

基于多重网格尺寸场的非结构自适应网格生成

曹杰¹⁾, 关振群²⁾, 李震波^{1)*}

¹⁾ (南华大学数理学院 衡阳 421001)

²⁾ (大连理工大学工程力学系 大连 116024)

(lizhenbo@usc.edu.cn)

摘要: 针对复杂外形自适应四面体网格生成问题, 以递归细分的规则网格——多重网格作为三维尺寸场构建的背景网格, 提出一种基于多重网格的尺寸场构建算法. 首先从网格密度和几何贴合度可控的几何网格上提取曲率自适应网格尺寸; 然后利用简化的尺寸源将单元尺寸约束信息传递给背景网格, 将规则网格细分递归地应用到尺寸远大于单元尺寸约束的单元格; 最后提出一种高效的双向渐进光滑方法, 限制背景网格上的尺寸场梯度. 采用规则网格、多重网格等网格结构计算了多个工业模型的尺寸分布. 数值算例结果表明, 多重网格在单元形状质量方面具有较好的效果, 使用更少的时间和约 14% 的内存构建了与规则网格几乎相同的渐变尺寸场.

关键词: 网格生成; 自适应网格; 尺寸场; 规则网格

中图分类号: TP391.41 **DOI:** 10.3724/SP.J.1089.2025-00037

Unstructured Adaptive Mesh Generation Based on the Multi-Grid Size Field

Cao Jie¹⁾, Guan Zhenqun²⁾, and Li Zhenbo^{1)*}

¹⁾ (School of Mathematics and Physics, University of South China, Hengyang 421001)

²⁾ (Department of Engineering Mechanics, Dalian University of Technology, Dalian 116024)

Abstract: For the generation of adaptive tetrahedral meshes, a new algorithm is presented to construct a three-dimensional size field taking the recursively subdivided regular grid, multi-grid, as the background grid. First, curvature adaptive mesh sizes are extracted from the geometric mesh controlled by refinement angle and global mesh size. Then, size constraint information is passed to the background grid using simplified size sources, and a regular grid is inserted into those cells where the element size requirement is much smaller than that of the current cell. Finally, a bi-directional progressive smoothing method is proposed to limit the size gradient. Various grid constructions such as regular grid, multi-grid are employed to represent the size distribution of a number of industrial models. The numerical examples show that multi-grid is superior in terms of element shape quality. Compared to regular grid, multi-grid required less time and about 14% memory to construct an almost identical gradient size field.

Key words: mesh generation; adaptive mesh; size field; regular grid

复杂的单元尺寸控制(包括几何自适应、计算自适应和用户自定义等)需要构建大量不同类型的尺寸源指导生成合适的网格, 包括点源、线源、面

源、体源等. 面对愈加精细的计算模型和复杂的网格尺寸控制要求, 构建一个精细的、满足复杂自适应细化条件的尺寸场尤为重要.

判断尺寸场的背景网格是否有效的标准^[1]如下: (1) 简单. 背景网格可在用户合理的时间和精力范围内构建; (2) 准确. 背景网格在整个问题域上可提供平滑可控的尺寸分布; (3) 效率. 从背景网格到问题域, 查询点的尺寸插值计算是高效的; (4) 灵活. 背景网格的拓扑修改和尺寸场更新是便捷的. 背景网格的这些特性对非结构化网格生成器的鲁棒性和可用性有相当大的影响, 比较流行的三维背景网格有非结构网格、八叉树(octree, OCT)和均匀规则网格(uniform regular grid, RG).

非结构背景网格的单元形状质量一般较差, 节点稀疏, 需要求解额外的约束方程限制节点之间的尺寸梯度^[2]. 此外, 查询点网格尺寸的插值依赖于连接关系的存储和超快速的点定位方法, 而且, 狭长、扁平的三角形以及剧烈的单元尺寸变化会引起不理想的尺寸插值结果^[3-4].

将尺寸函数控制下的尺寸场存储在局部自适应细化的 OCT 背景网格上, 可以得到整个问题域上更精细的尺寸分布^[5-8]. 然而, 对于愈益精细的复杂计算模型, 大面积的单元细化显著地增加了维持数据结构所需要的内存以及尺寸场的构造时间, 在尺寸场构建效率和捕捉急剧尺寸变化的准确性之间难以取得平衡.

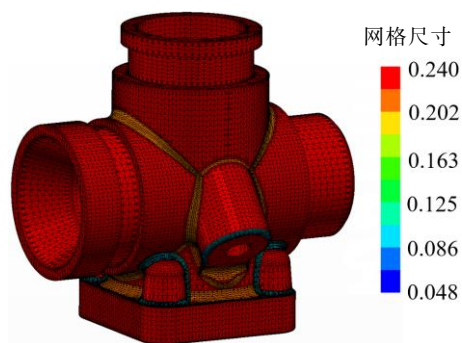
RG 使用具有线性运算复杂度的最简数据结构^[1,9], 并具有强大的尺寸梯度控制能力; 但它的细分级别取决于较小的特征尺寸, 因此对于精细的大型网格, 其内存成本令人望而却步.

本文提出一个基于多重均匀规则网格(multiple uniform regular grid, MG)的尺寸场自动构建算法. 首先基于高可控的几何网格生成分布合适的表面曲率采样点集并得到曲率尺寸函数; 然后通过简化传统尺寸源, 将单元尺寸函数信息

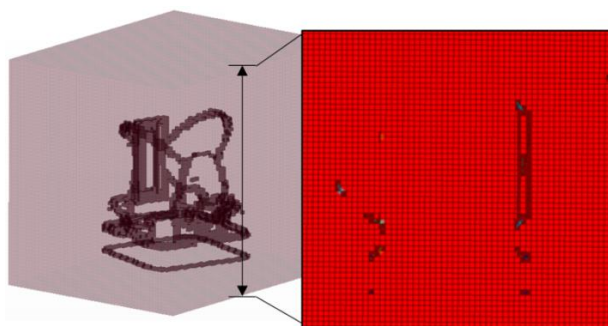
以一种更通用和简洁的方式传递给背景网格; 最后在多重网格上构建三维尺寸场生成自适应网格. MG 是在单元格中递归地插入 RG 的结果, 延续了后者的简单性和强大的尺寸梯度控制能力, 同时在较小特征区域的局部细化能力节省了计算机的内存.

1 基于多重网格的三维尺寸场生成

作为背景网格, RG 的拓扑是简单的. MG 是 RG 的一个扩展, 它是基于空间递归细分对每个单元应用规则网格插入的结果. 图 1 所示是在只细分一次的 MG 上构建尺寸场的基本流程: (1) 生成贴合几何模型的三角形网格, 在网格上计算出曲面曲率, 并根据用户曲率自适应参数得到曲率网格尺寸, 如图 1a 所示; (2) 将几何表面的曲率网格尺寸以节点坐标和网格尺寸的组合形式加载到 RG 上, 并把网格尺寸附在背景网格的节点上, 在网格尺寸较小的节点所在的单元内部插入一个更细的 RG, 更新细网格节点上的网格尺寸, 因此由网格节点的尺寸值在单元内部的线性插值形成一个初始的空间尺寸场, 如图 1b 截面图所示; (3) 光顺网格节点上的网格尺寸, 限制节点与节点之间的尺寸梯度, 由于较小的网格尺寸逐渐向周围扩散, 需要在影响范围内的单元内部也插入细的 RG, 得到光顺后的尺寸场, 如图 1c 截面图所示; (4) 在尺寸场的指导下, 使用非结构网格生成器生成三角形/四面体网格, 如图 1d 所示. 数值算例结果表明, 双层的 MG 已经可以满足尺寸场构建的要求, 接下来的讨论主要围绕双层的 MG 展开.



a. 基于三角形网格计算的曲率网格尺寸



b. 加载曲率网格尺寸后的背景网格和尺寸场

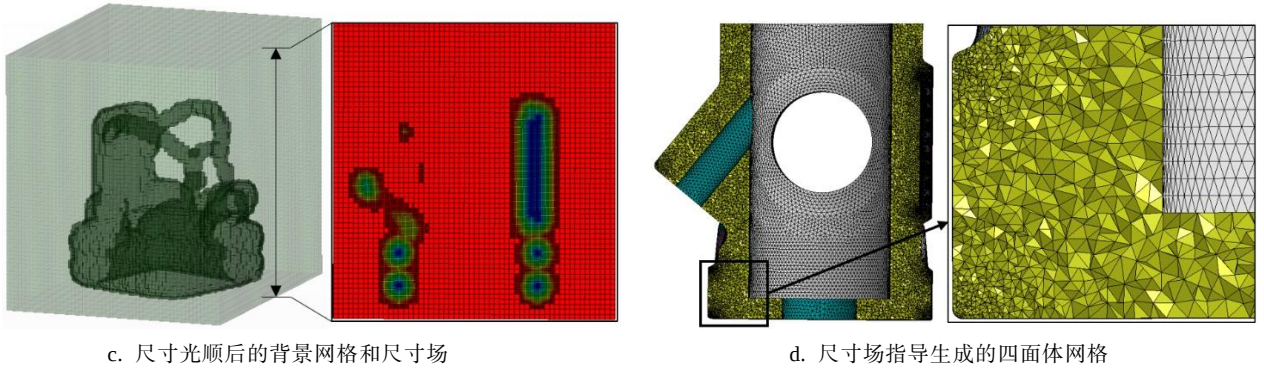


图1 双层的MG上尺寸场构建流程图

1.1 曲率网格尺寸计算

渐变网格的单元尺寸控制有几何自适应、计算自适应和用户自定义3种方式. 本文提出一种基于几何网格的曲率尺寸提取方法, 用于几何曲率自适应表面网格生成.

通常, 曲面网格需要准确地贴合几何曲面, 它与几何曲面之间的角度偏差在给定范围内, 这样的曲面网格称为几何网格^[10-11]. 张角 θ 反映网格边与曲线/曲面的贴近程度, 全局网格尺寸 h_{global} 控制网格边的平均长度. 如图2所示, p 点处的网格尺寸 h 满足

$$\begin{cases} \frac{h}{d_1 + d_2} > \cos \frac{\theta_{\max}}{4} \\ h < h_{\text{global}} \end{cases} \quad (1)$$

其中, 最大张角 $\theta_{\max}$ 和 h_{global} 由用户定义; m 点是边 pq 在参数空间中的中点, 由2个端点在曲线上的参数坐标 t 或曲面上的参数坐标 (u, v) 的二分计算得出, d_1 和 d_2 分别表示边 pm 和 mq 的长度.

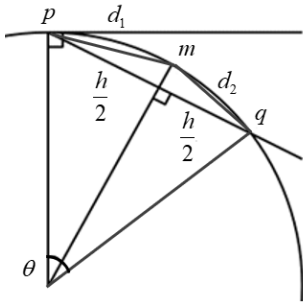


图2 网格尺寸的约束条件

对于各向同性网格生成, 在计算曲面上某点的曲率时, 实际上是计算该点的2个主曲率, 并取其中大的那个作为该点的曲率值. 根据用户设定的 $\theta_{\max}$, 曲面上任意一点的曲率自适应网格尺寸定义为

$$h_c(\mathbf{x}) = 2\rho(\mathbf{x}) \sin \frac{\theta_{\max}}{2},$$

其中, $\rho(\mathbf{x})$ 表示曲面上位置 $\mathbf{x}$ 处的最小曲率半径, 可以调用几何建模器的API得到^[2,12]. 但是, 这类接口通常是按单点查询设计的, 所使用的参数化计算方法较为耗时, 为了保证效率, 可以使用基于几何网格的近似求法^[8,13].

本文首先采用对齐曲线网格划分(aligned curve meshing, ACM)方法^[14]生成CAD模型的几何网格; 然后采用曲率近似求法计算出几何表面的曲率自适应网格尺寸. 如果几何是离散的形式, 如STL三角化模型, 则需要借助单元重心或长边插点等方式先进行网格细化, 再计算曲率自适应网格尺寸. CAD模型作为输入时的步骤如下:

Step1. 曲线离散. 由曲线的2个端点构成的边被视为初始曲线网格. 在参数域对任意区间的边或其子边进行递归二等分, 直到所有子边的物理域长度满足式(1). 将曲线网格投影到曲面上形成曲面的剖分域, 其中, 边界离散边被标记为约束边. 在曲面参数域生成保边界约束的初始Delaunay网格, 如图3a所示.

Step2. 基础曲面离散. 如图3b所示, 由于矩形是参数曲面的基本形状, 因此可以很自然地沿着参数线对其进行离散化. 首先创建一个四边形作为初始面网格; 然后在参数域对网格边进行递归二等分, 类似于曲线离散化, 直到任何一条边的物理域长度都满足式(1). 每次边分裂将产生一条与参数域坐标轴正交的参数线, 依次分裂与参数线相交的边. 生成的网格是参数域中的非均匀的规则网格, 规则网格的边界点与曲线离散点不一致.

Step3. 曲面离散. 将位于剖分域内的基础网格节点Delaunay插入剖分域中, 并恢复基础网格中完全落在剖分域内的边, 将曲线网格与基础曲面网格融合成剖分域内的约束三角形网格, 如图3c所示. 删除位于剖分域外的多余三角形单元, 如图3d所示.

Step4. 曲率网格尺寸近似计算. 采用Rusinkiewicz^[15]描述的张量平均方法计算每个网格节点处的最大曲率, 特征边上的最大曲率取2个曲面中的较

大者. 计算得到图 3e 所示的节点曲率网格尺寸分布.

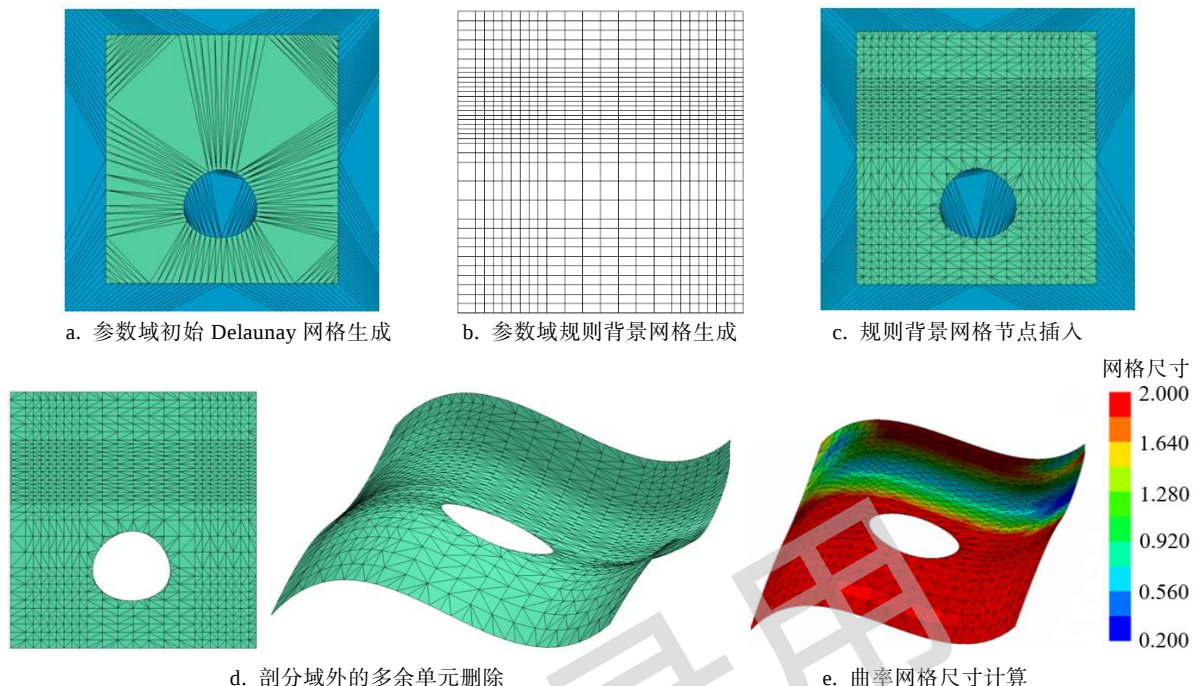


图 3 几何网格上曲率网格尺寸计算流程图

1.2 尺寸源的加载

除了几何曲率自适应网格尺寸, 尺寸控制还包括用户自定义的网格尺寸和来自后验误差分析的自适应细化尺寸. 复杂的几何外形和迭代的仿真分析对网格尺寸的要求是复杂且动态的, 尺寸源的复杂组合通常被用来构建覆盖问题域的光滑尺寸场, 构建尺寸场的背景网格需要足够简单且高效地接收这些尺寸信息. 为了解决这些难点, 本文对尺寸源进行简化, 使得它们以一种更简单的模式将尺寸信息传递给各类型的背景网格.

根据维度和支撑对象, 传统的尺寸源分为点源、线源、面源、体源和函数源^[16-19]. 本文将它们进一步简化为点集、有界盒的集合、实体域等尺寸源模板, 以一种简单、通用的方式将尺寸信息加载到背景网格上, 尺寸源模板的任意组合可以满足空间任意区域网格细化的复杂控制要求.

(1) 点集. 如果点源由几何点和指定的网格尺寸组成, 则在背景网格中找到包含该点的单元格, 并更新附在单元格叶节点上的网格尺寸(叶节点通常指单元格的左下角节点). 尺寸源的载体为任意复杂的几何曲线(面)或曲线(面)网格时, 其效果相当于点源的叠加. 承载表面曲率网格尺寸信息的 ACM 网格是一个典型的网格类型的面源.

(2) 有界盒的集合. 如果离散化的二/三维单元近似各向同性, 则其平行于笛卡儿坐标轴的有界盒可以视为一个广义点源, 与有界盒相交的那

些单元格上节点的网格尺寸将被更新.

(3) 实体域. 作为体源载体的实体几何或网格可以简化为一个位于有界盒内的实体区域. 首先快速地找到与有界盒相交的背景网格单元; 然后构建一个定位函数, 确定背景网格的叶节点(单元的左下角节点)是否位于该区域内. 常见的几何类型的载体包括椭球体、圆锥台、多面体等, 借助局部坐标系可以快速地计算出节点的相对位置. 此外, 对于任意复杂的实体几何或网格, 将它离散成四面体网格, 每个四面体单元即可视为一个广义点源. 如果尺寸源是一个关于空间坐标的连续函数, 同样可以将它简化为一个位于有界盒内的实体区域, 有界盒表示尺寸源的影响范围, 位于有界盒内部的任意背景网格叶节点的网格尺寸由连续函数计算得到.

1.3 尺寸场生成

1.3.1 初始化和细化

将 MG 初始化为单元大小为 h_{global} 的 RG. h_{global} , h_{max} 和 h_{min} 分别表示全局、最大和最小网格尺寸. 初始层在外包围盒对角线上的细分数为

$$N_r^0 = \min(T_0, \lceil D/h_{\text{max}} \rceil).$$

其中, D 表示对角线长度, 默认情况下最大细分数 $T_0 = 64$. 各坐标轴方向上的细分数由 N_r^0 按边长比例计算得到, 每个单元格的长、宽和高相同, 细

分等级为 1.

对于给定的单元格, 如果它满足细化条件, 对单元格进行均匀细分, 细分等级为 2. 不同单元格的细分数相同, 相同细分等级的单元格属于同一重网格. 细分数公式为

$$N_r^1 = \min(T_1, \lceil h_{\text{global}}/h_{\text{min}} \rceil).$$

其中, 最大细分数 T_1 默认为 10. 单元格的细化条件为

$$h_k < \min(\gamma h_{\text{global}}, h_{\text{max}}), \gamma \geq 1.$$

其中, h_k 表示存储在单元叶节点上的网格尺寸; 通过调整系数 γ 可以控制细化区域的面积, 默认为 1. 通常, 在细化阶段细化 1 次足以满足复杂的网格尺寸要求.

对于任意空间点, 可以快速地确定包含该点的单元格的索引坐标 (L^0, L^1) 为

$$\begin{cases} I_x = \left\lfloor \frac{x - x_{\min}}{d} \right\rfloor, I_y = \left\lfloor \frac{y - y_{\min}}{d} \right\rfloor, I_z = \left\lfloor \frac{z - z_{\min}}{d} \right\rfloor \\ L_x^0 = \left\lfloor \frac{I_x}{N_z^1} \right\rfloor, L_y^0 = \left\lfloor \frac{I_y}{N_z^1} \right\rfloor, L_z^0 = \left\lfloor \frac{I_z}{N_z^1} \right\rfloor \\ L_x^1 = (I_x \% N_z^1), L_y^1 = (I_y \% N_z^1), L_z^1 = (I_z \% N_z^1) \end{cases} \quad (2)$$

其中, d 表示单元格的最小边长, $[n]$ 表示 n 的整数部分, $(x_{\min}, y_{\min}, z_{\min})$ 表示背景网格中坐标值最小的节点的坐标. 空间点快速地定位有助于尺寸的快速更新和查询.

如果单元格在尺寸源的辐射范围内, 存储在叶节点上的网格尺寸更新为

$$h_k = \min(h_k, h_s).$$

其中, h_s 表示尺寸源指定的网格尺寸. 当单元格满足细化条件时, 先对它进行细分, 再更新它的子单元格上的网格尺寸. 加载实体域尺寸源时, 由于实体区域的边界是任意复杂的, 计算背景网格与区域相交的单元非常耗时. 为了避免相交计算, 从单元格中提取一组间隔为 $0.5h_{\text{global}}$ 的采样点, 如果存在任意一个点位于区域内, 则该单元格与区域相交.

1.3.2 尺寸光顺

尺寸场中出现大的梯度变化时, 容易生成低质量的网格. 任意方向上的 2 条相邻边在网格中的长度比应小于用户定义的尺寸增长率 $r > 1$, 即长

度比遵循几何级数. 这样的几何级数是通过限制尺寸梯度的 2 阶范数来实现的^[8], 公式为

$$\|\nabla h(\mathbf{x})\|_2 \leq r - 1.$$

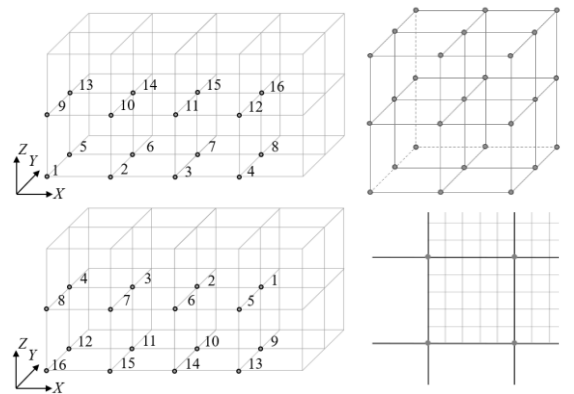
在尺寸场的背景网格上光顺网格尺寸时, 为了提高计算效率, 通常将网格尺寸存储在单元格的叶节点上, 相邻节点上的网格尺寸分别为 h_1 和 h_2 , 如果 $h_2 > h_1$, 则调整 h_2 为

$$h_2 = \min(h_2, h_1 + \Delta x(r - 1)) \quad (3)$$

它类似于 Persson^[20]在梯度限制问题的连续性公式中提出的稳态方程的近似解, 确保对网格尺寸的限制沿值增加的方向上传播. 较小的网格尺寸在限制过程中保持不变, 保留了来自尺寸源的尺寸约束.

由于预期的尺寸梯度只在背景网格单元的边上得到保证, 因此在单元内部可能存在更大的尺寸变化. 为了同时限制背景网格单元内的尺寸梯度, 在背景网格的边上(包括单元的对角线)迭代地调整存储在节点处的网格尺寸的同时, 需要对满足细化条件的单元进行细化, 直到所有节点处的网格尺寸取得最小值, 且不存在需要细化的单元, 则视为计算收敛.

为了提高尺寸光顺的收敛速度, 本文提出一种双向渐进光顺方法. 如图 4a 所示, 节点以索引坐标分别按照 Z, Y, X 值从小到大的顺序进行排序, 再分别按正向和反向的节点顺序调整同层相邻节点上的网格尺寸梯度. 除了上述定义的顺序外, Z, Y, X 值的先后顺序可以任意调整, 同时还可以从大到小进行排序, 因此起始节点可以是背景网格 8 个角点中的任意一个.

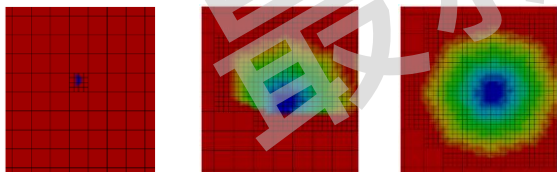


a. 正向和反向的同层节点顺序 b. 同层和邻层关联节点

图 4 网格尺寸光顺时的节点示意图

尺寸光顺时, 如果遍历到节点 i , 则检查该节

点的所有同层相邻节点(如图 4b 第 1 行所示, 最多存在 26 个关联节点), 调整相邻节点上的网格尺寸. 其中, 规则网格中相邻节点的索引可由式(2)的第 2 式和第 3 式快速得到. 光顺过程如下: 首先光顺第 2 层背景网格上的网格尺寸; 然后找到邻层网格的重合节点(如图 4b 第 2 行所示), 使用第 2 层网格节点上的网格尺寸更新与它重合的第 1 层节点上的值; 最后, 再次使用双向渐进光顺方法光顺第 1 层网格节点上的值. 图 5 所示为双层的 MG 上双向渐进光顺后的尺寸场, 网格中附着在所有单元节点上的网格尺寸构成了最终的尺寸场. 这个过程类似于野火传播法, 图 5a 所示为加载点源后的初始尺寸场, 式(3)确保对网格尺寸的限制沿值增加的方向上传播, 由于起点和终点为背景网格外包围盒对角线的 2 个角点, 较小的值的扩散是有序的, 先后以正向和反向的顺序递归传播(如图 5b 和图 5c 所示). 在本文的数值算例结果中, 共 2 次传播即可使得在背景网格中的任意边或单元对角线的 2 个节点的网格尺寸取得最小值. 该方法是有顺序的, 适用于所有拓扑均匀规则的背景网格. 而传统无序迭代光顺的收敛通常需要数十次迭代, 但是无序的迭代光顺可以较为容易地进行并行设计. 对本文方法进行并行设计是未来的一个研究方向.



a. 初始尺寸场 b. 正向逐点光顺 c. 反向逐点光顺

图 5 双层 MG 的尺寸双向渐进光顺效果图

1.3.3 内存消耗

不同背景网格的数据结构占用的 CPU 内存是不同的. OCT 的每个节点至少需要 84 B 内存记录叶节点的坐标, 以及父节点和 8 个子节点的关键信息, 点定位的时间复杂度为 $O(\log n)$. RG 的每个节点只需要 4 B 内存存储节点上的网格尺寸, 点定位的时间复杂度为 $O(1)$. 与 RG 相比, MG 的每个节点还需要存储 1 个网格尺寸数组指针, 便于查询局部细化后的网格节点上的网格尺寸, 点定位的时间复杂度也是 $O(1)$.

2 数值算例与分析

Bawin 等^[8]提出一种基于体积依赖模式的 OCT 背景网格上尺寸场构建方法. 在体积依赖模式下,

单元格上的网格尺寸存储在单元格的中心点, 并以中心点为原点在单元格内形成一个线性尺寸场, 尺寸梯度由相邻单元格网格尺寸的有限差分二阶中心格式近似计算得到, 全局尺寸场由所有单元格上的局部尺寸场组合构成. 尺寸光顺时, 该模式只限制笛卡儿坐标轴方向上的尺寸变化, 而不限单元格对角线方向上的尺寸变化. 本文中, 将网格尺寸存储在背景网格节点上的模式称为节点依赖模式; 将基于体积依赖模式的尺寸场构建方法应用到 MG 上, 得到的背景网格称为 MG_Volume(MG_V). 与 MG 和 MG_V 作为对照的背景网格分别为节点依赖模式下的 RG 和体积依赖模式下的 OCT. 所有测试算例均在 i7-8850H CPU(2.60 GHz)和 16 GB RAM 的单核 PC 上运行.

2.1 不同尺寸场的对比

设置尺寸源的输入为 4 种典型的非均匀分布点源集合, 如图 6 所示, 分别是包含 3 000 个点的螺旋形、椭球面、簇状和圆锥台分布的点集, 位于 $100 \times 100 \times 100$ 的包围盒内; 每个点表示一个边长为 $2R_1$ 的广义点源, 圆锥台的底部和顶部直径分别为 $2R_2$ 和 $3R_2$. 本文以基于 RG 构建的尺寸场为参考尺寸场, 在包围盒内取间隔为 $2h_{\min}$ 的均匀空间采样点阵, 计算各点在不同尺寸场中查询得到的网格尺寸的偏差比, 偏差比公式为 $D_v = \|V_1 - V_0\| / V_0$, 其中, V_1 和 V_0 分别表示从被测试的尺寸场和参考尺寸场中查询到的网格尺寸值. 如果采样点处的网格尺寸值都等于 $h_{\max}$, 则认为它是一个无效采样点. 因此, 由不同偏差比区间的采样点数量的占比给出不同尺寸场的计算精度. 此外, 本文还重点对比了不同尺寸场的尺寸光顺时间和内存消耗, 尺寸光顺通常是尺寸场构建时最耗时的步骤.

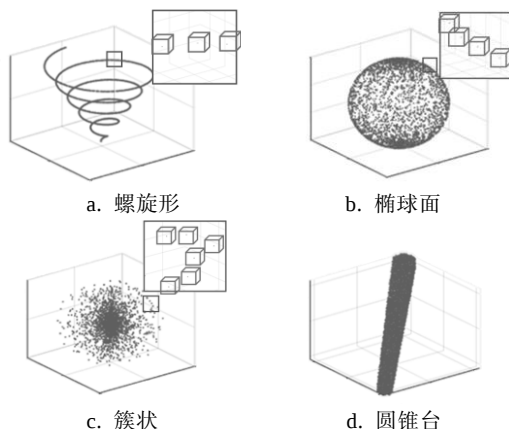


图 6 $100 \times 100 \times 100$ 包围盒内的非均匀分布尺寸源

(1) 均匀细分的背景网格

将背景网格统一细分为单元格大小为 $h_{\min}$ 的 RG. 根据背景网格的特点, 尺寸场的构建分别采用节点依赖模式和体积依赖模式, 尺寸光顺的方法分别采用本文提出的双向渐进光顺方法和传统的无序迭代光顺方法. 其中, OCT 的尺寸光顺采用传统迭代光顺方法; 基本控制参数分别为 $h_{\max}=1.5$, $h_{\text{global}}=1.5$, $h_{\min}=0.3$, $r=1.1$, $R_1=1.6$, $R_2=16.0$.

表 1 所示为各背景网格的内存消耗和尺寸光顺时间, 其中, M1 表示节点依赖模式; M2 表示体积依赖模式; S1 表示双向渐进光顺; S2 表示无序迭代光顺; 背景网格初始化和尺寸源加载的时间相

比尺寸光顺的可忽略不计, 故此处未列出. 可以看出, MG 的总内存消耗只略多于 RG, 单元格的平均内存消耗非常接近, 却只有 OCT 的 4% 左右; MG_V 和 MG_V2 的对比中, 本文提出的双向渐进光顺方法的效率是传统迭代光顺方法的 10 倍左右. 并且背景网格越密, 传统迭代光顺方法达到收敛所需的迭代次数越多, 而本文方法仍然只要迭代 2 次; MG_V2 和 MG 的对比中, 体积依赖模式下的尺寸光顺效率是背景依赖模式下的 4 倍左右, 且与 RG 的基本持平, 而 MG_V2 的尺寸光顺效率是 OCT 的 50 倍左右.

表 1 均匀细分背景网格的结果对比

背景网格	方法	单元数量	内存占用/MB	平均内存占用/B	光顺时间/s	平均光顺时间/s
RG	M1+S1	16.2M	63	3.9	5.9	3.6E-07
MG	M1+S1	17.2M	72	4.2	21.1	1.2E-06
OCT	M2+S2	19.2M	2 097	109.2	271.6	1.4E-05
MG_V	M2+S2	17.2M	72	4.2	48.4	2.8E-06
MG_V2	M2+S1	17.2M	72	4.2	5.2	3.0E-07

图 7 所示为尺寸场在不同偏差比区间的采样点数量的占比. 从图 7a 可以看出, 节点依赖模式下的 MG 的尺寸场偏差最小; MG_V 和 MG_V2 的

对比中, 双向渐进光顺后的尺寸场偏差分布与传统迭代光顺后的基本一致, 说明 MG_V 可以得到一个收敛的结果.

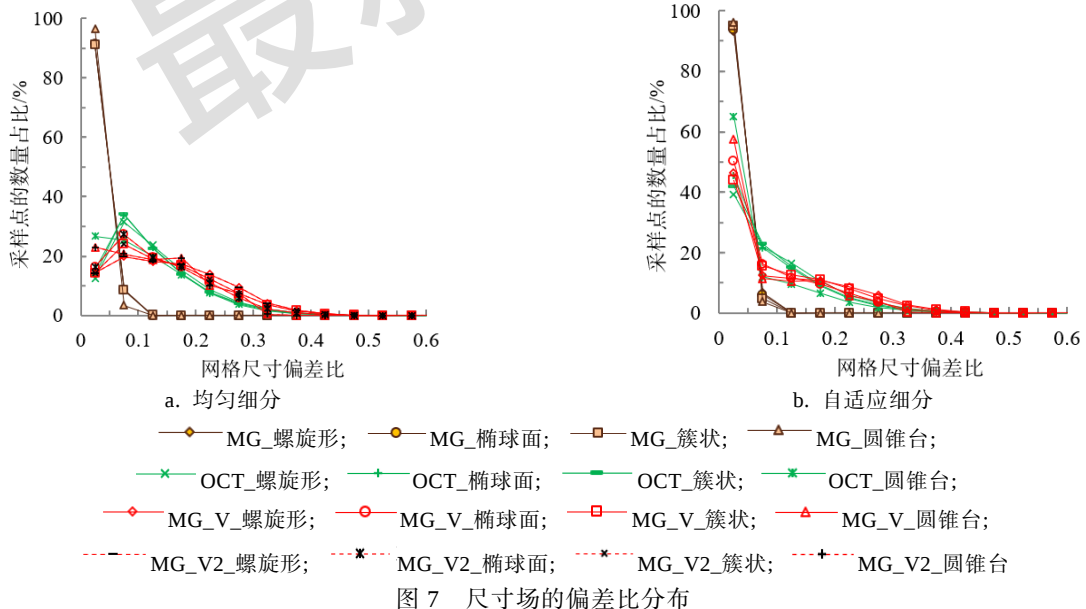


图 7 尺寸场的偏差比分布

(2) 自适应细分的背景网格

当背景网格随网格尺寸进行自适应细分时, 对表 1 中的 RG, MG, OCT 和 MG_V2 背景网格进行对比. 其中, 将 MG_V2 记为 MG_V; 控制参数设为 $h_{\max}=1.5$, $h_{\text{global}}=1.5$, $h_{\min}=0.3$, $r=1.1$, $R_1=1.6$, $R_2=16.0$.

当不同尺寸源输入时, 各背景网格的内存消耗和尺寸光顺时间对比如表 2 所示. 可以看出, MG 的总内存消耗与尺寸光顺时间皆少于 RG, 其中, 内存消耗约占 RG 的 14%; 除螺旋形分布尺寸源外, MG_V 的总内存消耗与尺寸光顺时间显著少于 OCT, 特别是实心圆锥台中只占 OCT 的 10% 左

右,说明加密范围越大, MG 的优势越明显. 此外,尺寸场之间的差异也受到尺寸增长率、最小网格尺寸

等用户参数的影响,这些参数都会影响背景网格的细化范围或等级.

表 2 自适应细分背景网格的结果对比

背景网格	螺旋形		椭球面		簇状		圆锥台	
	内存/MB	光顺时间/s	内存/MB	光顺时间/s	内存/MB	光顺时间/s	内存/MB	光顺时间/s
RG	810	52.8	806	57.6	802	63.2	778	53.6
MG	119	23.8	136	30.9	208	48.5	201	44.0
OCT	140	13.1	580	52.9	687	66.9	1986	247.4
MG_V	97	4.3	113	5.5	188	9.0	170	6.8

从图 7b 可以看出,因为 OCT 和 MG_V 自适应细分后尺寸渐变区域的计算精度下降,所以有效采样点数量增多且主要聚集在 $[0,0.05]$,使得该区间的采样点占比显著变大. 图 8 所示为由采样点集构建的尺寸场在 L 型截面上的分布结果. 可以看出, MG 的尺寸分布与 RG 的几乎相同;体积依赖模式下, MG_V 和 OCT 的尺寸场分布比较接近,在与笛卡儿坐标轴呈 45° 的方向上尺寸梯度过大,因为该模式下只在坐标轴方向上进行尺寸光顺.

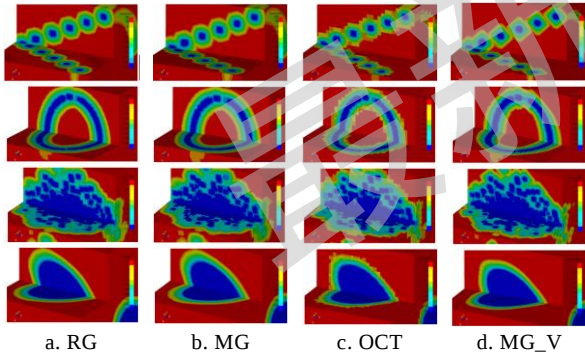


图 8 不同背景网格上构建的尺寸场

2.2 复杂工业模型

尺寸场的基本控制参数为 $\theta_{\max} = 15^\circ$, $r = 1.1$, 构建部分模型的尺寸场时输入了若干自定义的尺寸源. CAD 模型的表面网格由大连理工大学 MeshGene V5 网格生成器中的 Delaunay-AFT 算法生成,四面体网格由 Si^[21]开发的高质量 Delaunay 四面体网格生成器 Tetgen 生成. 基于本文算法生成的尺寸场为确定新节点的位置提供了尺寸信息.

测试模型采用实际的工业模型:

Block 模型. 包含 1 256 个曲面的薄壁结构件,由不同直径的管道相互连接而成;尺寸设置为 $h_{\max} = 6.0$, $h_{\text{global}} = 6.0$, $h_{\min} = 1.2$.

Bike Engine 模型. Honda 运动摩托车的四缸发动机模型共包含 7 992 个复杂曲面;尺寸设置为 $h_{\max} = 6.0$, $h_{\text{global}} = 6.0$, $h_{\min} = 1.2$.

F6 Aircraft 模型. F6 飞机外流场模型包括机身外壳、对称面和远场边界面;尺寸设置为 $h_{\max} = 100.0$, $h_{\text{global}} = 5.0$, $h_{\min} = 0.5$.

(1) 尺寸场

从表 3 可以看出,由于 RG 和 OCT 对大范围的网格细化比较敏感,内存消耗显著大于 MG. 特别地,对于 F6 Aircraft 模型,如果单元格大小取 $h_{\min}$,RG 的高内存消耗会使 16 GB RAM 的 PC 过载;同等条件下, MG_V 的光顺时间也要少于 OCT.

表 3 尺寸场生成的内存占用和时间消耗对比

模型	背景网格	内存/MB	光顺时间/s
Block	RG	685	58.2
	MG	94	40.4
	OCT	105	9.7
	MG_V	79	3.7
Bike Engine	RG	1 173	115.5
	MG	177	39.6
	OCT	1 026	111.1
	MG_V	154	7.9
F6 Aircraft	RG		
	MG	311	55.2
	OCT	172	32.9
	MG_V	129	6.1

(2) 网格质量评估

在 MG 背景网格下,尺寸场指导曲面和实体网格生成器生成的自适应三角形、四面体网格数量、尺寸场生成时间和总时间消耗对比如表 4 所示.

表 4 MG 下网格剖分结果对比

模型	三角形网格数量	四面体网格数量	尺寸场生成时间/s	总时间/s
Block	952 k	11.4 M	28.3	236.5
Bike Engine	1 177 k	23.5 M	80.4	567.5
F6 Aircraft	178 k	18.6 M	56.6	353.2

图 9 所示为 2 个模型的曲率自适应网格. 网格

尺寸以曲率为主导, 在倒角处向区域内部逐渐增长, 从高曲率到低曲率区域生成了光滑过渡的局部自适应网格。

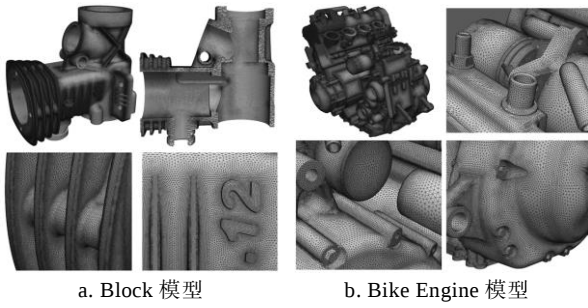


图 9 曲率自适应网格

图 10 所示为不同增长率 r 下生成的表面网格。可以看出, r 越小, 过渡单元的质量越高。

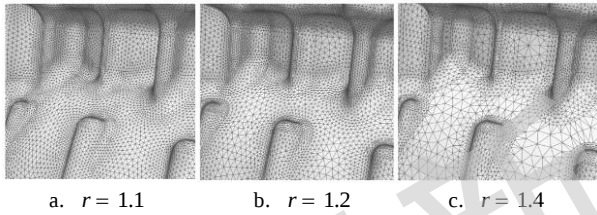


图 10 Bike Engine 模型局部表面网格

在不同 r 和 $\theta_{\max}$ 下, 生成的 F6 Aircraft 模型发动机入口处的表面网格如图 11 所示。可以看出,

随着 r 或 $\theta_{\max}$ 的改变, 网格的疏密变化是合理的。

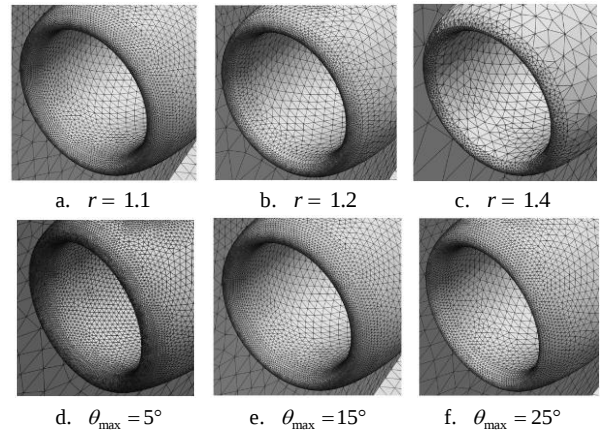
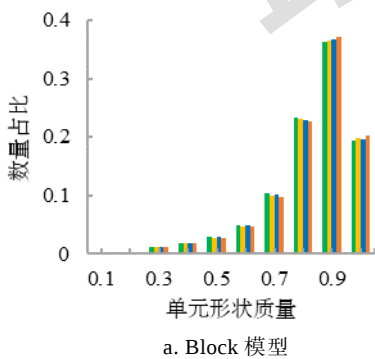
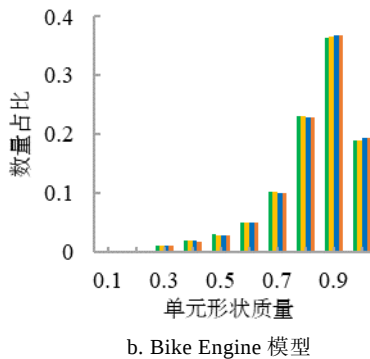


图 11 F6 Aircraft 模型发动机前沿表面网格

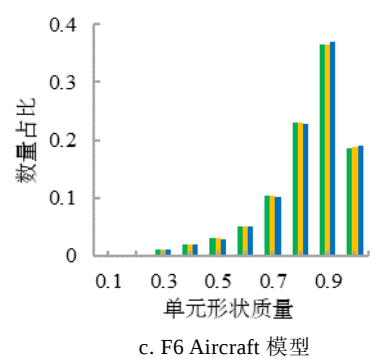
图 12 所示为四面体网格的质量分布, 将单元形状质量定义为: 四面体的内切球半径 r 的 3 倍与其外切球半径 R 的比值, 其值为 1 时, 四面体为正四面体。可以看出, 体积依赖模式下 OCT 和 MG_V 尺寸场指导生成的网格质量略低于节点依赖模式下的 RG 和 MG 尺寸场, 但整体上看, 基于不同尺寸场生成的网格质量分布基本一致, 这是因为尺寸场主要控制网格的疏密, 质量的提升主要依赖一个好的网格优化算法。



a. Block 模型



b. Bike Engine 模型



c. F6 Aircraft 模型

图 12 单元形状质量分布

(3) 网格生成时间对比

网格生成时间对比如图 13 所示。可以看出, Block 模型、Bike Engine 模型和 F6 Aircraft 模型的尺寸场生成总时间占网格生成时间的百分比分别

为 12.0%, 14.1% 和 16.0%, 其中, 尺寸光顺最耗时, 而曲率网格尺寸计算时间与模型的曲面数量有关; 曲面网格和空间网格生成时, 尺寸查询时间的百分比平均为 6.3%。

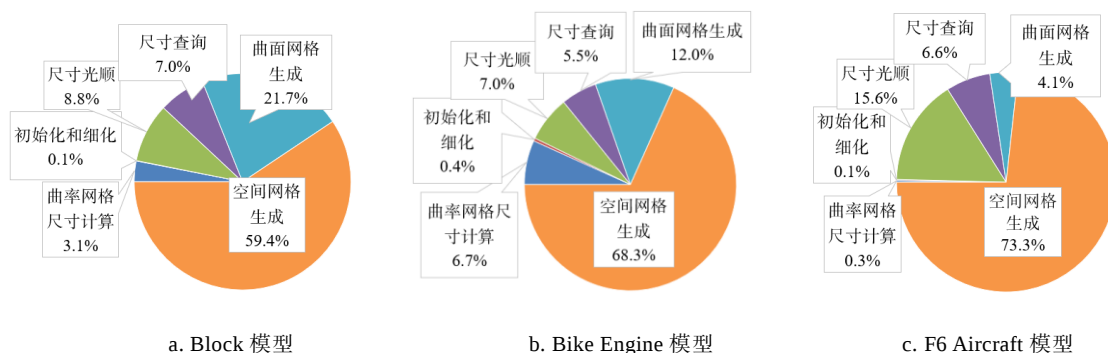


图 13 自适应网格生成时间的百分比

3 结 语

本文提出基于 MG 的尺寸场自动生成算法, 在具有大曲率和几何特征的复杂模型上生成三角形/四面体渐变网格。该算法具有如下优势: (1) 通过构建网格密度和几何贴合度可控的几何表面网格, 实现了对曲面曲率的高效捕捉; (2) 通过分类简化尺寸加密源, 以一种通用的加载方式将单元尺寸控制系统地引入由包括 RG, MG 和 OCT 在内的背景网格构建的尺寸场上; (3) 提出一种双向渐进光滑方法, 加快了渐变尺寸场的构建, 仅需要 2 次迭代即可得到收敛的结果, 与传统无序迭代方法相比, 效率提升 1 个数量级; (4) MG 继承了拓扑结构化、线性计算复杂性和数据结构简单等 RG 的良好特性, 同时具备背景网格局部细化的能力, 在相同尺寸场控制参数下, MG 使用更少的时间和约 14% 的内存构建了与 RG 几乎相同的光滑尺寸场。对于复杂模型的大范围网格细化, 与 OCT 相比, MG 具有明显的优势。

参考文献(References):

- [1] Pirzadeh S. Structured background grids for generation of unstructured grids by advancing-front method[J]. AIAA Journal, 1993, 31(2): 257-265
- [2] Chen J J, Xiao Z F, Zheng Y, *et al.* Automatic sizing functions for unstructured surface mesh generation[J]. International Journal for Numerical Methods in Engineering, 2017, 109(4): 577-608
- [3] Watson D F, Philip G M. A refinement of inverse distance weighted interpolation[J]. Geoprocessing, 1985, 2(4): 315-327
- [4] Owen S J, Saigal S. Surface mesh sizing control[J]. International Journal for Numerical Methods in Engineering, 2000, 47(1-3): 497-511
- [5] Pirzadeh S. Advanced unstructured grid generation for complex aerodynamic applications[J]. AIAA Journal, 2010, 48(5): 904-915
- [6] Deister F, Tremel U, Hassan O, *et al.* Fully automatic and fast mesh size specification for unstructured mesh generation[J]. Engineering with Computers, 2004, 20(3): 237-248
- [7] Quadros W R, Shimada K, Owen S J. Skeleton-based computational method for the generation of a 3D finite element mesh sizing function[J]. Engineering with Computers, 2004, 20(3): 249-264
- [8] 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
- [9] Lo S H, Lee C K. Generation of gradation meshes by the background grid technique[J]. Computers & Structures, 1994, 50(1): 21-32
- [10] Borouchaki 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
- [11] Laug P. Some aspects of parametric surface meshing[J]. Finite Elements in Analysis and Design, 2010, 46(1-2): 216-226
- [12] Kania L A, Pirzadeh S Z. A geometrically-derived background function for automated unstructured mesh generation[C] //Proceedings of the 17th AIAA Computational Fluid Dynamics Conference. Reston: AIAA Press, 2005: 5240
- [13] Tchon K F, Khachan M, Guibault F, *et al.* Three-dimensional anisotropic geometric metrics based on local domain curvature and thickness[J]. Computer-Aided Design, 2005, 37(2): 173-187
- [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] Rusinkiewicz S. Estimating curvatures and their derivatives on triangle meshes[C] //Proceedings of the 2nd International Symposium on 3D Data Processing, Visualization and Transmission. Los Alamitos: IEEE Computer Society Press, 2004: 486-493
- [16] Tremel U, Deister F, Hassan O, *et al.* Automatic unstructured surface mesh generation for complex configurations[J]. International Journal for Numerical Methods in Fluids, 2004, 45(4): 341-364
- [17] Chen Jianjun. Unstructured mesh generation and its parallelization[D]. Hangzhou: Zhejiang University, 2006(in Chinese) (陈建军. 非结构化网格生成及其并行化的若干问题研究[D]. 杭州: 浙江大学, 2006)
- [18] Zheng Y, Weatherill N P, Turner-Smith E A. An interactive

- geometry utility environment for multi-disciplinary computational engineering[J]. International Journal for Numerical Methods in Engineering, 2002, 53(6): 1277-1299
- [19] Weatherill N P, Marchant M J, Hassan O, *et al.* Grid adaptation using a distribution of sources applied to inviscid compressible flow simulations[J]. International Journal for Numerical Methods in Fluids, 1994, 19(9): 739-764
- [20] Persson P O. Mesh size functions for implicit geometries and PDE-based gradient limiting[J]. Engineering with Computers, 2006, 22(2): 95-109
- [21] Si H. TetGen, a Delaunay-based quality tetrahedral mesh generator[J]. ACM Transactions on Mathematical Software, 2015, 41(2): Article No.11

最新录用