

一个高效可靠的三维 AFT 四面体网格生成算法

单菊林 关振群 宋 超

(大连理工大学工程力学系工业装备结构分析国家重点实验室 辽宁 大连 116024)

摘 要 针对三维推进波前算法(AFT-Advancing Front Technique)存在的效率与收敛性问题,文中提出了一整套改进方案,给出了基于拓扑连接的网格数据结构和基于 Hash 表的网格元素的插入、查找、删除算法,提高了整个算法的效率.通过在网格生成过程中动态维护前沿的尺寸信息,提高四面体单元的整体质量.在内核回退求解时通过引入前沿优先因子,改变前沿推进的路径,大大增加了回退求解的成功概率;对于极少数不能回退求解的内核采用基于线性规划的插点方法加以解决,这样就基本保证了整个算法的收敛.在网格生成以后,通过删除不必要的内部节点、合并相关四面体单元以及对所有内部节点进行基于角度的优化,从而进一步有效提高了网格质量.数值算例表明,文中提出的改进算法具有接近线性的时间复杂度,生成网格质量好.该算法已经得到工程应用.

关键词 四面体网格的生成;推进波前法;有限元
中图法分类号 TP391

A Reliable and Effective Tetrahedral Meshing Algorithm

SHAN Ju-Lin GUAN Zhen-Qun SONG Chao

(State Key Laboratory of Structural Analysis for Industrial Equipment, Department of Engineering Mechanics,
Dalian University of Technology, Dalian, Liaoning 116024)

Abstract A reliable and effective tetrahedral meshing algorithm is proposed based on advancing front method. The detailed design of the data structure is given which representing a mesh topology connection. The operators such as insert, query and delete like a database are implemented by using hash table which accelerates the whole algorithm. Instead of preparing a background mesh for mesh spacing control, this information is estimated at the beginning of each layer at each node from the area of connecting triangles on the front and a user-specified stretching factor. A Cell Searcher is prepared to correct the mesh spacing information and to perform geometric search efficiently. During rolling back the advancing path is changed by changing preferential factor of front, as a result the times of rolling back is decreased significantly. Node inserting based on linear programming technique guarantees the convergence of the algorithm. At the end of the mesh generation process, unwanted node removing and angle-based smoothing are employed to enhance the resulting mesh quality. The examples demonstrate that high quality tetrahedral meshes can be generated within a reasonable time limit.

Keywords tetrahedral mesh generation; advancing front technique; finite element

1 引 言

近年来,有限元方法在工程数值分析领域中得

到了非常广泛的应用^[1].应用有限元方法进行数值分析的前提是待分析域的网格剖分.由于实际工程应用中大多数问题是三维实体问题,因此研究建立一个全自动的、健壮的、快速的三维复杂几何实体的

网格剖分算法就显得尤为重要。较之六面体网格,四面体网格由于其优良的几何适应性而在实际中获得了更广泛的应用,因而四面体网格剖分算法也一直是网格生成研究领域的热点之一。目前比较有代表的四面体网格生成方法有 Delaunay 三角化方法^[2]、改进八叉树法^[3]和波前推进法(AFT)^[4-5]。Delaunay 三角化方法首先在剖分域内产生一些点,然后利用著名的 Delaunay 空心球原理把这些点连接成四面体。Delaunay 方法速度快,产生的单元质量较高,其主要问题是边界恢复。改进八叉树法几何适应性强,网格密度可控制,且算法效率较高,该方法的主要问题是边界网格质量差。

三维 AFT 方法生成四面体单元是从待剖分区域的表面三角形单元的集合开始的。这些表面三角形单元构成初始前沿队列。AFT 方法是一个反复迭代,直到整个前沿队列为空的过程,它是一种局部方法,具有如下特点:(1)它在生成节点的同时生成单元,如果能有效地控制生成节点的位置,就可以控制单元的 shape 和尺寸,从而达到单元质量控制、局部加密以及网格过渡等多种要求。这是 AFT 方法的一个突出的优势;(2)效率低是 AFT 方法的一个突出问题,因为它在生成新单元的同时需要进行大量的有效性检查,如体积检查、相交检查、包含检查等。这些有效性检查占用了该算法绝大多数的机时,因此在实施 AFT 方法时,数据结构的设计就尤为重要;(3)收敛性问题是三维 AFT 方法的另一个突出问题,因为 AFT 方法是一种启发式算法,其收敛性没有理论基础的保证,而仅仅依靠算法的收敛策略或开发者的直觉,并且 AFT 算法的失败原因是多种多样的,很难定位。这些困难都使得很多开发者放弃这种方法。

本文主要针对三维 AFT 四面体剖分算法存在的效率与收敛性问题,提出一些改进方案,并给出算法实现的详细步骤。数值算例和大量的实际测试和应用表明,本算法有较好的稳定性并具有接近线性时间的复杂度,能有效地满足实际工程的需要。

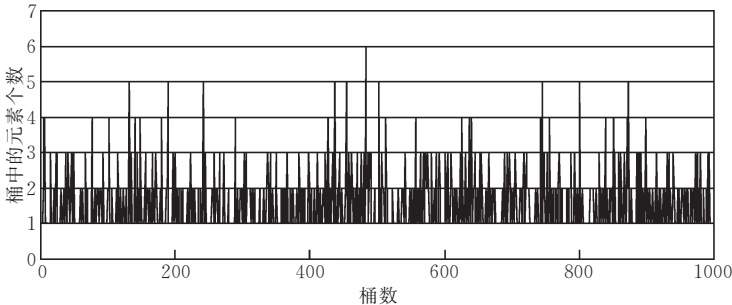
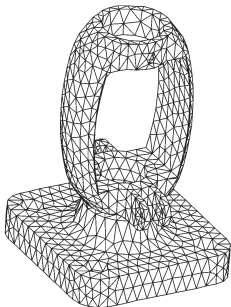


图 2 模型及其单元在 Hash 表桶中的元素个数统计

2 数据结构的设计

如前所述,三维 AFT 方法中,涉及大量的几何相交判断、单元质量判断以及包含判断等,为了提高算法的效率,必须尽量减少这些判断的数量;另外在该算法中,点、线段、三角面等几何元素的增加、查找和修改操作是频繁发生的,为了能根据当前前沿三角形的位置,迅速得到给定区域的所有元素,就需要设计一个满足该算法需要的特定的网格数据结构。一个典型网格数据结构应该包括节点、边、面、体单元这些子结构。Garimella^[6]仔细比较了 10 种不同的网格数据结构,对于不同的应用网格数据结构的设计不尽相同。本文的网格数据结构主要用于四面体单元的生成和优化,综合时间和空间两个方面,本文采用 R4 这种数据结构,如图 1 所示。在网格元素的搜索、插入、删除方面,比较有代表性的是 Bonet 等提出的 ADT^[7]数据结构。理论上 ADT 具有对数平均时间的表现,但这是建立在一个假设上:输入数据有足够的随机性。本文采用开链法完成 Hash 表(Hash table)^[8]来实现相同的功能。使用 Hash 表的关键是要选取一个好的 Hash 函数使冲突最少。对于节点,其 Hash 函数只要返回其 Id 就可以了。而对于边、面以及单元,由于它们是由两个或多个节点构成,其 Hash 值应该是其构成元素 Hash 值的一个组合。实践证明,按式(1)计算组合类型对象的 Hash 值可以大大减少冲突。

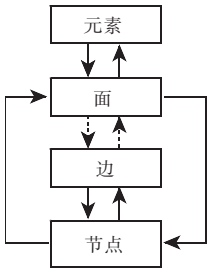


图 1 网格元素间的 R4 连接

$$hash = hash \times 37 + c \quad (1)$$

其中, c 是组合中每个元素的散列值。

图 2 是某机械零件的剖分结果采用式(1)计算得到的前 1000 个桶中单元在 Hash 表中的分布情况统计。可以看出, Hash 表中每个桶中的元素个数都很少, 平均 1.33 个, 少数几个桶中的元素达到 6 个。这样在网格元素的搜索、插入、删除方面具有常数平均时间的表现, 而且这种表现以统计为基础, 不需要依赖输入元素的随机性。

3 四面体网格的生成

与传统的三维四面体 AFT 方法类似, 本文以实体的表面网格作为输入, 生成模型的四面体网格。AFT 方法是一个反复迭代, 直到整个前沿队列为空的过程, 整个算法的整体流程如下:

(1) 生成实体模型的表面三角形网格, 具体过程在作者的文章^[9]中有详细的介绍;

(2) 完成表面三角形网格初始化, 使得表面网格中每个三角形的法线都指向实体的内部;

(3) 根据表面网格初始化前沿队列, 进入以下反复迭代, 直到整个前沿队列为空:

(a) 取出一个前沿作为当前前沿, 计算理想点;

(b) 得到当前前沿附近的一些节点, 按照一定的顺序(例如, 以和当前前沿组成四面体的节点的质量排序)放到节点队列中, 并把一个理想点放到队列的最后;

(c) 在节点队列中找到一个通过相关有效性检查的节点和当前前沿组成一个四面体单元;

(d) 更新前沿队列;

(4) 单元优化, 导出生成结果。

3.1 实体表面三角形网格初始化

在进行实体网格生成之前, 首先需要完成表面网格的初始化:

(1) 使表面网格中每个三角形的法线方向都指向实体的内部。

(2) 封闭性检查, 即所有的表面三角形是否构成一个空间封闭的区域, 如果否, 直接退出, 剖分失败。

3.1.1 初始化背景网格

为了尽可能生成局部均匀一致的四面体单元, 并加快相关前沿数据的搜索速度, 在剖分过程中需要维护前沿三角形的单元尺寸和连接关系信息。这里以节点为中心, 在整个剖分过程中维护的数据和关系与曲面网格生成过程^[9]类似, 主要有

(1) 节点 N_i 的坐标 (x_i) ;

(2) 第 n 层前沿上的节点 (N_i) 参数: 包括参照高度 h_{oi} , 搜索半径 r_{oi} , 所连接的面的集合 F_{oi} 。这里节点的参照高度的计算公式为 $h_{oi} = 1.427 \sqrt{\bar{a}_i}$, 其中 $\bar{a}_i = (\sum_{k=1}^n a_k) / n$, a_k 为当前节点所连接的某个面的面积, n 为连接面的个数。搜索半径 r_{oi} 首先等于该节点所连接的最长边的边长。为了得到变化均匀的网格, 参照高度和搜索半径的计算还要考虑到其周边区域对它的影响。对于节点 N_i , 首先定义一个球 $(x_i, 3r_{oi})$ (即球心在 x_i , 半径为 $3r_{oi}$), 如果剖分域中有节点 N_j 在球内, 那么节点 N_i 和节点 N_j 的搜索半径和参照高度需要调整, 调整方法见文献^[9]中的式(11)。当上述节点参数计算完成以后, 就需要把节点信息的计算结果存到背景网格中去。三维四面体待剖分域背景网格的构造方法和曲面网格生成过程中背景网格的构造方法一样^[9], 这里就不再赘述。

3.1.2 初始化前沿队列

本文采用按层推进的方式完成模型的四面体剖分。然而, 实践表明, 对于大多数模型, 三维 AFT 初始剖分都会遗留一个或多个不能按照三角剖分的要求继续剖分的核。此时, 一般用回退法来解决, 也就是删除内核附近的一层或多层四面体单元, 形成若干个空腔; 然后对这些空腔进行再剖分。经验表明, AFT 方法的收敛性速度和前沿推进的路径密切相关。为了减少回退的次数, 提高整个算法的剖分速度, 对于在前沿队列中的三角形, 本文引入一个优先因子 γ , 默认情况下, γ 等于三角形的面积, 当进入第 i 次回退过程时, γ 值按如下公式计算:

$$\gamma_i = \begin{cases} \alpha, & i \in [0, 4) \\ \alpha\beta, & i \in [4, 8) \\ \beta, & i \in [8, 12) \\ 1.0 - \beta, & i \in [12, 16) \\ (1.0 - \beta)\alpha, & i \in [16, 20) \end{cases} \quad (2)$$

其中, α 是三角形的面积, $\beta \in (0, 1.0]$ 为三角形的质量系数, 可以用三角形的内切圆和外接圆半径之比表示 $\beta = 2r/R$, 对正三角形 $\beta = 1.0$ 。三角前沿队列 Ω 是一个按 γ 从小到大排列的优先队列, 所有未剖分区域边界三角形的集合构成初始前沿队列 Ω_0 。在剖分过程中还存在暂时不能按照三角剖分要求进行推进的前沿三角形, 这些前沿三角形单独放到前沿队列 Ω_H 中。这样本文的前沿队列一共有 3 个: 当前层的前沿队列 Ω_C (初始化完成以后 $\Omega_C = \Omega_0$), 下一层的前沿队列 Ω_N 以及特殊的暂时不可剖分的前沿队

列 Ω_H . 初始化前沿队列包括下面操作:

(1) 把曲面剖分程序得到的按右手规则编号的模型表面三角网格放到前沿对队列 Ω_C 中, 这些前沿三角形的集合形成封闭的待剖分域.

(2) 按照 3.1.1 节介绍的方法, 求得每个节点的参照高度和搜索半径. 把 Ω_C 中所有节点及其参照高度和搜索半径放到背景网格中.

3.2 前沿推进

对当前层的前沿队列 Ω_C 中的每一个前沿向前推进, 其推进算法如下:

1. 计算单元的理想尺寸和理想点.

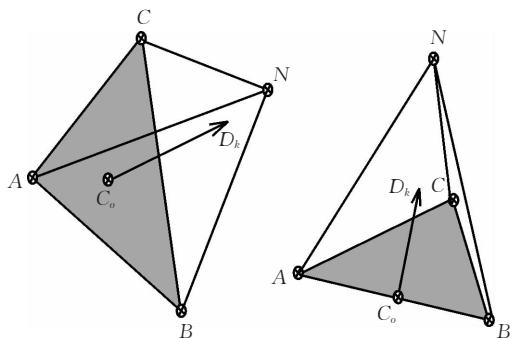
从 Ω_C 中取出一个三角形, 记为 $\triangle ABC$. 那么新生成的单元的理想尺寸为

$$h = \max\left(\frac{h_{oA} + h_{oB} + h_{oC}}{3}, \frac{\sqrt{\alpha_k}}{2}\right) \quad (3)$$

$h_{oA}, h_{oB}, h_{oC}, \alpha_k$ 分别表示三角形 3 个节点的参照高度和面积. 单元理想尺寸确定以后, 四面体的理想点为

$$N_0 = C_0 + gh\mathbf{D}_k \quad (4)$$

\mathbf{D}_k 是 $\triangle ABC$ 的单位法向量, C_0 是计算理想点的基准点. 如图 3 所示, 当 $\triangle ABC$ 的质量系数 $\beta > 0.5$ 时, C_0 是三角形的重心, 否则 C_0 是三角形长边的中点. 系数 g 是一个动态调整的参数, 其初始值是 1.0.



(a) 以三角形中心为基准 (b) 以长边中点为基准

图 3 理想点的确定

2. 形成可见节点集 ψ_s 、可见边集 Φ_s 和可见前沿 Ω_s .

可见节点集 ψ_s 是 $\triangle ABC$ 周围的一系列节点, 计算方法为

2.1. 取得 $\triangle ABC$ 3 个节点的搜索半径 r_{oA}, r_{oB}, r_{oC} ;

2.2. 计算: $r_{3D} = \delta \cdot \max(r_{oA}, r_{oB}, r_{oC})$, δ 是一个经验值, 本文取 1.2;

2.3. 构造球 $S(x_c, r_{3D})$, 其中 $x_c = 0.4C_0 + 0.6N$, C_0 是 $\triangle ABC$ 的重心;

2.4. 计算 Ω_C 中落在球 $S(x_c, r_{3D})$ 内的节点, 这些节点构成 ψ_s .

在步 2.4 中, 为了加快节点搜索的过程, 可以首先构建球 $S(x_c, r_{3D})$ 的外接立方体, 取得落在这个外接立方体的所有节点, 然后判断这些节点是否落在球内. 在 Ω_C 中, 和 ψ_s 中节点连接的所有三角形构成可见前沿 Ω_s , Ω_s 中的所有边形

成可见边集 Φ_s .

3. 形成活动球和活动节点队列 ψ_A .

当可见前沿和可见节点集形成以后, 根据已有点优先的原则, 接下来的工作是形成活动球和活动节点队列. 活动球为 $S(x_c, h)$, 即以 x_c 为圆心, 当前前沿的参照高度 h 为半径的球. 把属于搜索节点集并且在活动球内位于 $\triangle ABC$ 上方的节点, 按照和理想点距离由近及远排序, 形成活动节点队列 ψ_A , 并把理想点 N_0 放入 ψ_A 的最后.

4. 在活动节点队列 ψ_A 中查找合适的点.

依次从 ψ_A 取出一个节点 P 和当前前沿 $\triangle ABC$ 组成四面体单元 T_{ABCP} . 如果 $P \neq N_0$, 进行下列所有检查; 否则, 如果 $P = N_0$ 只需进行 (3~5) 的节点可行性检查即可 (如图 4 所示). 在下列检查中, P_i 表示 ψ_s 中除 A, B, C, P 以外的任何一个节点.

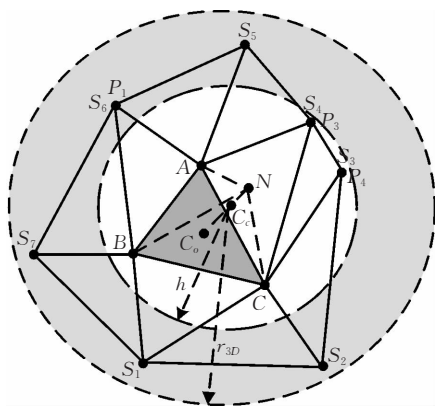


图 4 可行性检查

4.1. 体积检查, 即 $(\mathbf{AB} \times \mathbf{AC}) \cdot \mathbf{AP} > \epsilon$, 即点 P 在 $\triangle ABC$ 的上向, 且 T_{ABCP} 的体积大于一个最小容许值, 本文 $\epsilon = 0.01 \times h \times \alpha_k$, h 是当前前沿的参照高度, α_k 是当前三角形的面积;

4.2. 距离检查, 即点 P 到基准点的距离 C_0 的距离满足: $d_{PC} < d_{\max}$. 默认: $d_{\max} = 2h$; 若 $\triangle ABC$ 的质量系数 $\beta < 0.5$, 且 $\triangle ABC$ 所连接的最长边的边长 $l_{\max} > 2h$, 则 $d_{\max} = l_{\max}$.

4.3. 不包含其它节点检查, 即四面体 T_{ABCP} 中不包含任何一个搜索节点. 也就是说, 对于 ψ_s 中的任一节点 P_i 都要在 T_{ABCP} 的外部.

4.4. 夹角检查. 假如新生成的四面体 T_{ABCP} 的第 4 个点 P 离相邻三角前沿的距离过近, 势必在后续生成四面体的过程中生成质量较差的薄元. 为此需要进行下列边-面、面-边的夹角检查. 本文中把在 Ω_C 中和 $\triangle ABC$ 共享同一节点的三角形称为 $\triangle ABC$ 的相邻三角形.

(a) 边-面夹角检查: 当 $P_i \neq S_i$ 时, 边 AP_i, BP_i, CP_i 和 $\triangle ABC$ 的相邻三角形的夹角 $\alpha_1 \geq \alpha_{\min}$.

(b) 面-边夹角检查: $\triangle ABP_i, \triangle BCP_i, \triangle CAP_i$ 和 $\triangle ABC$ 的相邻三角形中除边 AB, BC, CB 以外的所有边的夹角 $\alpha_2 \geq \alpha_{\min}$.

以上检查中, 首次剖分时 $\alpha_{\min} = 10^\circ$, 当进入第 n 次回退过程时, $\alpha_{\min} = 10^\circ \sqrt{\beta_{\min}}$. β_{\min} 是当前前沿队列中质量系数最小的三角形的质量系数.

4.5. 相交检查, 这也包括两个方面:

(a) 边-面相交检查: 即 AP_i, BP_i, CP_i 和可见前沿 Ω_s 中的任何一个三角形不能相交。

(b) 面-边相交检查: 即 $\triangle ABP_i, \triangle BCP_i, \triangle CAP_i$ 和可见边集 Φ_s 中任何一条边不能相交。

对活动节点队列中的节点 P , 如果以上检查都通过则进入步 8, 否则进入步 5。

5. 在可见节点集 ϕ_s 中查找合适的点。

此时, 对 ϕ_s 中剔除 ϕ_A 后的所有剩余节点按照与理想点的距离由小到大进行排列, 形成新的节点队列 ϕ_L 。依次从 ϕ_L 取出一个节点 P , 进入步 4。若通过步 4 的所有检查进入步 8, 否则进入步 6。

6. 重新计算并判断理想点。

将式(6)中的系数 g 由 1.0 改为 0.7, 0.7² 和 0.7³ 计算得到 3 个新的理想点 N_1, N_2, N_3 。对这 3 个点依次进行步 4 中(3~5)的节点可行性检查。如果有一个点通过以上检查则进入步 8, 否则进入步 7。

7. 把当前前沿放到 Ω_H 中。

此时认为当前前沿 $\triangle ABC$ 在当前条件下不可推进, 需要特别处理。在这里, 把前沿 $\triangle ABC$ 放到 Ω_H 中, 并把它从当前前沿队列 Ω_C 中删除。

8. 更新前沿队列。

将步 4~步 6 中任一步得到的节点设为可行点 P_F 。如果 $P_F = N_i (i=0, 1, \dots, 3)$, 即 P_F 是新生成的理想点, 那么就会新生成前沿 $\triangle ABP_F, \triangle BCP_F, \triangle CAP_F$; 如果 $P_F \neq N_i$, 那么就会新生成上述 3 个前沿三角形中的 2 个或 1 个前沿。把所有新生成的前沿三角形以相应优先因子为序放到下一层前沿队列表 Ω_N 中, 同时把 $\triangle ABC$ 从 Ω_C 中删除。

以上步骤(步 1~步 8)重复执行, 当 Ω_C 为空时, 互换 Ω_C 和 Ω_N , 直到 Ω_C 和 Ω_N 都为空时, 此时初次剖分结束。

3.3 局部网格重新生成

一般来说, 按以上过程完成初始剖分后, 都会在剖分域内留下一些不可剖分的前沿, 也就是说一般 Ω_H 是一个非空集合。经验表明, 在采用 AFT 方法生成有限元网格时, 生成网格质量的高低和成败很大程度上依赖于前沿推进路径的选择。对于初始剖分和回退过程中形成的不可剖分的前沿三角形需要删除后形成新的空腔, 对于这些空腔的剖分就需要尝试不同的推进策略。实践证明, 这种方法对绝大多数情况是快速有效的。

3.3.1 回退

如前所述, 当执行完 3.2 节所有操作步后如果 Ω_H 不为空, 说明还存在尚未剖分的区域, 需要进入回退过程, 即: 删除 Ω_H 中节点所连接的所有四面体单元, 并形成包围这些四面体单元的外空腔三角形面, 这些三角面形成新的剖分域, 这时就可以进入第 1 次回退过程。有时回退需要经过多次才可能完成

剖分, 设当前进行的是第 i 次回退, 按式(2)可得到当前前沿的优先因子 γ_i 。空腔形成的算法如下:

1. 设初始剖分形成的四面体单元集合为 $\Lambda(T)$ 。

2. 提取 Ω_H 中所有节点形成节点表 ϕ_H 。

3. 计算和 ϕ_H 相连的所有四面体单元 $\Lambda(T_H)$ 。

4. 设包围 $\Lambda(T_H)$ 的空腔面为 Ω_{ca} , Ω_{ca} 的形成方法如下

(a) 循环 $\Lambda(T_H)$ 中的每一个四面体单元 $T_{Hi} \in \Lambda(T_H)$ 。
(b) 对于构成 T_{Hi} 的每一个三角面 Δ_{Hi} , 调整其法线方向, 使其指向 T_{Hi} 内部, 如果 $\Delta_{Hi} \notin \Omega_{ca}$, 则 Δ_{Hi} 以 γ_i 为优先序插入 Ω_{ca} ; 如果 $\Delta_{Hi} \in \Omega_{ca}$ 但 $\Delta_{Hi} \notin \Omega_O$ (Ω_O 为初始前沿队列), 则从 Ω_{ca} 中删除 Δ_{Hi} 。

(c) 对于 Ω_H 中的每一个 Δ_{Hj} , 如果 $\Delta_{Hj} \in \Omega_{ca}$ 但 $\Delta_{Hj} \notin \Omega_O$, 则从 Ω_{ca} 中删除 Δ_{Hj} 。

Ω_{ca} 中所有三角形的几何构成了一个或多个封闭的剖分域。这时, 清空 Ω_H , 并按照 3.1.1 节进行数据更新, 使得 $\Omega_C = \Omega_{ca}$, 调用 3.2 节的剖分过程进行剖分。值得注意的是, 上述回退算法并不能保障三维 AFT 的收敛性, 亦即经过若干次回退(本文设定最多经过 20 次)后仍有可能存在不可剖分的多面体, 这就需要进入内核多面体的剖分过程。

3.3.2 内核多面体的剖分

本文研究发现, 对于给定的一类简单形式的内核多面体, 通过建立一个简约的数学模型并采用线性规划方法求解该模型能够精确可靠地计算出一个 Steiner 节点的位置, 该节点能够有效地与内核多面体的所有三角面片连接生成合法四面体单元, 进而实现该多面体的四面体化。设内核多面体的三角形表面外法线方向指向多面体外部, 欲保证作为新的插入节点的 P 点, 就要使 P 点与每一个表面 $\triangle V_i V_j V_k$ 之间满足 $\mathbf{V}_i \mathbf{P} \cdot \mathbf{n}_i < 0$, 其中 \mathbf{n}_i 为 $\triangle V_i V_j V_k$ 的外法线方向向量。设 V_i 点坐标为 $V_i(x_v, y_v, z_v)$, \mathbf{n}_i 向量 3 个分量为 (n_x, n_y, n_z) , P 点坐标为 (x, y, z) , 则上述不等式转化为

$$(x - x_v, y - y_v, z - z_v) \cdot (n_x, n_y, n_z) < 0,$$

即

$$xn_x + yn_y + zn_z < x_v n_x + y_v n_y + z_v n_z \quad (5a)$$

对于式(5)中提出的不等式, 可以归纳为形式 $\mathbf{AX} < \mathbf{b}$, 其中

$$\mathbf{A} = \begin{bmatrix} n_{1x} & n_{1y} & n_{1z} \\ n_{2x} & n_{2y} & n_{2z} \\ \vdots & \vdots & \vdots \\ n_{8x} & n_{8y} & n_{8z} \end{bmatrix}, \quad \mathbf{X} = \begin{bmatrix} x \\ y \\ z \end{bmatrix},$$

$$\mathbf{b} = \begin{bmatrix} x_{1v}n_{1x} + y_{1v}n_{1y} + z_{1v}n_{1z} \\ x_{2v}n_{2x} + y_{2v}n_{2y} + z_{2v}n_{2z} \\ \vdots \\ x_{8v}n_{8x} + y_{8v}n_{8y} + z_{8v}n_{8z} \end{bmatrix}.$$

若再构造一个目标函数 f , 则可将不等式求解问题转化为线性规划问题, 此线性规划问题的构造思路如下:

对于不等式约束 $\mathbf{AX} < \mathbf{b}$, 通过引入一个松弛变量 \mathbf{S} 对其进行改造, 即

$$\mathbf{AX} + \mathbf{S} < \mathbf{b} \quad (5b)$$

则线性规划问题的完整提法为

$$\begin{cases} \min & f = \mathbf{S} \\ \text{s. t.} & \mathbf{AX} + \mathbf{S} < \mathbf{b} \\ & \mathbf{S} \geq \mathbf{0} \end{cases} \quad (5c)$$

此线性规划问题与原来的不等式方程组是等价的. 解此线性规划问题就得到 Steiner 节点的位置.

3.4 自适应四面体网格的生成

采用控制度量(metric tensor)来控制单元的尺寸和方向是自适应网格生成的通用方法. 在三维空间中某一点的控制度量是一个 3×3 实对称阵^[9], 在考虑控制度量的情况下线段 $AB = (A + t\mathbf{AB})_{0 \leq t \leq 1}$ 的长度计算公式为

$$l(AB, M) = \int_0^1 \sqrt{\mathbf{AB}^T M(A + t\mathbf{AB}) \mathbf{AB}} dt \quad (6)$$

这里 $M(A + t\mathbf{AB})$ 是点 $A + t\mathbf{AB}$ 处的控制度量. 网格自适应的目标就是按公式计算得到的每条边的长度为 1. 这里首先根据用户自适应的要求生成待剖分域的控制度量, 可以在四面体内部、三角面或边上插入新节点的方法来达到自适应目的. 经验表明, 在边上插入节点的方法实现简单而且效果较好, 具体实现方法简述如下:

1. 遍历四面体网格中所有的边, 如果对于某条边, 按照式(6)计算得到的长度 $l > \sqrt{2}$, 则待分裂的边的集合 L 中.
2. 对 $L = \{l_i, i=1, 2, \dots, n\}$ 中的边按长度进行排序, 使得 $l_1 \geq l_2 \geq \dots \geq l_n$.
3. 依次遍历 L 中的所有边, 对于当前边 l_i , 对分 and l_i 连接的所有四面体单元, 对于新产生的边, 如果边长 $l > \sqrt{2}$, 放到集合 N 中. 如果集合 N 的最长的边大于 l_i , 那么这里需要首先分裂这条最长边, 然后继续分裂 L 的其它边.
4. 清空 L , 并使得 $L = N$.
5. 重复步 2~步 4 的步骤, 直到 L 和 N 都为空, 自适应过程结束.

4 四面体网格的优化

在四面体单元生成的过程中, 虽然每生成一个单元都力图使之质量最好, 但其最终结果中还是会有存在单元质量很差的单元, 四面体的优化包括拓扑连接和节点位置优化两个方面. 对于拓扑连接的优

化这里主要采用 Shewchuk JR^① 给出的边删除和面删除的方法. 除了连接关系优化外, 本文对生成的四面体还进行以下优化.

4.1 节点删除

如图 5(a) 所示, 如果一个实体内部的节点 P 连接 4 个四面体, 那么这 4 个四面体可以合并为 1 个四面体 T_{ABCD} , 从而删除节点 P . 另外一种节点可以删除的情况如图 5(b) 所示, 如果一个位于实体内部的节点 P 连接 6 个四面体, 那么这 6 个四面体可以合并为 2 个四面体 (T_{ABDE}, T_{BCDE} 或者 T_{ABCE}, T_{ACDE} , 取这两种合并中 2 个四面体质量系数最小的最大者优先).

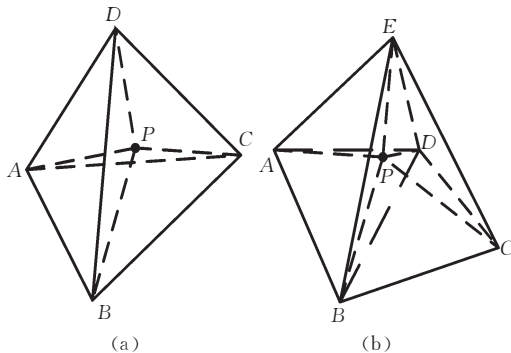


图 5 节点删除

4.2 基于角度的优化

在网格优化中, 拉普拉斯优化(Laplacian smoothing)是经常使用的一种优化方法. 这种方法的突出优点是速度快, 然而该方法并不能保证单元的合法性, 亦即有时会产生负体积的单元. Zhou^[10] 等在不损失计算效率的前提下提出了一种基于角度的二维单元的优化方法, Ito 等^[11] 把这种方法扩展应用到三维四面体的优化中, 现简述如下:

设实体内部的节点 N_i 的坐标为 x_i 和 N_i 相连的边的个数是 n_{ei} . 取和节点 N_i 相连的节点 N_j , 则节点 N_i, N_j 构成边 E_{ij} . 设 E_{ij} 所连接的四面体集合为 T_{ik} ($k=1, 2, \dots, n_{ij}$), f_{ik} ($k=1, 2, \dots, n_{ij}$) 是 T_{ik} 中节点 N_i 对面的集合, 即图 6 中的阴影部分. 那么 f_{ik} ($k=1, 2, \dots, n_{ij}$) 的平均单位法向量 (\mathbf{N}_{ij}) 及面积和 (a_{ij}) 为

$$\mathbf{N}_{ij} = \frac{\sum_{k=1}^{n_{ij}} \mathbf{N}_{f_{ik}}}{\left| \sum_{k=1}^{n_{ij}} \mathbf{N}_{f_{ik}} \right|}, \quad a_{ij} = \sum_{k=1}^{n_{ij}} a_{f_{ik}} \quad (7a)$$

① Shewchuk J R. Two discrete optimization algorithms for the topological improvement of tetrahedral meshes. <http://www.cs.berkeley.edu/~jrs/papers/edge.ps> (最后一次访问时间: 2007 年 5 月 17 日)

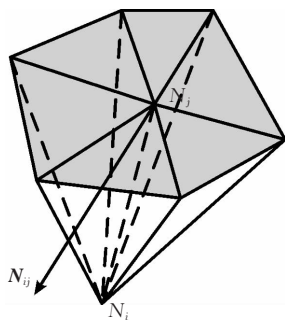


图 6 基于角度的优化

如果单独考虑节点 N_j , 新节点的坐标为: $x'_i = x_j + |x_j - x_i| N_{ij}$, 其中 $|x_j - x_i|$ 表示 E_{ij} 的长度. 因为一共有 n_{ei} 边和 N_i 相连, 以面积和为权求取优化后新节点的位置(x_{inew}):

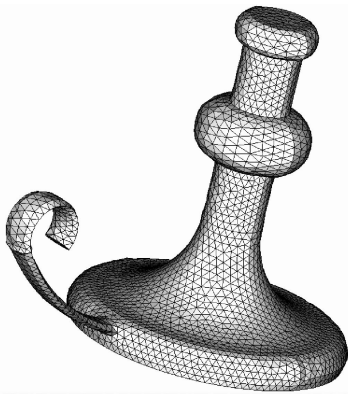
$$x_{inew} = \frac{\sum_{j=1}^{n_{ei}} a_{ij} (x_j + |x_j - x_i| N_{ij})}{\sum_{j=1}^{n_{ei}} a_{ij}} \quad (7b)$$

实践证明按上述方法得到的新节点可能会产生负体积的单元, 一旦出现这种情况, 该节点就恢复到原来的位置而不进行上述优化. 否则继续节点集合的下一个内部节点的优化, 直到所有内部节点优

化完毕.

5 数值算例

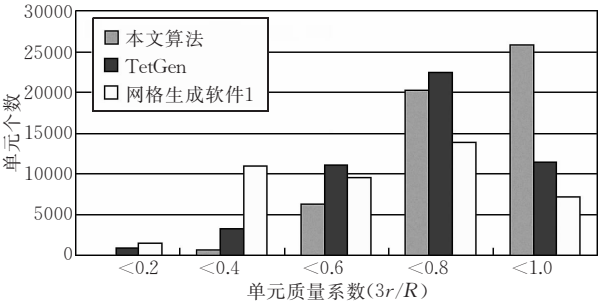
图 7(a)是一个实体模型的三维四面体剖分算例, 图 7(b)是该模型的一个剖视图. 表 1 是在输入同样表面网格的情况下, 本文算法生成的网格和 TetGen^① 及另外一个商业网格生成软件的质量系数分布对比, 图 7(c)是和该表对应的质量系数分布柱形图. 这里四面体质量系数: $\beta_T = 3r/R$, 即为内切圆和外接圆半径比值的 3 倍. $\beta_T \in (0, 1]$, β_T 越大说明该四面体质量越好. 由上面的统计可以看出本文算法生成的网格对 $\beta_T < 0.2$ 的个数是最少的, 占 0.085%, 而 TetGen 和网格生成软件 1 分别占 1.6% 和 3.5%; 本文算法单元质量 $\beta_T > 0.6$ 的数目为 86.95%, 而 TetGen 和网格生成软件 1 分别占 69.1% 和 69.9%. 对于单元的生成速度, 是很难统计的, 原因有两个: (1) 商业 CAE 软件网格生成的方法往往是 Delaunay、AFT 等方法的结合, 而 Delaunay 由于不需要进行像 AFT 方法如此多的几何有效性判断, 所以 Delaunay 算法本身比 AFT 方



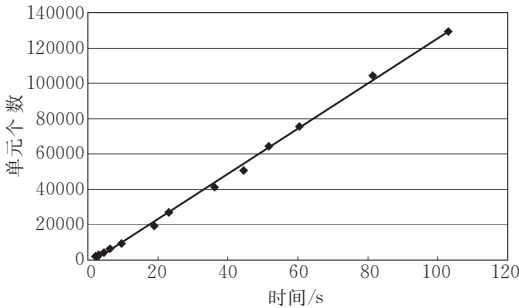
(a) 数值算例 1



(b) 算例 1 的剖视图



(c) 算例 1 的单元质量系数分布图



(d) 算例 1 的单元个数和时间的关系曲线

图 7 数值算例 1

法快; (2) 商业 CAE 软件往往是几何模型预处理、曲面网格、四面体网格生成同时进行, 很难获取生成四面体网格究竟所需的时间. 图 7(d) 是本文算法单

① Si Hang. TetGen, A quality tetrahedral mesh generator and three-dimensional delaunay triangulator. <http://tetgen.berlios.de/> (最后一次访问时间: 2007 年 5 月 17 日)

元生成个数和所花费时间的关系曲线,生成单元的整个过程主要包括单元生成、单元的优化两个方面,计算环境是 1.86GHz 主频,512MB 内存的 PC 计算机,曲线表明本算法的时间复杂度基本是线性的.经过对多个实际使用的机械零件的剖分统计,本文给出的算法生成单元的平均速度是 6 万单元/分钟,显

然这已经能满足实际工程的需要了.

图 8 给出的算例是一个复杂的玩具狗模型的自适应四面体剖分.这个模型在上述两个网格生成软件中都失败了.图 8(b),(c)是这个模型的剖视图,图 8(d)是剖分结果单元质量系数统计.

表 1 算例 1 生成的网格质量系数分布对比

算法	单元个数					β_T			总单元个数
	$\beta_T < 0.2$	$\beta_T < 0.4$	$\beta_T < 0.6$	$\beta_T < 0.8$	$\beta_T < 1.0$	最小值	最大值	平均	
TetGen	801	3194	11069	22427	11393	0.0004	0.999	0.867	48884
网格生成软件 1	1491	10911	9520	13982	7103	0.0131	0.999	0.509	42907
本文算法	45	602	6258	20230	25788	0.0920	0.999	0.785	52923

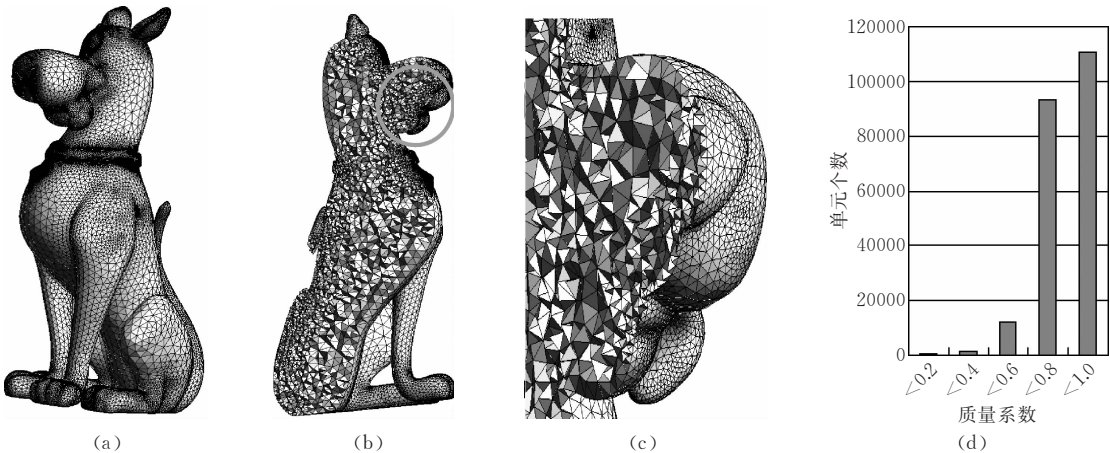


图 8 数值算例 2

6 结 论

本文给出了采用波前推进方法(AFT)实现三维四面体剖分算法的详细过程,包括数据结构的设计、背景网格的构造、前沿推进的过程以及剖分后单元的优化等.采用 AFT 进行四面体剖分的关键是内核的剖分,本文对内核剖分主要采用回退的方法,回退过程中通过设定不同的前沿优先因子来提高成功回退的可能性,对于少数几个不能用回退完成剖分的单元采用基于线性规划的在内核内进行强制插点的方法加以解决.在实体四面体单元剖分完成以后再进行单元的优化从而提高网格的质量.算例与工程实践表明,本文给出的三维四面体 AFT 算法具有很好的时间特性和健壮性,生成网格质量较好,能有效地满足工程实际的需要.

参 考 文 献

[1] Zienkiewicz OC. Achievements and some unsolved problems of the finite element method. *International Journal for Numerical Methods in Engineering*, 2000, 47(1-3): 9-28

[2] Du Qiang, Wang De-Sheng. Tetrahedral mesh generation and optimization based on centroidal voronoi tessellations. *International Journal for Numerical Methods in Engineering*, 2003, 56(9): 1355-1373

[3] Shephard MS, Georges MK. Automatic three-dimensional mesh generation by the finite Octree technique. *International Journal for Numerical Methods in Engineering*, 1991, 32(4): 709-749

[4] Rassineux A. Generation and optimization of tetrahedral meshes by advancing front technique. *International Journal for Numerical Methods in Engineering*, 1998, 41(4): 647-651

[5] Yamakawa Soji, Shimada Kenji. Anisotropic tetrahedral meshing via bubble packing and advancing front. *International Journal for Numerical Methods in Engineering*, 2003, 57(13): 1923-1942

[6] Cha C T, Garimella Rao V. Mesh data structure selection for mesh generation and FEA applications. *International Journal for Numerical Methods in Engineering*, 2002, 55(4): 451-478

[7] Bonet J, Peraire J. An alternating digital tree (ADT) algorithm for 3D geometric searching and intersection problems. *International Journal for Numerical Methods in Engineering*, 1991, 31(1): 1-17

[8] Hou Jie. *The Annotated STL Sources (using SGI STL)*. First Edition. Wuhang: Huazhong University of Science and

Technology (HUST) Press, 2002: 247-282(in Chinese)
(侯捷. STL 源码剖析. 武汉: 华中科技大学出版社, 2002: 247-282)

[9] Guan Zhen-Qun, Shan Ju-Lin, Gu Yuan-Xian. Surface mesh generation based on riemannian metric. Chinese Journal of Computers, 2006, 29(10): 1423-1433(in Chinese)
(关振群, 单菊林, 顾元宪. 基于黎曼度量的复杂参数曲面有限元网格生成方法. 计算机学报, 2006, 29(10): 1423-1433)

[10] Zhou T, Shimada K. An angle-based approach to two-dimensional mesh smoothing//Proceedings of the 9th International Meshing Roundtable. New Orleans, Louisiana, USA, 2000, 373-384

[11] Ito Y, Nakahashi K. Improvements in the reliability and quality of unstructured hybrid mesh generation. International Journal for Numerical Methods in Fluids, 2004, 45(1): 79-108



SHAN Ju-Lin, born in 1976, Ph.D.. His research interests focus on finite element generation.

GUAN Zhen-Qun, born in 1965, professor. His research interests include finite element generation, visualization in scientific computing, CAD/CAE.

SONG Chao, born in 1976, Ph.D. . His research interests focus on finite element generation.

Background

The automatic generation of unstructured meshes plays an important role in computational mechanics especially in the framework of the finite element analysis (FEA). For three-dimensional mesh generation, some useful methods have been developed, such as the advancing front technique (AFT), Delaunay triangulation and spatial decomposition or Octree techniques. It is noted that most of these methods have focused on tetrahedral elements since tetrahedral elements are easily adapted for discretization of a complex, three-dimensional region.

Over the years, the advancing front technique has achieved considerable success in two and three-dimensional mesh generation . In this approach, elements and nodes are generated simultaneously starting from the boundary of the domain and proceeding into the interior. The boundary dis-

cretization is taken as the initial front, and a set of heuristics determines where a new element is to be created, where to place a new interior node, and how to reconnect the front. The algorithm terminates when the front is empty.

The main objective of this study is to develop an automatic tetrahedral mesh generator for FEA. However, the performance of the conventional advancing technique in three dimensions is not always effective and stable. In this paper, topology based meshing data structures are implemented. A node inserting algorithm is proposed based on linear programming technique to solve the Schoenhardt polyhedron problem which is the main barrier of AFT in three-dimensional tetrahedral meshing. The algorithm of adaptive tetrahedral generation and the optimization of tetrahedral meshing are also given at the end of this paper.