可能会失败。文献^[160]提出了一种基于边界元的方法, 用于将平面域自动划分为一组四边形区域。其基本思想是通过边界元法求解偏微分方程来生成一个精确的光滑交叉场, 然后通过分析交叉场的结构来识别奇异顶点, 并对从这些点发出的流线进行追踪和简化来分割输入区域, 该方法适用于半结构化四边形网格的生成。场方法为区域分块提供了新的思路, 表面场决定了生成四边形面片的方向和尺寸。当交叉场足够光滑时, 追踪的四边形网格流线也是光滑的, 能够生成高质量的四边形网格。

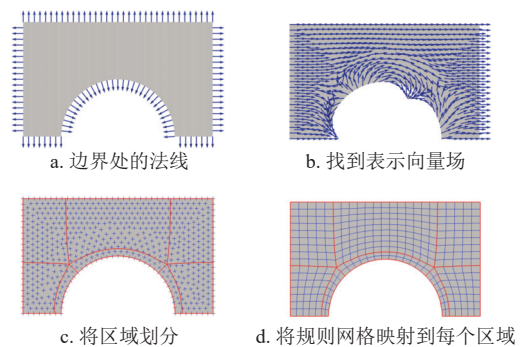


图 34 基于交叉场的网格划分方法^[147]

2.1.5 基于度量的方法

基于度量的方法也是一种流行的方法。该方法利用网格的度量来建立曲面上的点与参数域上的点之间的一一对应关系, 其本质是计算流形在参数域上的浸入。四边形网格首先在参数域上生成, 然后逆映射回原模型表面, 通过全局定义的参数化生成四边形网格。有时复杂曲面需要分解成一系列简单曲面, 然后将这些简单曲面组合起来, 形成复杂曲

面的最终网格, 但如何分解也是一个非常复杂的过程。文献 [115] 提出了一种全局参数化算法, 通过寻找锥奇异点的位置和角度以生成具有低度量失真的无缝参数化。该方法的核心思想是逐渐演变曲面的度量, 从原始度量开始, 以零高斯曲率约束曲面面积增长的部分, 曲率逐渐集中在一小组顶点上成为锥点, 使其参数化结果具有更低的度量失真。

基于度量的网格生成方法是一种可行的方法, 在理论上该方法是严格收敛的, 特别是基于曲率流的方法。该方法计算量小, 易于实现。然而, 基于度量的方法通常在特征保持方面表现不佳, 而且还需要精确的奇异点信息来确保所获得的四边形网格的面积畸变最小。

2.1.6 基于计算共形几何的方法

基于共形几何的方法奠定了结构化网格生成的理论基础, 解决了很多基本的理论问题, 同时提出了严密高效的算法。

文献 [161] 证明了曲面上四边形网格等价于黎曼面上的亚纯四次微分, 提出了一种基于黎曼面的阿贝尔-雅可比理论的四边形网格生成算法。该方法首先计算满足阿贝尔-雅可比方程的四边形网格奇异点, 然后应用动态 Ricci 流算法计算含有锥奇异点的平直度量, 之后进行共形结构的调整, 诱导出有效的四边形网格。基于该方法开发了四边形网格自动生成算法, 算法分为两个阶段: 1) 生成亚纯四次微分来诱导 T 网格; 2) 通过求解线性系统来调整 T 网格的边长, 以满足甲板变换条件, 从而生成四边形网格。文献 [162] 通过联络和平行移动引入四边形网格上的和乐群, 分析发现四边形网格的和乐群是平面旋转群的子群, 提出基于黎曼度量的四边形网格生成算法框架。通过离散 Ricci 流算法和共形结构变形技术, 可以得到满足四边形网格度量定理中所有条件的黎曼度量, 从而得到所求的四边形网格。文献 [163] 提出了一种在拓扑多环上生成具有最少奇异点的四边形网格算法。每一个顶点度为 4 的四边形网格会诱导一个全纯微分, 所有的全纯微分形成一个有限维线性空间。通过优化线性空间中的面积畸变能量, 可以生成尽可能均匀的四边形网格。该算法中所优化的能量可以被其他约束条件所替代, 因此相同的方法可推广应用于其他约束条件下的四边形网格自动生成问题。

2.2 六面体网格

当前六面体网格生成仍面临众多挑战和问题,

由于其生成难度大而被称为网格界的“神圣网格”。六面体网格生成方法可以分为基于模型整体的生成方法和基于模型分解的生成方法。基于模型整体的生成方法以整体几何模型为对象生成六面体网格, 现有的方法包括: 推波前进法、扫掠法、栅格法、多立方体法、标架场法等。基于模型分解的方法是基于“分而治之”的思想, 将模型分解为易于生成六面体网格的子块, 每个子块在保持分割截面网格一致的约束下, 对子块进行六面体网格划分, 进而拼接为整体网格, 方法主要包括: 基于扫掠体的分解方法、基于中轴线/骨架的分解方法。

2.2.1 推波前进法

推波前进法是以边界为起始, 向内部逐步推进生成六面体网格的方法。此类方法主要存在两个问题: 1) 在推进过程中, 生长的前沿会发生相交, 需要处理相应的几何问题; 2) 算法并不能保证生成一个可用的全六面体网格。针对第一个问题, 文献 [132] 提出了 H-morph 方法, 该方法以四面体网格作为输入, 根据相邻四面体的网格质量, 逐步将相邻四面体合并为六面体, 从而生成网格, 在生成过程中, 通过保留原有四面体网格的方式避免发生碰撞。针对第二个问题, 近几十年在理论和工程实践上都有大量研究。文献 [164] 证明在一个亏格为 0 的封闭曲面四边形网格中, 其内部能够划分六面体单元的充分必要条件是四边形网格的单元数目为偶数, 然而该结论对网格的几何质量无法保证。此外, 即便理论上存在六面体剖分, 如果在前沿推进的过程中出现边界为奇数个的封闭区域, 则无法继续用全六面体单元填充, 这就需要对网格进行回溯, 且不能保证可以获得更好的结果。文献 [165] 在输入的四边形曲面内部定义一个种子六面体网格, 之后分别由内向外和由外向内共同推进, 能够提高生成全六面体网格的成功率。但是该方法仍然局限于亏格为 0 的几何区域。综上所述, 推波前进法在生成高质量六面体网格的可靠性不足, 在很大程度上依赖于输入网格的质量, 如图 35 所示^[166]。

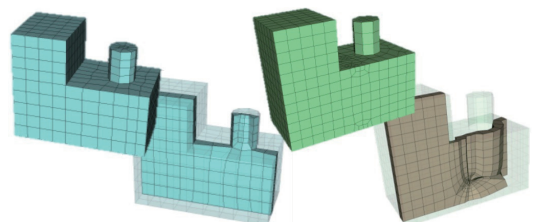


图 35 同一曲面不同边界网格对前沿推进法的影响^[166]

2.2.2 扫掠法

扫掠法是针对一类特殊结构——扫掠体模型，生成六面体网格的方法。该方法首先将模型表面分为源面（source）、目标面（target）以及侧面（side），如图 36 所示^[167]。之后在源面生成四边形网格，沿着侧面的方向，将网格扫掠到目标面上，扫掠的过程会在侧面生成结构化四边形网格，同时内部生成六面体网格。扫掠法可以分为一对一扫掠、多对一扫掠、多对多扫掠以及轴对称扫掠。

一对一扫掠仅处理具有单一源面和单一目标面的扫掠体，如何确定扫掠过程中节点的几何位置，是该方法的主要研究内容。直接采用源面投影到内部生成新网格点的方法，会在复杂外形中生成质量较差的网格，针对这一问题，研究者们提出了众多方法。文献 [168] 提出在生成网格后，利用平滑操作优化网格质量，但这种方式针对复杂外形，算法效率低，且最终网格的单元质量无法达到预期。文献 [169] 提出两种解决方法，一种方法是基于仿射变换矩阵，根据边界点的质心位置来放置内部点；另一种方法采用分层平滑的方法，分别对每层四边形网格进行平滑操作。然而当源面和目标面曲率的差别较大时，仍会产生质量差的网格单元。文献 [170-171] 通过最小二乘法，逼近源面和目标面的仿射变换，进而计算出内部点的位置，然而如果源面或目标面中存在凹面，仍会产生质量差甚至是负体积的网格单元。文献 [172] 采用调和映照的方法确定源面到目标面的映射，从而确定内部点位置，该方法能够在复杂模型中保证六面体网格质量，然而算法效率较低。

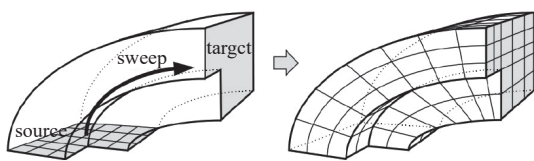


图 36 扫掠法示意图^[167]

一对一扫掠法往往无法适用复杂模型，为了扩展算法应用，衍生出了多对一扫掠法以及多对多扫掠法。多对一扫掠方法将模型分解为多个源面和一个目标面，可以进一步分解为多个一对一扫掠操作^[173]。多对多扫掠是多对一扫掠的进一步扩展，将模型分解为多个源面和多个目标面，进而能够被分解为多个多对一和一对一扫掠操作，实例模型如图 37 所示^[174]。扫掠法的难点主要在于识别扫掠面以及分解扫掠区域。文献 [175] 开发了一个工具，可以自

动处理和识别多对多的扫掠体，该方法将模型分为拓扑圆柱的扫掠子块，通过压印、匹配所有源面和目标面，生成高质量六面体网格。文献 [176] 通过对源面和目标面的边界进行布尔运算，将模型分解为多个多对一扫掠体，目前此类方法不能针对任意模型进行鲁棒地分解。

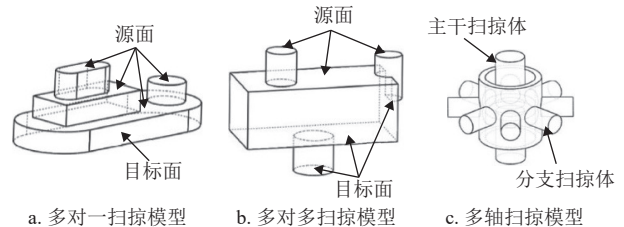


图 37 3 种不同类型的扫掠模型^[174]

对于具有不同扫掠方向的扫掠体构成的模型，如图 37c 所示，需采用多轴扫掠法对其生成扫掠网格。文献 [177] 提出了多轴分解的方法，按照先主干后分支的顺序生成六面体网格，但当源面和目标面出现内环模型，则无法处理。文献 [178] 提出了嫁接算法，将模型分为扫掠主干和分支，在生成网格时首先生成主干网格，之后在主干的侧面中提取出分支的嫁接源面，进而对分支网格进行扫掠，生成六面体网格。文献 [179] 首先生成模型表面的结构化四边形网格，然后对每一个扫掠子块生成扫掠网格，最后采用拓扑优化方法改善网格嫁接处的六面体单元质量。总而言之，扫掠法能够对扫掠体生成质量较高的六面体网格，广泛应用于 Abaqus、Hypermesh、Trelis 等商业软件中。但该类仅适用于扫掠体模型，方法的适用范围有限。

2.2.3 栅格法

栅格法是目前最为鲁棒的算法，可以针对任何几何模型自动化地生成六面体网格。该类方法利用笛卡尔网格坐标系将模型所在空间分割为栅格单元，之后将网格单元映射到原始模型内部，算法整体流程如图 38 所示。在划分笛卡尔网格时，应同时满足局部约束和全局约束。局部约束通常是指单元大小必须与输入模型的局部大小相匹配，以保持几何特征。全局约束则是指在笛卡尔网格中，必须存在一个网格单元的子集，使得其拓扑结构与原始模型的拓扑结构相匹配，即六面体网格的表面和输入曲面必须是同伦的，它们之间需要满足双射。



图 38 栅格法示意图

栅格法通过网格局部细化的方式, 来满足局部约束。常见的细化标准包括法向相似性^[180]、局部厚度相似性^[181-183]、表面逼近程度^[184]或以上标准的组合^[185]。为了满足全局约束, 通常需要依次细化所有网格单元, 直到存在一个网格单元的子集, 使得其拓扑结构与原始模型拓扑等价^[184]。细化的方式可以分为2-细化和3-细化。2-细化将一个六面体中的每条边分为两部分, 将一个六面体单元分为8个六面体单元; 3-细化将一个六面体中的每条边分为3部分, 将一个六面体单元分为27个六面体的单元。两种细化方式序列被编码为树结构, 2-细化对应于八叉树, 3-细化对应于27叉树。

细化后的笛卡尔网格不一定满足网格单元的一致性约束, 即相邻的网格单元之间的细化程度不同, 导致相邻的网格单元的大小不同, 从而产生悬挂节点。处理悬挂节点的方法分为两种: 1) 通过模板对原始网格进行修改; 2) 将网格视为一个六面体网格的对偶, 对其进行修改, 称之为对偶法。

在原始网格上进行修改通常使用3-细化。文献^[186]提出, 对于使用3-细化的算法, 如果相邻的网格单元之间的细化程度差异为一级, 则存在20种模板可以替换悬挂节点。然而在凹凸区域, 该方法会产生包含奇数个四边形单元的边界, 已知这种边界条件无法产生全六面体网格^[164]。因此, 文献^[187-189]额外添加了4种过渡方案用来解决凹凸区域无法生成全六面体网格的问题。但目前尚未得到一个完美的解决方案。

对偶法通常使用2-细分方法^[190-192]来处理悬挂节点。文献^[182]提出对偶结构能够诱导一个六面体网格的充分必要条件是, 对偶结构中所有点的相邻边的数目均为6, 且所有边的相邻面的数目均为4。基于上述结论, 文献^[181]提出了一组切割方案, 通过调整网格中单元的度, 得到满足要求的对偶结构, 进而得到一个纯六面体网格, 该方法通过成对地连接悬挂节点构成六面体网格的对偶网格。

生成六面体网格之后, 栅格法需要将边界顶点投影到原始曲面上。这个问题的难点是在保持特征的同时, 必须保证六面体网格中没有翻转单元。文献^[182, 193]通过迭代平滑的方法, 逐渐将顶点移动到边界上, 该方法在六面体单元发生翻转时可以方便地回溯, 但仍不能保证复杂模型的网格单元中没有翻转单元。文献^[194]可将非流形装配体模型利用栅格法划分六面体网格, 并一定程度保持尖锐特征, 然而算法并不鲁棒。文献^[195]根据表面曲

率、几何特征、密度场信息, 利用8种不同类型的模板, 指导细化操作, 并提出了一种以网格质量为目标函数的优化方法。通过将拉普拉斯平滑法与优化方法相结合, 提高网格质量, 保持网格特征。文献^[184]将八叉树网格边界与模型表面构建特征双射映射, 之后通过变形, 将八叉树边界保持特征地映射到原始曲面上。之后采用全局填充(Global Padding)技术, 保证所有六面体单元具有正雅可比值, 然而该算法的计算效率很低。

2.2.4 多立方体方法

多立方体方法能够生成结构化程度高的六面体网格。该方法首先将模型实体映射到一个嵌入在 Z^3 中的正交多面体(或称为多立方体), 然后对多立方体进行采样, 得到一个规则的全六面体网格, 最后根据逆映射将多立方体拉回原始模型内部, 整体计算流程如图39所示。该算法关键点是建立原始曲面和多立方体的映射关系。

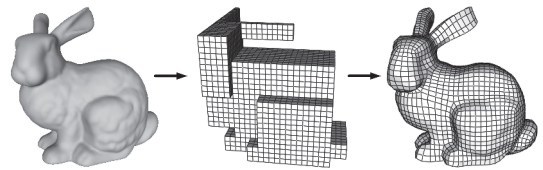


图39 多立方体法示意图

映射的方法是聚类法, 其中最常用的做法是将网格中每个面映射到距离法向最接近的轴上 $(\pm X, \pm Y, \pm Z)$ ^[196]。然而这种方式通常无法诱导一个全局非立方体单元。文献^[197]使用重心Voronoi镶嵌(Centroidal Voronoi Tessellation, CVT)对表面网格进行聚类, 形成多立方体结构。虽然结果有所改进, 但方法仍不鲁棒。现有算法通常需要通过启发式的算法对聚类结果进行后处理, 一类方法将具有多立方体标签的网格迭代旋转表面法向, 直到所有的标记可以构成多立方体结构^[198-199]; 另一类方法将变形与标记操作交替进行, 直到满足多立方体结构的条件^[200]。即便如此, 特征约束下的多立方体方法仍很难找到满足要求的映射。文献^[201]列出了能够构成有效多立方体的充分条件, 并提出相应算法, 即从一个需要后处理的面片出发, 系统地探索聚类空间, 以找到满足所有标准的聚类。但这种方法仍不能保证收敛到一个有效的多立方体结构, 因此也有研究者们提出了一些交互标记多立方体标签的工具^[202-204], 可以对已经标记的多立方体进行手动修改。

多立方体方法生成的六面体网格, 奇异线通常位于表面, 但在平坦的表面区域, 这些奇异线导致网格质量下降。针对该问题, 常用的方法是将网格表面进行层插入操作, 将奇异线向内推, 但是对于尖锐特征, 这种操作会破坏原有网格质量^[205]。文献 [205] 通过筛选层插入操作初始面的方法, 选择性地使奇异线推向六面体网格内部。文献 [206] 在多立方体映射时, 通过对多立方体结构进行切割的方式, 将奇异线向内移动。

2.2.5 标架场法

标架场法通过在网格的边界和内部生成标架场, 使得每个标架与模型在三维中的嵌入方式一致, 从而诱导一个体参数化, 进而确定网格内部排列, 整体流程如图 40 所示^[207]。该方法能够对一些模型生成高质量结构化的六面体网格, 然而标架场理论基础仍不完备, 能够生成六面体网格的标架场的充要条件尚未确定, 因此甚至对一些简单模型都会出现网格退化的情况, 但依然是一种很有前景的网格生成方法, 受到广泛关注。



图 40 标架场方法示意图^[207]

标架场可以看作是向量场的泛化, 用于局部描述一个立方体的形状。一个标架由 3 个线性无关的向量组成, 代表一个六面体。全局标架场有可能包含奇异线, 即当矢量绕着奇异线旋转时, 它不会返回到原来的位置。标架通常通过 3 个显式向量 $\mathbf{u}, \mathbf{v}, \mathbf{w} \in \mathbb{R}^3$ 来表示, 组成矩阵 $F = (\mathbf{u}, \mathbf{v}, \mathbf{w}) \in \mathbb{R}^{3 \times 3}$ 。在实践中, 常采用简化的表示方法, 例如单位四元数或欧拉角^[207-208]。但这些表示方法有一个缺点, 即不唯一性。如矩阵 $(\mathbf{u}, \mathbf{v}, \mathbf{w})$ 描述的六面体与矩阵 $(\mathbf{v}, \mathbf{w}, -\mathbf{u})$ 描述的六面体是相同的。另一类方法是通过八面体群^[209]来表示标架, 八面体是立方体的对偶体, 它们的对称变换是相同的, 可以将一个等价类中的元素减少到 24 个元素。得到标架场的奇异结构后, 可以基于奇异结构对待分割区域进行分解, 将模型分块, 进而生成六面体网格。文献 [210] 通过构建的标架场, 提取奇异结构, 从而获得模型分块。文献 [211] 根据提取标架场提取的分割线, 之后创建六面体网格对偶面, 进而对网格进行分块, 从而生成六面体网格, 如图 41 所示。此类方法需要根据六面体网格奇异结构对模型进行分割, 因此需要获得良好的奇异结构, 然而当前标架场由

于没有理论保证, 往往会得到不合理奇异线, 需要对奇异线进行修复。文献 [212] 利用与表面特征和曲率对齐的表面标架场作为引导, 获取切割环路, 均匀地覆盖输入表面, 避免了获取内部奇异线。之后根据环路向内部延申, 获取内部切割面, 从而分割模型, 生成六面体网格。该方法的鲁棒性有所提升, 但是并不能保证针对所有模型生成全六面体网格。

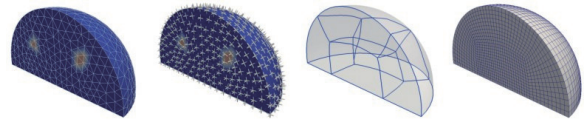


图 41 基于标架场分解的算法流程^[211]

2.2.6 基于扫掠体分解的方法

基于扫掠体分解的方法首先将模型分解为可扫掠体的子块, 之后扫掠生成六面体网格, 如图 42 所示, 不同颜色代表扫掠体的分解结果^[179]。基于扫掠体分解的方法其核心问题是将模型自动化地分解为可扫掠块, 并识别出扫掠路径。文献 [213] 通过构建包含几何模型的中轴和 Voronoi 图, 将模型分解为可扫掠的子块, 但会存在分解程度过大, 导致算法计算效率低的问题。文献 [214] 通过迭代地查找扫掠体, 对模型分解, 但是该方法需要扫掠路径严格垂直于模型底面, 且无法处理复杂扫掠体模型。文献 [215-216] 提出了一种基于中轴变换的扫掠体分解方法, 首先利用中轴变换自动识别模型中可扫掠子区域, 然后以交互的方式创建切割面, 但该方法没有考虑中轴退化的情况。文献 [217] 通过对模型表面法向进行模糊聚类, 得到潜在扫掠方向, 然后构建切割面, 将模型分割成一组扫掠子块, 但该分解方法仍不鲁棒。

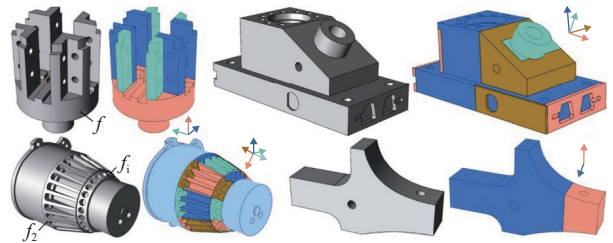


图 42 扫掠体分解实例模型^[179]

2.2.7 基于中轴线/骨架的方法

基于中轴线/骨架的方法是通过提取模型的中轴线/骨架, 首先将模型分解为不同子区域, 每个子区域都可以被划分为没有奇异线的六面体网格, 从而获得六面体网格的拓扑连接关系, 进而生成六

面体网格, 如图 43 所示^[218]。文献 [219-220] 首先提出了使用骨架的方法构建六面体网格, 将该方法首先应用于凸模型之后应用于平面区域和凹区域, 该方法的关键问题是如何处理分支处六面体网格的划分。文献 [218] 通过设置模板的方式确定骨架分支点的网格结构, 之后通过启发式的方法解决 T 型衔接问题。但针对一个分支点具有过多分支的模型, 该方法无法放置合理的六面体网格结构。文献 [221] 将骨架与前沿推进法相结合, 创建六面体占优的网格。该方法局限性较大, 依赖骨架的质量, 不同的骨架对应于不同的六面体网格结构。

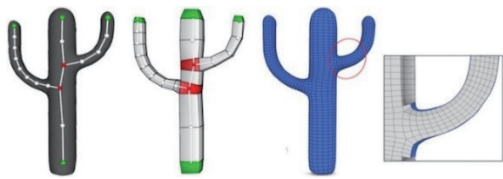


图 43 基于骨架的六面体网格生成方法示意图^[218]

2.2.8 基于计算共形几何的方法

文献 [222] 提出了一种实用的六面体网格生成自动算法, 从理论上证明了可染色四边形网格 (红-蓝网格)、带测度的叶状结构 (Measured Foliations) 和曲面上 Strebel 微分之间的等价关系, 为结构化六面体网格生成奠定了理论基础。算法流程如下:

- 1) 利用裤子分解 (Pants Decomposition), 将模型表面分解为多个子单元, 每个子单元是包含 3 个边界亏格为 0 的开曲面;
- 2) 根据组合类型和用户输入指定的高度计算唯一的 Strebel 微分;
- 3) 根据 Strebel 微分, 将曲面分解为拓扑圆柱面;
- 4) 为每个拓扑圆柱体生成六面体网格, 然后黏合形成全局一致的六面体网格。

该工作提出了一种新的曲面四边形网格方法, 即可着色四边形网格。可用于诱导高亏格封闭曲面内部的结构化六面体网格。方法构造的四边形网格和六面体网格都具有局部张量积结构, 奇异顶点和奇异线的数目最少且算法具有严格的几何性、自动性和几何共形性, 结果如图 44 所示。

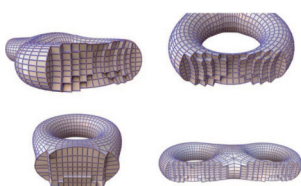


图 44 共形几何方法生成的六面体网格^[222]

3 结束语

本文对目前有限元分析领域中比较主流的网格生成方法进行了全面的综述。文章探讨了多种构建不同类型网格的技术, 包括无结构三角形表面网格、实体四面体网格、结构化四边形表面网格和实体六面体网格。针对每种方法, 文章分析了其算法的基本原理以及工程实践中的优缺点。

随着曲面和实体几何、拓扑复杂度的不断增加, 以及各种约束条件和工程需求的日益复杂, 网格生成算法面临着越来越多的挑战。现代微分几何、代数几何和代数拓扑方法开始深入渗透到网格生成领域, 为其奠定了坚实的理论基础, 并带来了更加严密和有效的新思路和方法。同时, 计算共形几何、人工智能技术的引入为网格生成带来了范式的改变, 预示着该领域有望迎来颠覆性的创新。

参考文献

- [1] FU L, HAN L H, HU X Y, et al. An isotropic unstructured mesh generation method based on a fluid relaxation analogy[J]. *Computer Methods in Applied Mechanics and Engineering*, 2019, 350: 396-431.
- [2] GVS P R, MONTAS H J, SAMET H, et al. Quadtree-based triangular mesh generation for finite element analysis of heterogeneous spatial data[C]//Proceedings of the Annual International Meeting of the American Society of Agricultural Engineers. Sacramento: American Society of Agricultural Engineers, 2001: 3072.
- [3] KARMAN S L, WYMAN N J. Automatic unstructured mesh generation with geometry attribution[C]//Proceedings of the AIAA Scitech 2019 Forum. Reston, Virginia: AIAA, 2019: 1721.
- [4] WANG Z, SRINIVASA A R, REDDY J, et al. Flowmesher: An automatic unstructured mesh generation algorithm with applications from finite element analysis to medical simulations[EB/OL]. [2024-09-01]. <https://arxiv.org/abs/2103.05640>.
- [5] DAI A, NIEBNER M. Scan2Mesh: From unstructured range scans to 3D meshes[C]//Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. New York: IEEE, 2019: 5574-5583.
- [6] HART K A, RIMOLI J J. MicroStructPy: A statistical microstructure mesh generator in Python[J]. *SoftwareX*, 2020, 12: 100595.
- [7] EDELSBRUNNER H, SEIDEL R. Voronoi diagrams and arrangements[J]. *Discrete & Computational Geometry*, 1986, 1(1): 25-44.
- [8] SEIDEL R. Voronoi diagrams in higher dimensions[D]. Graz: Graz University of Technology, 1982.
- [9] SEIDEL R. On the number of faces in higher-dimensional Voronoi diagrams[C]//Proceedings of the Third Annual Symposium on Computational Geometry. New York:

- ACM, 1987: 181-185.
- [10] TSAI V J D. Fast topological construction of delaunay triangulations and voronoi diagrams[J]. *Computers & Geosciences*, 1993, 19(10): 1463-1474.
- [11] CHARTRAND C, PEROT J B. A method for generating moving, orthogonal, area preserving polygonal meshes[J]. *Journal of Computational Physics*, 2022, 454: 110940.
- [12] WEATHERILL N P, HASSAN O. Efficient three-dimensional Delaunay triangulation with automatic point creation and imposed boundary constraints[J]. *International Journal for Numerical Methods in Engineering*, 1994, 37(12): 2005-2039.
- [13] CHEN J J, ZHAO D W, HUANG Z G, et al. Three-dimensional constrained boundary recovery with an enhanced Steiner point suppression procedure[J]. *Computers & Structures*, 2011, 89(5/6): 455-466.
- [14] CHEN J J, ZHENG Y. Redesign of a conformal boundary recovery algorithm for 3D Delaunay triangulation[J]. *Journal of Zhejiang University: Science A*, 2006, 7(12): 2031-2042.
- [15] CHEW L P. Guaranteed-quality mesh generation for curved surfaces[C]//Proceedings of the Ninth Annual Symposium on Computational Geometry. New York: ACM, 1993: 274-280.
- [16] RUPPERT J. A new and simple algorithm for quality 2-dimensional mesh generation[C]//Proceedings of the Fourth Annual ACM-SIAM Symposium on Discrete Algorithms. Austin Texas: Society for Industrial and Applied Mathematics, 1993: 83-92.
- [17] RUPPERT J. A delaunay refinement algorithm for quality 2-dimensional mesh generation[J]. *Journal of Algorithms*, 1995, 18(3): 548-585.
- [18] RIVARA M C, HITSCHFELD N, SIMPSON B. Terminal-edges Delaunay (small-angle based) algorithm for the quality triangulation problem[J]. *Computer-Aided Design*, 2001, 33(3): 263-277.
- [19] BOROUCAKI H, HECHT F, SALTEL E, et al. Reasonably efficient delaunay based mesh generator in 3 dimensions[C]//Proceedings of the 4th International Meshing Roundtable, Albuquerque. New Mexico: Academia Inc., 1999: 3-14.
- [20] REBAY S. Efficient unstructured mesh generation by means of delaunay triangulation and Bowyer-Watson algorithm[J]. *Journal of Computational Physics*, 1993, 106(1): 125-138.
- [21] GEORGE P L, HECHT F, SALTEL E. Automatic mesh generator with specified boundary[J]. *Computer Methods in Applied Mechanics and Engineering*, 1991, 92(3): 269-288.
- [22] ERTEN H, ÜNGÖR A. Quality triangulations with locally optimal steiner points[J]. *SIAM Journal on Scientific Computing*, 2009, 31(3): 2103-2130.
- [23] WENG Y J, CAO J, CHEN Z G. Global optimization of optimal Delaunay triangulation with modified whale optimization algorithm[J]. *Engineering with Computers*, 2024, 40(4): 2595-2616.
- [24] CIRROTTOLA L, RICCHIUTO M, FROEHLI A, et al. Adaptive deformation of 3D unstructured meshes with curved body fitted boundaries with application to unsteady compressible flows[J]. *Journal of Computational Physics*, 2021, 433: 110177.
- [25] YANG J X, WANG J, KWAK S, et al. A modified Allen-Cahn equation with a mesh size-dependent interfacial parameter on a triangular mesh[J]. *Computer Physics Communications*, 2024, 304: 109301.
- [26] ROBERTS K J, PRINGLE W J, WESTERINK J J. OceanMesh2D 1.0: MATLAB-based software for two-dimensional unstructured mesh generation in coastal ocean modeling[J]. *Geoscientific Model Development*, 2019, 12(5): 1847-1868.
- [27] WANG S Y, YANG B C, BAO H J, et al. Alternating size field optimizing and parameterization domain CAD model remeshing[J]. *Computer Aided Geometric Design*, 2024, 111: 102294.
- [28] BOROUCAKI H, GEORGE P L, MOHAMMADI B. Delaunay mesh generation governed by metric specifications. Part II. Applications[J]. *Finite Elements in Analysis and Design*, 1997, 25(1-2): 85-109.
- [29] BOROUCAKI 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.
- [30] GUO J W, DING F, JIA X H, et al. Automatic and high-quality surface mesh generation for CAD models[J]. *Computer-Aided Design*, 2019, 109: 49-59.
- [31] BEAUFORT P A, GEUZAIN C, REMACLE J F. Automatic surface mesh generation for discrete models—A complete and automatic pipeline based on reparametrization[J]. *Journal of Computational Physics*, 2020, 417: 109575.
- [32] LU W, LIANG J, REN M D, et al. Robust and fast CAD model tessellation for inspection[J]. *IEEE Transactions on Instrumentation and Measurement*, 2022, 71: 5007014.
- [33] VALLE Y. An efficient restricted quadtree triangulation based on the hilbert space-filling curve[C]//Spanish Computer Graphics Conference. Barcelona: Eurographics Association, 2008: 241-244.
- [34] YU K X, CHEN J J, FU K J, et al. On the efficiency of the advancing-front surface mesh generation algorithm[J]. *Computer-Aided Design*, 2022, 153: 103403.
- [35] NAKAHASHI K, SHAROV D. Direct surface triangulation using the advancing front method[C]//Proceedings of the 12th Computational Fluid Dynamics Conference. San Diego: AIAA, 1995: 1686.
- [36] BOROUCAKI H, LAUG P, GEORGE P L. Parametric surface meshing using a combined advancing-front generalized Delaunay approach[J]. *International Journal for Numerical Methods in Engineering*, 2000, 49(1-2): 233-259.
- [37] MA C D, HUANG J Z, LUO H, et al. CPAFT: Aconsistent parallel advancing front technique for unstructured triangular/tetrahedral mesh generation [EB/OL]. [2024-05-31]. <https://arxiv.org/abs/2405.20618>.
- [38] ZHOU Q Y, WANG Q Q, YU Z F. SAFT: Shotgun

- advancing front technique for massively parallel mesh generation on graphics processing unit[J]. *International Journal for Numerical Methods in Engineering*, 2022, 123(18): 4391-4406.
- [39] PERSSON P O, STRANG G. A simple mesh generator in MATLAB[J]. *SIAM Review*, 2004, 46(2): 329-345.
- [40] WANG Q S, YE J, WU H, et al. A triangular grid generation and optimization framework for the design of free-form gridshells[J]. *Computer-Aided Design*, 2019, 113: 96-113.
- [41] WANG Q S, GAO B Q, WU H. Triangular mesh generation on free-form surfaces based on bubble dynamics simulation[J]. *Engineering Computations*, 2019, 36(2): 646-663.
- [42] 郭宇飞, 赵康, 海永清. 面向有限元分析的三角网格布尔运算方法[J]. *系统仿真学报*, 2022(5): 1003-1014.
GUO Y F, KANG Z, HAI Y Q. Triangular mesh boolean operation method for finite element analysis[J]. *Journal of System Simulation*, 2022(5): 1103-1114.
- [43] ZENG Q, ZHU S J, LI Z N, et al. Self-adaptive triangular mesh generation framework for free-form single-layer reticulated shells based on virtual interaction forces[J]. *Automation in Construction*, 2023, 148: 104750.
- [44] ZENG Q, OHSAKI M, ZHANG J Y, et al. Structured triangular mesh generation method for free-form gridshells based on conformal mapping and virtual interaction forces[J]. *Engineering Structures*, 2023, 295: 116879.
- [45] LUO Q B. Automatic Delaunay mesh generation method and physically-based mesh optimization method on two-dimensional regions[J]. *Engineering with Computers*, 2022, 38(2): 1021-1031.
- [46] LEI N, LI Z Z, XU Z B, et al. What's the situation with intelligent mesh generation: A survey and perspectives[J]. *IEEE Transactions on Visualization and Computer Graphics*, 2024, 30(8): 4997-5017.
- [47] CHEN Y W, HE T, HUANG D, et al. MeshAnything: Artist-created mesh generation with autoregressive transformers[EB/OL]. [2024-06-14]. <https://arxiv.org/abs/2406.10163>.
- [48] WANG W Y, CEYLAN D, MECH R, et al. 3DN: 3D deformation network[C]//Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. New York: IEEE, 2019: 1038-1046.
- [49] RIOS T, KONG J W, VAN STEIN B, et al. Back to meshes: Optimal simulation-ready mesh prototypes for autoencoder-based 3D car point clouds[C]//Proceedings of the IEEE Symposium Series on Computational Intelligence. New York: IEEE, 2020: 942-949.
- [50] HANOCKA R, METZER G, GIRYES R, et al. Point2Mesh[J]. *ACM Transactions on Graphics*, 2020, 39(4): 1-12.
- [51] GUPTA K. Neural mesh flow: 3D manifold mesh generation via diffeomorphic flows[D]. San Diego: University of California, 2020.
- [52] BADKI A, GALLO O, KAUTZ J, et al. Meshlet priors for 3D mesh reconstruction[C]//Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. New York: IEEE, 2020: 2846-2855.
- [53] DAROYA R, ATIENZA R, CAJOTE R. REIN: Flexible mesh generation from point clouds[C]//Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. New York: IEEE, 2020: 1444-1453.
- [54] SHARP N, OVSJANIKOV M. PointTriNet: Learned triangulation of 3D point sets[C]//Computer Vision—ECCV 2020. [S. l.]: Springer, 2020: 762-778.
- [55] BOULCH A, LANGLOIS P A, PUY G, et al. NeeDrop: Self-supervised shape representation from sparse point clouds using needle dropping[C]//Proceedings of the International Conference on 3D Vision (3DV). New York: IEEE, 2021: 940-950.
- [56] RAKOTOSAONA M J, AIGERMAN N, MITRA N J, et al. Differentiable surface triangulation[EB/OL]. [2024-09-01]. <https://doi.org/10.48550/arXiv.2109.10695>.
- [57] RAKOTOSAONA M J, GUERRERO P, AIGERMAN N, et al. Learning delaunay surface elements for mesh reconstruction[C]//Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. New York: IEEE, 2021: 22-31.
- [58] MA B R, LIU Y S, HAN Z Z. Reconstructing surfaces for sparse point clouds with on-surface priors[C]//Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. New York: IEEE, 2022: 6305-6315.
- [59] CHEN Z Q, TAGLIASACCHI A, FUNKHOUSER T, et al. Neural dual contouring[J]. *ACM Transactions on Graphics*, 2022, 41(4): 1-13.
- [60] WICKRAMASINGHE U, REMELLI E, KNOTT G, et al. Voxel2Mesh: 3D mesh model generation from volumetric data[M]//Lecture Notes in Computer Science. Cham: Springer International Publishing, 2020: 299-308.
- [61] CHEN Z Q, ZHANG H. Neural marching cubes[EB/OL]. [2024-09-01]. <https://doi.org/10.48550/arXiv.2106.11272>.
- [62] MARUANI N, OVSJANIKOV M, ALLIEZ P, et al. PoNq: A neural QEM-based mesh representation [C]//Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. New York: IEEE, 2024: 3647-3657.
- [63] MESCHEDER L, OECHSLE M, NIEMEYER M, et al. Occupancy networks: Learning 3D reconstruction in function space[C]//Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. New York: IEEE, 2019: 4455-4465.
- [64] TANG J X, ZHOU H, CHEN X K, et al. Delicate textured mesh recovery from NeRF via adaptive surface refinement[C]//Proceedings of the IEEE/CVF International Conference on Computer Vision. New York: IEEE, 2023: 17693-17703.
- [65] RAKOTOSAONA M J, MANHARDT F, ARROYO D M, et al. NeRFMeshing: Distilling neural radiance fields into geometrically-accurate 3D meshes[C]//Proceedings of the International Conference on 3D Vision (3DV). New York: IEEE, 2024: 1156-1165.
- [66] SINHA A, UNMESH A, HUANG Q X, et al. SurfNet: Generating 3D shape surfaces using deep residual networks[C]//Proceedings of the IEEE Conference on

- Computer Vision and Pattern Recognition. New York: IEEE, 2017: 791-800.
- [67] WANG N Y, ZHANG Y D, LI Z W, et al. Pixel2Mesh: Generating 3D mesh models from single RGB images[C]//Lecture Notes in Computer Science. Cham: Springer International Publishing, 2018: 55-71.
- [68] WEN C, ZHANG Y D, LI Z W, et al. Pixel2Mesh: Multi-view 3D mesh generation via deformation[C]//Proceedings of the IEEE/CVF International Conference on Computer Vision. New York: IEEE, 2019: 1042-1051.
- [69] WANG N Y, ZHANG Y D, LI Z W, et al. Pixel2Mesh: 3D mesh model generation via image guided deformation[J]. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2021, 43(10): 3600-3613.
- [70] SUN J M, XIE Y M, CHEN L H, et al. NeuralRecon: Real-time coherent 3D reconstruction from monocular video[C]//Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. New York: IEEE, 2021: 15593-15602.
- [71] LI R Z, KARANAM S, LI R, et al. Learning local recurrent models for human mesh recovery[C]//Proceedings of the International Conference on 3D Vision (3DV). New York: IEEE, 2021: 555-564.
- [72] LI C J, PAN H, LIU Y, et al. Robust flow-guided neural prediction for sketch-based freeform surface modeling[J]. *ACM Transactions on Graphics*, 2018, 37(6): 1-12.
- [73] DU D, HAN X G, FU H B, et al. SAniHead: Sketching animal-like 3D character heads using a view-surface collaborative mesh generative network[J]. *IEEE Transactions on Visualization and Computer Graphics*, 2022, 28(6): 2415-2429.
- [74] MICHEL O, BAR-ON R, LIU R, et al. Text2Mesh: Text-driven neural stylization for meshes[C]//Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. New York: IEEE, 2022: 13482-13492.
- [75] HÖLLEIN L, CAO A, OWENS A, et al. Text2Room: Extracting textured 3D meshes from 2D text-to-image models[C]//Proceedings of the IEEE/CVF International Conference on Computer Vision. New York: IEEE, 2023: 7875-7886.
- [76] LIAO Y Y, DONNÉ S, GEIGER A. Deep marching cubes: Learning explicit surface representations[C]//Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. New York: IEEE, 2018: 2916-2925.
- [77] PARK J J, FLORENCE P, STRAUB J, et al. DeepSDF: Learning continuous signed distance functions for shape representation[C]//Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. New York: IEEE, 2019: 165-174.
- [78] ERLER P, GUERRERO P, OHRHALLINGER S, et al. Points2Surf learning implicit surfaces from point clouds[C]//Lecture Notes in Computer Science. Cham: Springer International Publishing, 2020: 108-124.
- [79] MA B R, HAN Z Z, LIU Y S, et al. Neural-pull: Learning signed distance functions from point clouds by learning to pull space onto surfaces[C]//Proceedings of the 38th International Conference on Machine Learning. [S.l.]: PMLR, 2021: 7246-7257.
- [80] WILLIAMS F, GOJCIC Z, KHAMIS S, et al. Neural fields as learnable kernels for 3D reconstruction[C]//Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. New York: IEEE, 2022: 18479-18489.
- [81] MARUANI N, KLOKOV R, OVSJANIKOV M, et al. VoroMesh: Learning watertight surface meshes with voronoi diagrams[C]//Proceedings of the IEEE/CVF International Conference on Computer Vision. New York: IEEE, 2023: 14519-14528.
- [82] VASWANI A, SHAZEER N, PARMAR N, et al. Attention is all you need[C]//Proceedings of the 31st International Conference on Neural Information Processing Systems. Red Hook: Curran Associates Inc., 2017: 6000-6010.
- [83] SIDDIQUI Y, ALLIEGRO A, ARTEMOV A, et al. MeshGPT: Generating triangle meshes with decoder-only transformers[C]//Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. New York: IEEE, 2024: 19615-19625.
- [84] EL GHARBI Y, PARRET-FRÉAUD A, BOVET C, et al. Two-level substructuring and parallel mesh generation for domain decomposition methods[J]. *Finite Elements in Analysis and Design*, 2021, 192: 103484.
- [85] JALLEPALLI A, GALBRAITH M, HAIMES R, et al. Non-uniform knot (NUK) SIAC post-processing of flow fields produced through unstructured grid adaptation and optimization[J]. *Journal of Computational Physics*, 2024, 514: 113238.
- [86] LI J J, GAO L, YE M L, et al. Topology optimization of irregular flow domain by parametric level set method in unstructured mesh[J]. *Journal of Computational Design and Engineering*, 2022, 9(1): 100-113.
- [87] LIU Y, SAPUTRA A A, WANG J C, et al. Automatic polyhedral mesh generation and scaled boundary finite element analysis of STL models[J]. *Computer Methods in Applied Mechanics and Engineering*, 2017, 313: 106-132.
- [88] YU F, ZENG Y, GUAN Z Q, et al. A robust Delaunay-AFT based parallel method for the generation of large-scale fully constrained meshes[J]. *Computers & Structures*, 2020, 228: 106170.
- [89] ZHANG Z Y, JIMACK P K, WANG H. MeshingNet3D: Efficient generation of adapted tetrahedral meshes for computational mechanics[J]. *Advances in Engineering Software*, 2021, 157: 103021.
- [90] XIAO Z F, OLLIVIER-GOOCH C, VAZQUEZ J D Z. Anisotropic tetrahedral mesh adaptation with improved metric alignment and orthogonality[J]. *Computer-Aided Design*, 2022, 143: 103136.
- [91] FREY P J, GEORGE P L. Mesh generation: Application to finite elements[M]. 2nd ed. [S.l.]: Wiley-ISTE, 2013.
- [92] SI H. TetGen, a Delaunay-based quality tetrahedral mesh generator[J]. *ACM Transactions on Mathematical Software*, 2015, 41(2): 1-36.

- [93] GÜRSOY H N. Tetrahedral finite element mesh generation from NURBS solid models[J]. *Engineering with Computers*, 1996, 12(3): 211-223.
- [94] SHEWCHUK J R. A condition guaranteeing the existence of higher-dimensional constrained Delaunay triangulations [C]//*Proceedings of the 14th Annual Symposium on Computational Geometry*. New York: ACM, 1998: 76-85.
- [95] SI H, GÄRTNER K. 3D boundary recovery by constrained Delaunay tetrahedralization[J]. *International Journal for Numerical Methods in Engineering*, 2011, 85(11): 1341-1364.
- [96] SI H, SHEWCHUK J R. Incrementally constructing and updating constrained Delaunay tetrahedralizations with finite-precision coordinates[J]. *Engineering with Computers*, 2014, 30(2): 253-269.
- [97] FANG Q Q, BOAS D A. Tetrahedral mesh generation from volumetric binary and grayscale images[C]//*Proceedings of the IEEE International Symposium on Biomedical Imaging: From Nano to Macro*. New York: IEEE, 2009: 1142-1145.
- [98] WEATHERILL N P, HASSAN O, MORGAN K, et al. Aerospace simulations on parallel computers using unstructured grids[J]. *International Journal for Numerical Methods in Fluids*, 2002, 40(1/2): 171-187.
- [99] ZHENG P, YANG Y, LIU Z W, et al. Parallel and automatic isotropic tetrahedral mesh generation of misaligned assemblies[J]. *CCF Transactions on High Performance Computing*, 2020, 2(2): 149-163.
- [100] DONG L, ZHANG Y H, GE Z P, et al. Study and verification of large-scale parallel mesh generation algorithm for centrifugal pump[J]. *Mathematical Problems in Engineering*, 2020(11): 1956852.
- [101] G S P, SINGH T. Tetrahedral 3D mesh generation for medical images[C]//*Proceedings of the 3rd International Conference on Inventive Systems and Control*. New York: IEEE, 2019: 443-449.
- [102] MOLINO N, BRIDSON R, FEDKIW R. Tetrahedral mesh generation for deformable bodies[EB/OL]. [2024-08-15]. <http://scroll.stanford.edu/papers/meshing-sig03/>.
- [103] TERAN J, MOLINO N, FEDKIW R, et al. Adaptive physics based tetrahedral mesh generation using level sets[J]. *Engineering with Computers*, 2005, 21(1): 2-18.
- [104] LIANG X H, ZHANG Y J. An octree-based dual contouring method for triangular and tetrahedral mesh generation with guaranteed angle range[J]. *Engineering with Computers*, 2014, 30(2): 211-222.
- [105] ZUO J Z, DENG X M, SUTTON M A. Advances in tetrahedral mesh generation for modelling of three-dimensional regions with complex, curvilinear crack shapes[J]. *International Journal for Numerical Methods in Engineering*, 2005, 63(2): 256-275.
- [106] FLEISCHMANN P, SELBERHERR S. Three-dimensional delaunay mesh generation using a modified advancing front approach[EB/OL]. [2024-08-15]. https://www.iue.tuwien.ac.at/pdf/ib_1997/CP1997Fleischmann_1.pdf.
- [107] JI Z, FU L, HU X Y, et al. A consistent parallel isotropic unstructured mesh generation method based on multi-phase SPH[J]. *Computer Methods in Applied Mechanics and Engineering*, 2020, 363: 112881.
- [108] GAO J, CHEN W Z, XIANG T, et al. Learning deformable tetrahedral meshes for 3D reconstruction [EB/OL]. [2024-08-30]. <https://arxiv.org/abs/2011.01437>.
- [109] GAO W, WANG A, METZER G, et al. TetGAN: A convolutional neural network for tetrahedral mesh generation[C]//*British Machine Vision Conference*. London: BMVA, 2022: 365.
- [110] ZHENG Y, WU L, LIU X, et al. Neural volumetric mesh generator[C]//*NeurIPS 2022 Workshop on Score-Based Methods*. New Orleans, Louisiana: SBM, 2022: 23.
- [111] FARAZI M, YANG Z, ZHU W H, et al. TetCNN: convolutional neural networks on tetrahedral meshes [M]//*Lecture Notes in Computer Science*. Cham: Springer Nature Switzerland, 2023: 303-315.
- [112] GRUAU C, COUPEZ T. 3D tetrahedral, unstructured and anisotropic mesh generation with adaptation to natural and multidomain metric[J]. *Computer Methods in Applied Mechanics and Engineering*, 2005, 194(48/49): 4951-4976.
- [113] LIN H J, LIU H, WEI P. A parallel parameterized level set topology optimization framework for large-scale structures with unstructured meshes[J]. *Computer Methods in Applied Mechanics and Engineering*, 2022, 397: 115112.
- [114] MONTENEGRO R, CASCÓN J M, ESCOBAR J M, et al. An automatic strategy for adaptive tetrahedral mesh generation[J]. *Applied Numerical Mathematics*, 2009, 59(9): 2203-2217.
- [115] MYLES A, ZORIN D. Global parametrization by incremental flattening[EB/OL]. [2024-09-01]. <http://doi.acm.org/10.1145/2185520.2185605>.
- [116] BLACKER T D, STEPHENSON M B. Paving: A new approach to automated quadrilateral mesh generation[J]. *International Journal for Numerical Methods in Engineering*, 1991, 32(4): 811-847.
- [117] USAI F, LIVESU M, PUPPO E, et al. Extraction of the quad layout of a triangle mesh guided by its curve skeleton[J]. *ACM Transactions on Graphics*, 2015, 35(1): 1-13.
- [118] TONG Y, ALLIEZ P, COHEN-STEINER D, et al. Designing quadrangulations with discrete harmonic forms[C]//*Proceedings of the 4th Eurographics Symposium on Geometry Processing*. Cagliari: Eurographics Association, 2006: 1-11.
- [119] BOMMES D, VOSSEMER T, KOBBELT L. Quadrangular parameterization for reverse engineering [C]//*Mathematical Methods for Curves and Surfaces, 7th International Conference*. Tnsberg, Norway: Springer, 2008: 55-69.
- [120] DONG S, BREMER P T, GARLAND M, et al. Spectral surface quadrangulation[J]. *ACM Transactions on Graphics*, 2006, 25(3): 1057-1066.
- [121] VIERTEL R, OSTING B. An approach to quad meshing based on harmonic cross-valued maps and the Ginzburg:

- Landau theory[J]. *SIAM Journal on Scientific Computing*, 2019, 41(1): A452-A479.
- [122] LEI N, ZHENG X P, LUO Z X, et al. Quadrilateral mesh generation II: Meromorphic quartic differentials and Abel–Jacobi condition[J]. *Computer Methods in Applied Mechanics and Engineering*, 2020, 366: 112980.
- [123] ZHENG X P, ZHU Y M, CHEN W, et al. Quadrilateral mesh generation III: Optimizing singularity configuration based on Abel–Jacobi theory[J]. *Computer Methods in Applied Mechanics and Engineering*, 2021, 387: 114146.
- [124] CATMULL E, CLARK J. Recursively generated B-spline surfaces on arbitrary topological meshes[J]. *Computer-Aided Design*, 1978, 10(6): 350-355.
- [125] ELGINDY H, EVERETT H, TOUSSAINT G. Slicing an ear using prune-and-search[J]. *Pattern Recognition Letters*, 1993, 14(9): 719-722.
- [126] VELHO L, ZORIN D. 4–8 subdivision[J]. *Computer Aided Geometric Design*, 2001, 18(5): 397-427.
- [127] TARINI M, PIETRONI N, CIGNONI P, et al. Practical quad mesh simplification[J]. *Computer Graphics Forum*, 2010, 29(2): 407-418.
- [128] GURUNG T, LANEY D, LINDSTROM P, et al. Squad: Compact representation for triangle meshes[J]. *Computer Graphics Forum*, 2011, 30(2): 355-364.
- [129] LO S H. Generating quadrilateral elements on plane and over curved surfaces[J]. *Computers & Structures*, 1989, 31(3): 421-426.
- [130] REMACLE J F, LAMBRECHTS J, SENY B, et al. Blossom-Quad: A non-uniform quadrilateral mesh generator using a minimum-cost perfect-matching algorithm[J]. *International Journal for Numerical Methods in Engineering*, 2012, 89(9): 1102-1119.
- [131] CASS R J, BENZLEY S E, MEYERS R J, et al. Generalized 3-D paving: An automated quadrilateral surface mesh generation algorithm[J]. *International Journal for Numerical Methods in Engineering*, 1996, 39(9): 1475-1489.
- [132] OWEN S J, SAIGAL S. H-Morph: An indirect approach to advancing front hex meshing[J]. *International Journal for Numerical Methods in Engineering*, 2000, 49(1/2): 289-312.
- [133] KALLINDERIS Y, KHAWAJA A, MCMORRIS H. Hybrid prismatic/tetrahedral grid generation for complex geometries[C]//Proceedings of the 33rd Aerospace Sciences Meeting and Exhibit. Reston, Virginia: AIAA, 1995: 211.
- [134] KALLINDERIS Y, WARD S. Prismatic grid generation for three-dimensional complex geometries[J]. *AIAA Journal*, 1993, 31(10): 1850-1856.
- [135] LO S H. A new mesh generation scheme for arbitrary planar domains[J]. *International Journal for Numerical Methods in Engineering*, 1985, 21(8): 1403-1426.
- [136] DUAN J, CHEN X M, QI H, et al. Boundary-constraint meshing based on paving method[J]. *Applied Mechanics and Materials*, 2014, 627: 262-267.
- [137] WHITE D R, KINNEY P. Re-design of the paving algorithm: Robustness enhancements through element by element meshing[C]//Proceedings of the 6th International Meshing Roundtable, Albuquerque. New Mexico: Sandia National Laboratories, 1997: 830-843.
- [138] ZHAO Y, YU B, TAO W Q. An improved paving method of automatic quadrilateral mesh generation[J]. *Numerical Heat Transfer, Part B: Fundamentals*, 2013, 64(3): 218-238.
- [139] MA X W, SUN L. An automatic approach to constrained quadrilateral mesh generation[J]. *Engineering Computations*, 2019, 37(3): 929-951.
- [140] LU F S, PANG Y F, JIANG X, et al. Automatic generation of structured multiblock boundary layer mesh for aircrafts[J]. *Advances in Engineering Software*, 2018, 115: 297-313.
- [141] BOIER-MARTIN I, RUSHMEIER H, JIN J Y. Parameterization of triangle meshes over quadrilateral domains[C]//Proceedings of the 2004 Eurographics/ACM Siggraph Symposium on Geometry Processing. New York: ACM, 2004: 193-203.
- [142] CAMPEN M, BOMMES D, KOBBELT L. Dual loops meshing: Quality quad layouts on manifolds[J]. *ACM Transactions on Graphics*, 2012, 31(4): 1-11.
- [143] LYON M, CAMPEN M, KOBBELT L. Simpler quad layouts using relaxed singularities[J]. *Computer Graphics Forum*, 2021, 40(5): 169-180.
- [144] LYON M, CAMPEN M, KOBBELT L. Quad layouts via constrained T-mesh quantization[J]. *Computer Graphics Forum*, 2021, 40(2): 305-314.
- [145] 李华, 程耿东, 顾元宪. 一种新的全四边形网格快速生成方法: 模板法[J]. *计算结构力学及其应用*, 1996, 13(1): 25-33.
- LI H, CHENG G D, GU Y X. A new method for fast mesh generation of quadrilateral finite elements—pattern module’s method[J]. *Chinese Journal of Computational Mechanics*, 1996, 13(1): 25-33.
- [146] 李华, 李笑牛, 程耿东, 等. 一种全四边形网格生成方法: 改进模板法[J]. *计算力学学报*, 2002, 19(1): 16-19.
- LI H, LI X N, CHENG G D, et al. A method for mesh generation of all quadrilateral finite elements—Improved pattern module’s method[J]. *Chinese Journal of Computational Mechanics*, 2002, 19(1): 16-19.
- [147] VIERTEL R, OSTING B, STATEN M. Coarse quad layouts through robust simplification of cross field separatrix partitions[EB/OL]. [2024-06-14]. <https://arxiv.org/abs/1905.09097>.
- [148] BLANCHI V, CORMAN É, RAY N, et al. Global parametrization based on ginzburg-landau functional [C]//Numerical Geometry, Grid Generation and Scientific Computing: Proceedings of the 10th International Conference. Moscow: Springer, 2021: 251-262.
- [149] RAZAFINDRAZAKA F H, REITEBUCH U, POLTHIER K. Perfect matching quad layouts for manifold meshes[J]. *Computer Graphics Forum*, 2015, 34(5): 219-228.
- [150] PALACIOS J, ZHANG E. Rotational symmetry field design on surfaces[J]. *ACM Transactions on Graphics*,

- 2007, 26(99): 55.
- [151] LI W C, VALLET B, RAY N, et al. Representing higher-order singularities in vector fields on piecewise linear surfaces[J]. *IEEE Transactions on Visualization and Computer Graphics*, 2006, 12(5): 1315-1322.
- [152] RAY N, SOKOLOV D. Robust polylines tracing for N-symmetry direction field on triangulated surfaces [EB/OL]. [2024-09-01]. <http://dx.doi.org/10.1145/2602145>.
- [153] BOMMES D, ZIMMER H, KOBELT L. Mixed-integer quadrangulation[J]. *ACM Transactions on Graphics*, 2009, 28(3): 1-10.
- [154] KNÖPPEL F, CRANE K, PINKALL U, et al. Globally optimal direction fields[J]. *ACM Transactions on Graphics*, 2013, 32(4): 1-10.
- [155] RAY N, VALLET B, LI W C, et al. N-symmetry direction field design[J]. *ACM Transactions on Graphics*, 2008, 27(2): 1-13.
- [156] RAY N, LI W C, LÉVY B, et al. Periodic global parameterization[J]. *ACM Transactions on Graphics*, 2006, 25(4): 1460-1485.
- [157] JIANG T F, FANG X Z, HUANG J, et al. Frame field generation through metric customization[J]. *ACM Transactions on Graphics*, 2015, 34(4): 1-11.
- [158] EBKE H C, CAMPEN M, BOMMES D, et al. Level-of-detail quad meshing[J]. *ACM Transactions on Graphics*, 2014, 33(6): 1-11.
- [159] KOWALSKI N, LEDOUX F, FREY P. A PDE based approach to multidomain partitioning and quadrilateral meshing[C]//Proceedings of the 21st International Meshing Roundtable. Berlin: Springer, 2013: 137-154.
- [160] XIAO Z F, HE S P, XU G, et al. A boundary element-based automatic domain partitioning approach for semi-structured quad mesh generation[J]. *Engineering Analysis with Boundary Elements*, 2020, 113: 133-144.
- [161] 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.
- [162] CHEN W, ZHENG X P, KE J Y, et al. Metric based quadrilateral mesh generation[EB/OL]. [2024-06-14]. <https://arxiv.org/abs/1811.12604>.
- [163] ZHAO Z, FANG S Y, LEI N, et al. Optimal surface quadrilateral mesh generation[C]//Proceedings of the 2024 International Meshing Roundtable (IMR). Philadelphia, PA: Society for Industrial and Applied Mathematics, 2024: 14-27.
- [164] MITCHELL S A. A characterization of the quadrilateral meshes of a surface which admit a compatible hexahedral mesh of the enclosed volume[C]//Proceedings of the 13th Annual Symposium on Theoretical Aspects of Computer Science. Berlin: Springer, 1996: 465-476.
- [165] RUIZ-GIRONÉS E, ROCA X, SARRATE J. The receding front method applied to hexahedral mesh generation of exterior domains[J]. *Engineering with Computers*, 2012, 28(4): 391-408.
- [166] LEDOUX F, WEILL J C. An extension of the reliable whisker weaving algorithm[C]//International Meshing Roundtable Conference. Berlin, Heidelberg: Springer Berlin Heidelberg, 2007: 215-232.
- [167] OWEN S J. A survey of unstructured mesh generation technology[C]//Proceedings of the 7th International Meshing Roundtable, Michigan: [s. n.], 1998: 239-267.
- [168] MINGWU L, BENZLEY S E, SJAARDEMA G, et al. A multiple source and target sweeping method for generating all-hexahedral finite element meshes[C]//Proceedings of the 5th International Meshing Roundtable. Pittsburgh, PA: Sandia National Laboratories, 1996: 217-225.
- [169] KNUPP P M. Next-generation sweep tool: A method for generating all-hex meshes on two-and-one-half dimensional geometries[C]//Proceedings of the 7th International Meshing Roundtable. Dearborn, MI: Sandia National Laboratories, 1998: 505-513.
- [170] ROCA X, SARRATE J, HUERTA A. Surface mesh projection for hexahedral mesh generation by sweeping[C]//Proceedings of the 13th International Meshing Roundtable. Williamsburg, Virginia: Sandia National Laboratories, 2004: 169-180.
- [171] ROCA X, SARRATE J. Least-squares approximation of affine mappings for sweep mesh generation: Functional analysis and applications[J]. *Engineering with Computers*, 2013, 29(1): 1-15.
- [172] CAI S Y, TAUTGES T J. One-to-one sweeping based on harmonic S-T mappings of facet meshes and their cages[J]. *Engineering with Computers*, 2015, 31(3): 439-452.
- [173] SCOTT M A, BENZLEY S E, OWEN S J. Improved many-to-one sweeping[J]. *International Journal for Numerical Methods in Engineering*, 2006, 65(3): 332-348.
- [174] 王瑞, 高曙明, 吴海燕. 六面体网格生成和优化研究进展[J]. *计算机辅助设计与图形学学报*, 2020, 32(5): 693-708.
- WANG R, GAO S M, WU H Y. Progress in hexahedral mesh generation and optimization[J]. *Journal of Computer-Aided Design & Computer Graphics*, 2020, 32(5): 693-708.
- [175] BLACKER T. The cooper tool[C]//Proceedings of the 5th International Meshing Roundtable. Pittsburgh, PA: Sandia National Laboratories, 1996: 13-29.
- [176] LAI M W, BENZLEY S, WHITE D. Automated hexahedral mesh generation by generalized multiple source to multiple target sweeping[J]. *International Journal for Numerical Methods in Engineering*, 2000, 49(1/2): 261-275.
- [177] MIYOSHI K, BLACKER T. Hexahedral Mesh Generation Using Multi-Axis Cooper Algorithm[C]//Proceedings of the 9th International Meshing Roundtable. New Orleans, CA: Sandia National Laboratories, 2000: 89-97.
- [178] JANKOVICH S R, BENZLEY S E, SHEPHERD J F, et al. The graft tool: An all-hexahedral transition algorithm for creating a multi-directional swept volume mesh[J]. *Surfaces*, 1999, 5: 6.

- [179] WU H Y, GAO S M, WANG R, et al. A global approach to multi-axis swept mesh generation[J]. *Procedia Engineering*, 2017, 203: 414-426.
- [180] ITO Y, SHIH A M, SONI B K. Octree-based reasonable-quality hexahedral mesh generation using a new set of refinement templates[J]. *International Journal for Numerical Methods in Engineering*, 2009, 77(13): 1809-1833.
- [181] LIVESU M, PITZALIS L, CHERCHI G. Optimal dual schemes for adaptive grid based hexmeshing[J]. *ACM Transactions on Graphics*, 2022, 41(2): 1-14.
- [182] MARÉCHAL L. Advances in octree-based all-hexahedral mesh generation: Handling sharp features [C]//Proceedings of the 18th International Meshing Roundtable. Berlin, Heidelberg: Springer Berlin Heidelberg, 2009: 65-84.
- [183] PITZALIS L, LIVESU M, CHERCHI G, et al. Generalized adaptive refinement for grid-based hexahedral meshing[J]. *ACM Transactions on Graphics*, 2021, 40(6): 1-13.
- [184] GAO X F, SHEN H X, PANOZZO D. Feature preserving octree-based hexahedral meshing[J]. *Computer Graphics Forum*, 2019, 38(5): 135-149.
- [185] BAWIN A, HENROTTE F, REMACLE J F. Automatic feature-preserving size field for three-dimensional mesh generation[J]. *International Journal for Numerical Methods in Engineering*, 2021, 122(18): 4825-4847.
- [186] WEILER F, SCHINDLER R, SCHNEIDERS R. Automatic geometry-adaptive generation of quadrilateral and hexahedral element meshes for the FEM[C]//International Conference on Numerical Grid Generation in Computational Fluid Dynamics and Related Fields. Starkville: Pineridge Press, 1996: 1-5.
- [187] SCHNEIDERS R. An algorithm for the generation of hexahedral element meshes based on an octree technique[C]//Proceedings of the 6th International Meshing Roundtable. Park City, UT: Sandia National Laboratories, 1997: 195-196.
- [188] SCHNEIDERS R. Octree-based hexahedral mesh generation[J]. *International Journal of Computational Geometry & Applications*, 2000, 10(4): 383-398.
- [189] TACK L H, SCHNEIDERS R, DEBYE J, et al. Two- and three-dimensional remeshing, mesh refinement and application to simulation of micromechanical processes[J]. *Computational Materials Science*, 1994, 3(2): 241-246.
- [190] EBEIDA M S, PATNEY A, OWENS J D, et al. Isotropic conforming refinement of quadrilateral and hexahedral meshes using two-refinement templates[J]. *International Journal for Numerical Methods in Engineering*, 2011, 88(10): 974-985.
- [191] OWEN S J, SHIH R M, ERNST C D. A template-based approach for parallel hexahedral two-refinement[J]. *Computer-Aided Design*, 2017, 85: 34-52.
- [192] ZHANG Y J, LIANG X H, XU G L. A robust 2-refinement algorithm in octree or rhombic dodecahedral tree based all-hexahedral mesh generation[J]. *Computer Methods in Applied Mechanics and Engineering*, 2013, 256: 88-100.
- [193] LIN H W, JIN S N, LIAO H W, et al. Quality guaranteed all-hex mesh generation by a constrained volume iterative fitting algorithm[J]. *Computer-Aided Design*, 2015, 67: 107-117.
- [194] QIAN J, ZHANG Y J. Automatic unstructured all-hexahedral mesh generation from B-Reps for non-manifold CAD assemblies[J]. *Engineering with Computers*, 2012, 28(4): 345-359.
- [195] HUANG L L, ZHAO G Q, WANG Z L, et al. Adaptive hexahedral mesh generation and regeneration using an improved grid-based method[J]. *Advances in Engineering Software*, 2016, 102: 49-70.
- [196] GREGSON J, SHEFFER A, ZHANG E. All-hex mesh generation via volumetric PolyCube deformation[J]. *Computer Graphics Forum*, 2011, 30(5): 1407-1416.
- [197] HU K K, ZHANG Y J. Centroidal Voronoi tessellation based polycube construction for adaptive all-hexahedral mesh generation[J]. *Computer Methods in Applied Mechanics and Engineering*, 2016, 305: 405-421.
- [198] FANG X Z, XU W W, BAO H J, et al. All-hex meshing using closed-form induced polycube[J]. *ACM Transactions on Graphics*, 2016, 35(4): 1-9.
- [199] HUANG J, JIANG T F, SHI Z Y, et al. ℓ_1 -based construction of polycube maps from complex shapes[J]. *ACM Transactions on Graphics*, 2014, 33(3): 1-11.
- [200] FU X M, BAI C Y, LIU Y. Efficient volumetric PolyCube-map construction[J]. *Computer Graphics Forum*, 2016, 35(7): 97-106.
- [201] LIVESU M, VINING N, SHEFFER A, et al. PolyCut[J]. *ACM Transactions on Graphics*, 2013, 32(6): 1-12.
- [202] LI L X, ZHANG P, SMIRNOV D, et al. Interactive all-hex meshing via cuboid decomposition[EB/OL]. [2024-09-01]. <https://doi.org/10.1145/3478513.3480568>.
- [203] YU Y, LIU J G, ZHANG Y J. Hex-dom: Polycube-based hexahedral-dominant mesh generation[C]//Mesh Generation and Adaptation: Cutting-Edge Techniques. [S. l.]: Springer, 2022: 137-155.
- [204] YU Y X, WEI X D, LI A R, et al. HexGen and Hex2Spline: Polycube-based hexahedral mesh generation and spline modeling for isogeometric analysis applications in LS-DYNA[C]//Springer INdAM Series. Cham: Springer International Publishing, 2022: 333-363.
- [205] CHERCHI G, ALLIEZ P, SCATENI R, et al. Selective padding for polycube-based hexahedral meshing[J]. *Computer Graphics Forum*, 2019, 38(1): 580-591.
- [206] GUO H X, LIU X H, YAN D M, et al. Cut-enhanced PolyCube-maps for feature-aware all-hex meshing[J]. *ACM Transactions on Graphics (TOG)*, 2020, 39(4): 1-14.
- [207] LIU H, ZHANG P, CHIEN E, et al. Singularity-constrained octahedral fields for hexahedral meshing[J]. *ACM Transactions on Graphics*, 2018, 37(4): 1-17.
- [208] GAO X F, JAKOB W, TARINI M, et al. Robust hex-dominant mesh generation using field-guided polyhedral agglomeration[J]. *ACM Transactions on Graphics*, 2017, 36(4): 1-13.

- [209] SOLOMON J, VAXMAN A, BOMMES D. Boundary element octahedral fields in volumes[J]. *ACM Transactions on Graphics*, 2017, 36(4): 16-24.
- [210] KOWALSKI N, LEDOUX F, FREY P. Block-structured hexahedral meshes for CAD models using 3D frame fields[J]. *Procedia Engineering*, 2014, 82: 59-71.
- [211] ZHENG Z H, WANG R, GAO S M, et al. Dual surface based approach to block decomposition of solid models[M]//*Lecture Notes in Computational Science and Engineering*. Cham: Springer International Publishing, 2019: 149-167.
- [212] LIVESU M, PIETRONI N, PUPPO E, et al. Loopycuts: Practical feature-preserving block decomposition for strongly hex-dominant meshing[J]. *ACM Trans Graph*, 2020, 39(4): 121.
- [213] SHEFFER A, ETZION M, RAPPOPORT A, et al. Hexahedral mesh generation using the embedded voronoi graph[J]. *Engineering with Computers*, 1999, 15(3): 248-262.
- [214] SHIH B, SAKURAI H. Automated hexahedral mesh generation by swept volume decomposition and recomposition[C]//*Proceedings of the 5th International Meshing Roundtable*. Pittsburgh, PA: Sandia National Laboratories, 1996: 280-288.
- [215] LU J H C, SONG I, QUADROS W R, et al. Volumetric decomposition via medial object and pen-based user interface for hexahedral mesh generation[C]//*Proceedings of the 20th International Meshing Roundtable*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2011: 179-196.
- [216] LU J H C, SONG I, QUADROS W R, et al. Geometric reasoning in sketch-based volumetric decomposition framework for hexahedral meshing[J]. *Engineering with Computers*, 2014, 30(2): 237-252.
- [217] WU H Y, GAO S M, WANG R, et al. Fuzzy clustering based pseudo-swept volume decomposition for hexahedral meshing[J]. *Computer-Aided Design*, 2018, 96: 42-58.
- [218] LIVESU M, MUNTONI A, PUPPO E, et al. Skeleton-driven adaptive hexahedral meshing of tubular shapes[J]. *Computer Graphics Forum*, 2016, 35(7): 237-246.
- [219] PRICE M A, ARMSTRONG C G, SABIN M A. Hexahedral mesh generation by medial surface subdivision: Part I. Solids with convex edges[J]. *International Journal for Numerical Methods in Engineering*, 1995, 38(19): 3335-3359.
- [220] PRICE M A, ARMSTRONG C G. Hexahedral mesh generation by medial surface subdivision: Part 2. Solids with flat and concave edges[J]. *International Journal for Numerical Methods in Engineering*, 1997, 40(1): 111-136.
- [221] QUADROS W R. LayTracks3D: A new approach for meshing general solids using medial axis transform[J]. *Computer-Aided Design*, 2016, 72: 102-117.
- [222] LEI N, ZHENG X P, JIANG J, et al. Quadrilateral and hexahedral mesh generation based on surface foliation theory[J]. *Computer Methods in Applied Mechanics and Engineering*, 2017, 316: 758-781.

编辑 张莉