

边界一致的 Delaunay 四面体网格稳定生成算法*

赵建军 王启付

(华中科技大学机械科学与工程学院 武汉 430074)

摘要: 提出了一种稳定的四面体网格生成算法, 克服了现有三维 Delaunay 四面体生成算法的如下不足: 针对现有算法的稳定性问题, 提出了随机扰动以及延迟插入的增量方法, 在不牺牲速度的条件下, 提高了稳定性; 针对边界一致问题, 改进了约束边和约束面恢复算法, 使之能够处理曲面边界; 针对质量问题, 提出了交换和插点相结合的网格优化及 Sliver 单元的去掉方法。并给出了几个例子检验提出的算法。

关键词: 网格生成 四面体网格 Delaunay 边界一致

中图分类号: TP391

0 前言

自动生成三维四面体网格在许多学科领域有重要应用, 如有限元分析、计算机图形学、计算机视觉、机器人和生物医学等。在各种四面体网格生成算法当中, 基于 Delaunay 的方法(以下简称 DT 法)受到了人们的充分重视^[1~4]。尽管用 DT 法生成的四面体网格质量不一定最优, 但它较之其他方法具有明显的优势, 这种优势主要表现在: ① Delaunay 四面体网格具有一些最优特性^[5]。② 无论输入模型如何复杂, 容易从理论上保证算法收敛。

然而, 三维 DT 法的构造算法却遇到了一些问题。这些问题可概括为以下三个方面。

第一是稳定性问题, 这也是几何算法的共性问题, 尽管一些精确计算的方法有助于解决此问题, 但计算开销都非常大^[1]。而原始的构造方法^[6,7]又很容易导致数值错误。有必要寻找计算开销小、稳定性也高的算法。

第二是边界一致问题, 一个有效的三角化应该与输入的几何模型边界一致, 然而, DT 算法只考虑了点的连接法则, 并不考虑几何模型边界面和边界线的拓扑约束, 如果给定的边界刚好满足三维 DT 中的边和面的存在性条件, 则 DT 网格能够包含边界边及边界面, 不幸的是, 这种巧合几乎为 0。边界一致问题长期没得到很好的解决。

最后一个是质量问题, 三维 DT 不存在二维 DT 的最大化最小角这一最优质量特性。特别是三维 DT 的每一种算法中, 都不能避免著名的薄四面体(Sliver 四面体)的存在。Sliver 四面体是体积几乎为

0 的单元, 其 4 个顶点几乎共圆。对于有限元分析等应用, Sliver 四面体引起了很多困难。

针对以上问题, 设计了一个稳定的边界一致的三维 DT 算法, 该算法提出了以下的具体方法。

(1) 针对 Delaunay 四面体算法插点过程中的稳定性问题, 采取了点的随机扰动, 延期插入等措施; 为实现稳定的边界恢复, 使用了最少的几何原语。

(2) 利用简单的插点算法实现边界一致。已知的各种实现边界一致方法都要经过求交操作运算^[8,9], 所提出的方法不需要几何求交运算, 只利用以下二个几何原语, 即可实现边界一致: ① 判断给定点是否在四面体的外接球内、外, 还是在球上。② 判断给定点在一个平面之上、平面之中、还是平面之下。在经过点的随机扰动后, 容易稳定实现这二个几何原语。

(3) 使用交换与插入相结合的 Sliver 四面体去除方法。Sliver 四面体的去除一直是一个难题, 我们的策略是根据 Sliver 四面体与周围单元的不同组合形式, 采用交换或插点等不同的方法。

1 定义

先规定三维 DT 的输入, 与通常 CAD 几何模型不同的是, 所提出的输入模型可以是非流形几何。

定义 1(输入模型) 输入模型 Ω 由 4 元组 $\{V, E, S, R\}$ 构成, $V = \{v_i\}$ 代表点的集合, $E = \{e_j\}$ 代表边的集合, $S = \{s_m\}$ 代表曲面的集合, $R = \{r_k\}$ 代表空实体的集合, r_k 为空实体中的种子点。

上述定义中定义了 4 类拓扑实体, 即点、线、面和不通孔。不通孔用种子点表示, 因此实际上只给出了零维、一维和二维三类拓扑实体。这里要特别解释一下不通孔的表示。由于 DT 法生成的是 Ω

* 国家自然科学基金资助项目(No.50275060)。20030620 收到初稿, 20031230 收到修改稿

的凸包, 外边界以外的单元可以通过边界匹配后删除, 对于实体中的不通孔, 在非流形几何中, 既可以为实, 也可以为虚, 以不通孔中的一个种子点表示不通孔应为空。

可以用 $\Delta = (P, F)$ 表示一个四面体网格, 其中, P 为网格的节点集, F 为网格的三角面片集。也可将 Δ 看作四面体的集合。 F 中任取一个面片 f , 除边界面外, f 都和二个四面体相关联, 若其中一个四面体的外接球不包含另一四面体的相对顶点, 称面 f 为局部最优。

输入模型 Ω 中包含 E, S 等约束边界, 而 DT 法只是对点集凸包的四面体化, E, S 中的元素可能没有出现在 DT 中, 称从点集凸包的 DT 到出现 E, S 约束边界的过程为边界一致, 边界一致的过程实际上是约束满足的过程, 边界一致可定义如下。

定义 2(边界一致) 如果输入模型 $\Omega = \{V, E, S, R\}$ 生成的一个四面体网格 Δ 满足如下条件, 称它与输入模型 Ω 边界一致。

- (1) V 中所有顶点均在 Δ 中。
- (2) E 中每一条边在 Δ 中有一系列的边逼近它。
- (3) S 中每一个面在 Δ 中有一系列面片逼近它。
- (4) R 中的种子点构成的“洞”已去除。

达到边界一致的过程也是逐步恢复 Ω 中边界集 E 和 S 的过程, 因此边界一致也称为边界恢复。

通过边界一致, 即可将 Δ 中的拓扑实体进行分类, 若 Δ 中的实体是 Ω 的一部分, 称该实体为约束实体, 否则为非约束实体。例如, 若 Δ 中的某个三角面片是 S 的一部分, 该面片为约束面片, 否则为非约束面片。

那么 Δ 满足什么条件, 会导致边界一致呢? Shewchuk 证明^[1], 若 Δ 中的约束边实体和约束三角面片实体满足以下条件, 则 Δ 是边界一致的。

定理(约束的存在性条件) 三角化 Δ 中, 若所有约束边实体没有被蚕食, 所有约束面片实体没有被蚕食, 则 Δ 是边界一致的。

所谓蚕食, 对于约束边实体, 指以约束边为直径的球包含了 Δ 中的其他顶点, 对于约束面片实体, 指过面片三顶点并以面片作为赤道平面的球包含了 Δ 中的其他顶点。所提出的边界恢复算法利用上述定理作为理论基础。然而, 由于上述定理需要的条件实在太强, 很难做到所有约束边实体和约束面片实体都不被蚕食, 为此在边界恢复时作了弱化处理。

2 算法概述

从输入模型 Ω 开始, 所提出的边界一致的四

面体网格生成算法要经过以下 4 个步骤。

(1) 边界离散。 Ω 中包含边界边和边界面, 边界离散的过程涉及到二个步骤, 首先要对边界边离散, 然后再对边界面进行离散。边的离散结果作为边界面离散的输入。边界面的离散即对其进行三角化。边界离散步骤如下。

步骤 1 对输入模型 $\{V, E, S, R\}$ 中 S 的每一个面的边界边和 E 中的每一条边, 调用曲线边离散算法进行离散, 即先对一维拓扑实体进行离散。

步骤 2 对输入模型 $\{V, E, S, R\}$ 中 S 的每一个面, 如果 S 为曲面, 则调用曲面三角化算法^[10]。如果 S 为平面, 则调用平面三角化算法^[11]。

(2) 三维无约束 DT。对第一步生成的点集进行随机扰动的无约束三维 DT, 得到的四面体网格构成点集的凸包。

(3) 边界一致。对第二步得到的四面体网格中的拓扑实体进行分类, 分为约束实体和非约束实体两类, 对那些还没有恢复的线段和三角片, 分别通过边恢复算法和面恢复算法进行恢复, 并通过种子点去除孔内的单元。

(4) 网格优化。以上三步得到的边界一致的网格质量可能很差, 通过逐步插点来精化网格。最后再用交换和插点的方法去除 Sliver 三角形。

图 1 演示了该算法的大致过程。在以下三节中详细讨论第(2), (3)和(4)三个步骤。

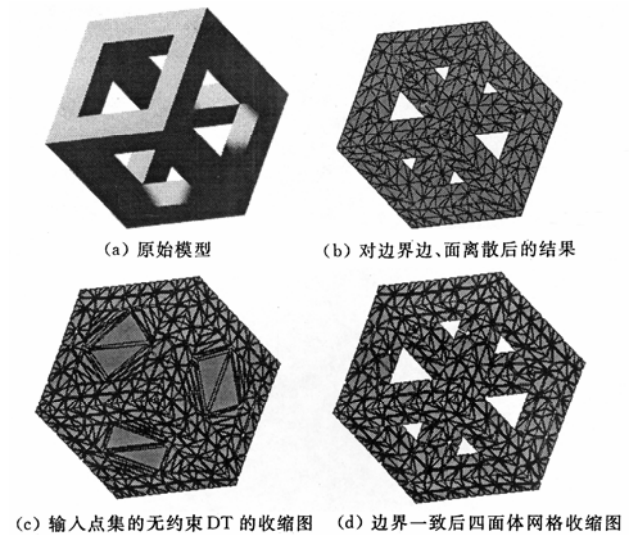


图 1 边界一致的 Delaunay 四面体网格生成步骤

3 随机扰动的增量插入 DT 算法

Watson^[6]和 Bowyer^[7]提出的增量插入 DT 算法每次插点时形成插入多边形, 但由于数值原因插入多边形很容易导致拓扑错。Joe^[12]将 Lawson^[13]基于

边交换的方法成功地扩展到三维,它最大的优点是始终保持一致的四面体拓扑结构。然而,基于交换的方法可能会由于数值误差的影响而不收敛,导致出现所谓的伪 Delaunay 四面体网格^[12]。通过对点的随机扰动,使点所处的位置随机化。因为试验表明^[14],处于特殊位置的点是导致以上算法失败的原因。这里的特殊位置,主要包括两种位置,一是多于 4 个点处于共球的位置上,这些点容易导致局部面最优判断时产生错误;二是指多于 3 点共面的位置,这些点容易导致点定位操作时出错。

通过对点进行随机扰动,大大减少了点处于特殊位置的可能性。除此之外,再对一些细小的特征延迟处理,细小的尺度也很容易导致点的定位出错。算法过程可用以下伪代码来描述。

算法 1 增量插点算法(V)

步骤 1 对 V 中的点进行随机扰动。

步骤 2 构造初始四面体 T_0 ,使得 V 包含在 T_0 的内部。

步骤 3 对于 V 中的每一点 p ,依次进行 4, 5, 6, 7 步。

步骤 4 定位 p 点所在的四面体 T_i 。

步骤 5 p 点距 T_i 的顶点是否小于指定的距离,若是,则将 p 点加入延迟插入点队列 DQuere,否则转下一步。

步骤 6 根据 p 点与四面体 T_i 的关系,将四面体 T_i 进行分裂,并将候选交换面进入交换面队列 FlipQuere。

步骤 7 对 FlipQuere 中的每一交换面,若不足局部最优,则进行交换或延迟交换,并更新 FlipQuere,直到 FlipQuere 为空。

步骤 8 对延迟插入点队列 DQuere 中的每一点,调用第 4、6 和 7 步。

步骤 9 删除所有与初始四面体共顶点的单元。

算法 1 中,第 1 步对点集 V 中的每一点进行随机扰动,所谓随机扰动,指对点平移一个很小的随机矢量(实现采用 10^{-5}),这样不但能有效地避免 5 点共球等退化现象,而且能避免多点共面时导致的定位误差。对提高算法的稳定性具有很好的作用。第 2 步构造一临时的“包容”四面体,包含所有点,以避免处理点落在点集合包外时的情形。第 4 步定位点所在的四面体,这是该算法最耗时的一步,第 5 步和第 8 步是提高算法稳定性采取的又一措施,我们发现,若网格中存在非常细小的单元,这些单元会影响到点的定位,尽量把细小单元的生成放在最后,可以延迟这些单元的出现。为此引入了延迟

插入点队列保存这些点。第 6 和第 7 步处理新插入点。

算法 1 的实现只需要两个几何测试,其他都是拓扑操作,这两个几何测试是:

测试 1 点在四面体外接球内的判断。

测试 2 点与三维空间面的相对位置的判断。

其中测试 1 测试在判断面是否局部最优时使用(算法 1 第 7 步),测试 2 在定位点以及分裂四面体时使用。由于对点进行了随机扰动和延迟插入,这两个测试都能很容易地得出正确的结果。

为测试算法 1 的稳定性与效率,分别用空间均匀分布的随机点、大致分布在球上的随机点、以及大致分布在立方体 4 个外表面上的随机点进行了测试。使用了 3 种方法,即随机扰动的增量插入算法、原始的增量插入算法^[6]和带精确计算的增量插入算法^[1],表 1 是测试的结果。

从表 1 可以看出,随机扰动的增量插入算法比原始的增量插入算法稳定,比精确计算的方法速度快。

表 1 无约束 Delaunay 四面体网格生成所花时间
(在 PIII800, 128 MB 内存上的测试结果) s

点分布	所用方法	点 数						
		10×10^3	50×10^3	100×10^3	150×10^3	200×10^3	250×10^3	300×10^3
均匀	随机扰动	0.6	3.2	7	11.9	16.7	22.1	28.9
均匀	原始	0.6	3.2	7	11.9	16.7	22.1	28.9
均匀	精确	0.8	4.0	10.3	15.8	21.5	28.3	36.2
球上	随机扰动	0.7	3.5	7.9	12.7	17.1	23.5	29.5
球上	原始	0.7	错	错	错	错	错	错
球上	精确	2.4	11.4	24.4	35.2	51.3	56.3	91.3
立方体边界上	随机扰动	0.7	3.5	7.9	12.7	17.1	23.5	29.5
立方体边界上	原始	错	错	错	错	错	错	错
立方体边界上	精确	1.8	7.9	16.8	26.9	42.5	50.1	64.3

4 边界一致

Gill Barequet^[15]证明,对于空间封闭的多边形,要确定其是否可三角化是一个 NP 问题,更糟糕的是,Ruppert 和 Seidel^[16]已经证明,要确定任意多面体是否可通过不插入点的约束方法恢复其边界是一个 NP 问题,因此,三维不插入点的约束 Delaunay 算法无论在理论上还是在实践上都不具有普遍意义,实用化的边界恢复算法通常要在边界或者内部插入额外的点。

边界一致 Delaunay 三角化中,既可以根据约束边和面的存在性条件预先生成足够多的点,则对点

集的无约束 DT 必定导致约束边和面的边界一致，例如 Krysl^[17]的线性规划方法。这个方法的计算量过大。也可以先对点集进行网格剖分，然后找出没有出现在网格中的约束边实体和约束面实体，通过恢复算法对这些约束实体进行恢复，该算法即为这种情况。

边界恢复的过程是恢复约束边和约束面的过程。先恢复约束边，所有的约束边恢复后再恢复约束面，许多恢复算法求解四面体网格与要恢复的约束实体的交点坐标，然后在交点处插入顶点来恢复约束实体。由于该交点可能与边十分接近，导致产生一个很小的特征，必须再进行网格光顺的处理。而该算法将点插在外接圆圆心处或约束段中点处，避免了这种情况。

4.1 边恢复

对于本应出现在网格中的约束边实体，若在网格中没有出现，则可以通过插点来使其出现。因为 Delaunay 四面体中任一点总是与距它最近的点有连接，经过有限次插点后，线段必出现。然而，插入新点时由于面交换可能会导致破坏已恢复线段。为了避免这种恢复线段的反复过程，只要线段恢复后就不允许再交换该线段，也就意味着只要线段恢复后，该线段就永远存在四面体化中。由于线段所在的边即使是非 Delaunay 边也不允许交换，故在恢复线段的过程中不能保证四面体总是 Delaunay 的。

边恢复算法可用以下伪代码表示。

算法 2 边恢复(ab)

- 步骤 1 在 ab 的中点 c 处插入一点。
- 步骤 2 若 ac 没恢复，递归调用边恢复(ac)。
- 步骤 3 若 bc 没恢复，则递归调用边恢复(bc)。

在所有约束边实体都恢复之后，再执行一次投影操作，将所有新插入的点都投影在它所关联的曲线上，图 2 演示了这一点。

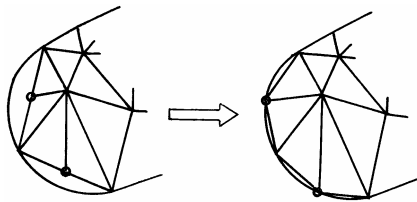


图 2 新插入点到关联曲线的投影操作

4.2 面恢复

尽管所有的约束边都已恢复，但这并不意味着所有约束子面都出现在四面体网格中。若输入约束三角片还没有出现在网格中，通过对约束三角片单元和四面体网格的面相比较，就能分辨出哪些子面需要恢复。对于需要恢复的子面要进行子面插点

操作。

所谓子面插点，即通过在该子面的外接圆圆心处插入一新点来切分该子面。但是，如果要插入的新点(也就是子面外接圆圆心)导致某一条或多条约束边被蚕食，则它暂不插入。而对该顶点所蚕食的所有约束边进行线段插点。这种操作被称为子面插点。

面恢复的算法可用以下伪代码表示。

算法 3 面恢复(要恢复约束子面的集合 L)

步骤 1 对于 L 中的每个约束子面 L_i ，依次做第 2 步和第 3 步。

步骤 2 如果 L_i 和四面体网格的面 F_i 重合，则标记 L_i ，否则计算新插点位置 v 。

步骤 3 如果 v 蚕食别的子段 s ，则在线段 s 的中点插点 v_s ，否则插入 v 到四面体网格中。

通过算法 3，所有的子面都恢复了，也不存在被蚕食的子段，但还存在被蚕食的子面，因此算法 3 得到的网格不一定完全满足 Delaunay 条件了。

最后对新插入到约束实体上的点进行投影操作，将他们投影到所关联的曲线和曲面上。

5 网格优化

边界边和边界面恢复后形成的边界一致 DT 中，只在边界边和边界面上插入了点，而在域内还没生成点。我们采用 Shewchuk^[1]的点生成策略，即当四面体的外接圆半径与最短边长之比大于某一上限 B 时，则把该四面体称为质量差的四面体。通常在质量差的四面体的外接球球心处插点来切分该四面体，并最终去除质量差的单元。称四面体外接球球心处插点的操作为四面体插点。然而，若新点(单元的外接球球心)蚕食子段或子面，则暂不插入它；而是先切分所有的被蚕食子段。如果这时该质量差的四面体还没有被去除，则切分所有的被新点蚕食的子面。也就是说，若通过子段插点，就能去除质量差的四面体，则算法不允许再切分新点所蚕食的子面。

维护三个队列，分别是蚕食子段队列、蚕食子面队列和质量差四面体队列。除了网格精化开始阶段，蚕食子段队列、蚕食子面队列内包含的对象一般很少。通过遍历并检测所有的子段、子面和四面体来初始化蚕食子段队列、蚕食子面队列和质量差四面体队列。测试一个新插入顶点是否蚕食子段和子面的最好方法是把这个新点插入网格中，然后测试该新点所对的约束子段和子面。如果该点蚕食子面和子段，则分别把它们插入蚕食子段队列和蚕食子面队列中，然后在三角化中删除该新点。

在算法中，对质量差四面体队列的维护是至关重要的。每插入一个新点，测试每个新生成的四面体，如果该四面体的质量达不到给定的质量要求，则把它添加入质量差四面体队列中。对在质量差四面体队列中的四面体根据质量进行排序。显然，先对 B 值较大的四面体进行四面体插点操作，有利于减少四面体插点的次数，提高算法的效率。

算法 4 质量保证(四面体网格 $\Delta=(P, F)$ ，要求的质量 B)

将网格 Δ 中所有质量大于 B 的四面体放到链表 list

循环(list 中每一四面体 T)

$p = T$ 的外接球球心

如果(p 点在约束边子段的直径球内)

插入子段中点到网格 Δ 中

将新生成的所有质量大于 B 的四面体放到

链表 list

否则，如果(p 点在约束子面的赤道球内)

插入子面外接圆圆心到网格 Δ 中

将新生成的所有质量大于 B 的四面体

放到链表 list

否则

插入 p 到网格 Δ 中

结束如果

结束循环

显然恢复被蚕食子段的优先级要高于被蚕食子面，被蚕食子面的优先级要高于形状差的四面体。

图 3 概括了四面体网格生成中经常会出现的一些差单元^[8]，其中的针状四面体、楔形四面体和帽形四面体都具有较大的 B 值，在内点的形成过程中都将会通过插点去除，然而 Sliver 四面体却例外，它具有很小的 B 值，体积几乎为 0，因此经过算法 4 后不会被去除。

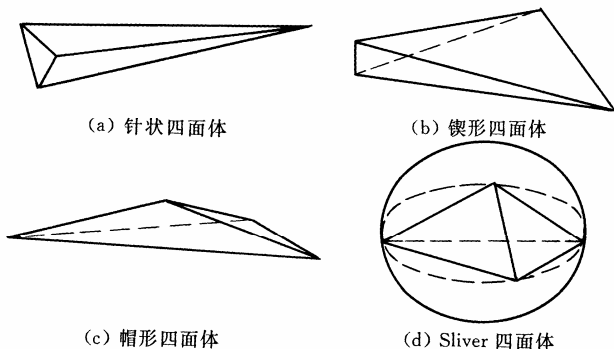


图 3 几种差质量的四面体

通过研究发现，Sliver 四面体与相邻四面体的位置关系有三种情况。如图 4 所示，图 4a 所示的

Sliver 四面体 $bdca$ 在边界上，只需直接去除即可。图 4b 上图所示的 Sliver 四面体 $bdca$ 在内部，若其可交换，则可以通过 3-2 交换^[12]来去除，变成图 4b 下图的形状，若其不可交换，则只能通过在离 Sliver 四面体某一平面有一定距离处插入一个新点并进行局部交换来尝试去除它，这种情况不一定能完全去除 Sliver 四面体。

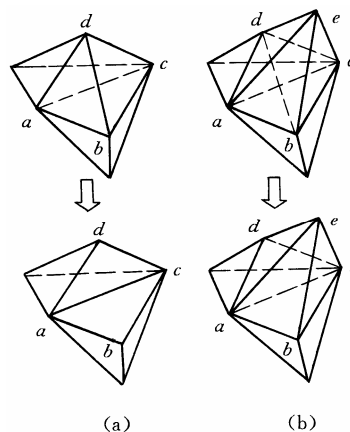


图 4 Sliver 四面体的去除

经过算法 4 和 Sliver 去除后的网格质量有很大改进，但还不令人满意，我们通过以下两种方式对网格进一步优化。

(1) 通过 2-3, 3-2 局部面交换操作^[12]改变某些单元的拓扑结构来优化网格。

(2) 通过 Laplacian 光顺^[18]移动顶点的几何位置来改进网格质量。

图 5 给出了图 6 中二个例子在网格优化前和优化后单元双面角的对照图，图中横坐标表示双面角($^\circ$)，纵坐标表示双面角的数量。由于一个四面体有 6 个双面角，因此每个网格的双面角总数是单元总数的 6 倍。从图中可知。经过优化后，绝大部分双面角都在 30° 到 120° 之间(支座 88 022 个，占总数 91 476 的 96%，四通 26 615 个，占总数 28 374 的 93.8%)。比优化前有较大的改进(支座 67 595 个，占总数 91 476 的 73.9%，四通 20 483 个，占总数 28 374 的 72.2%)。而且去除了绝大部分 Sliver 单元(见表 2)。

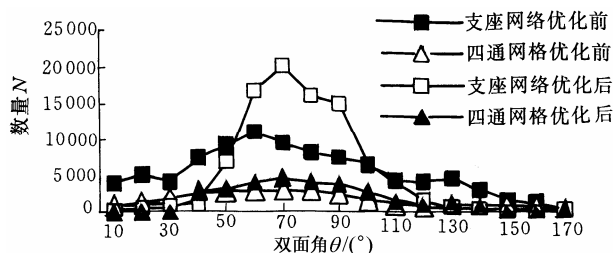


图 5 网格优化前后双面角对照图

6 剖分实例

图 6 给出了两个用该方法生成的四面体网格例子。左图是一个四通薄壁零件, 右图为一个支座类的机械零件。表 2 是这两个零件网格参数的统计结果。表 2 及图 6 表明, 该算法对这两个零件成功地产生了边界一致的四面体网格单元, 在具有边界约束的条件下, 平均一秒钟生成的四面体单元个数超过 500 个, 而且在优化后的质量也不差(见图 5)。

表 2 网格参数的统计结果(PIII800, 128MB 内存, Window2000, ACIS 造型引擎下得出的结果)

模型	节点数 N	单元数 E	剖分时间 s	优化前 Sliver 单元数 E_{sp}	优化后 Sliver 单元数 E_{sa}	约束边数 S_e	约束面片数 F_c
四通	1 528	4 729	9	210	3	3 965	2 650
支座	3 328	15 246	20	126	0	4 528	3 154

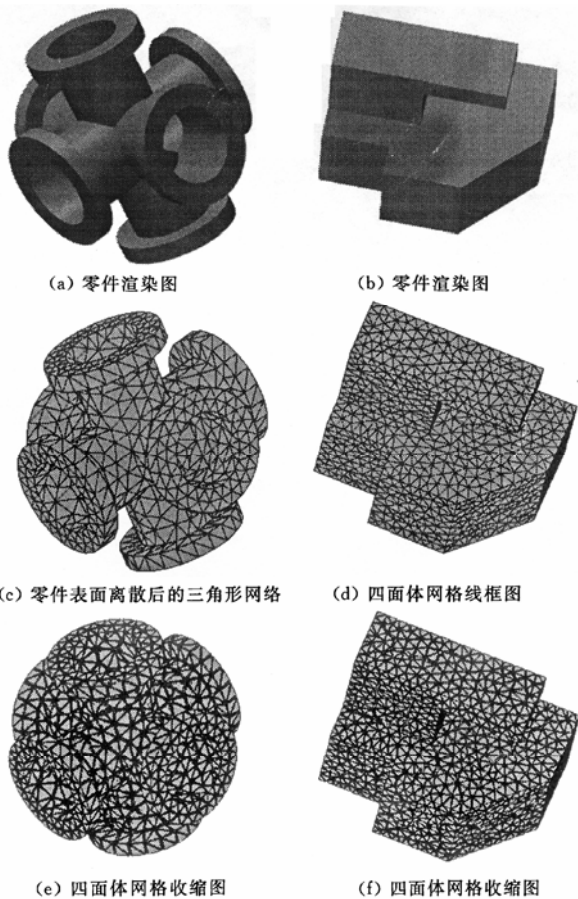


图 6 四面体网格生成例子

7 结论

四面体网格生成算法具有重要的理论与实践意义。针对四面体 DT 网格的稳定性问题, 提出了随

机扰动以及延迟插入技术, 在不牺牲速度的条件下, 提高了稳定性; 针对边界一致问题, 改进了约束边和约束面恢复算法。针对质量问题, 提出了交换和插点相结合的网格优化及 Sliver 单元的去掉方法。最后以几个例子检验了算法。

参 考 文 献

- 1 Shewchuk J R. Delaunay refinement mesh generation: [PhD Dissertation]. Pittsburgh: Canegie Mellon Univ., 1997
- 2 Owen S J. Non simplicial unstructured mesh generation: [PhD Dissertation]. Pittsburgh: Canegie Mellon Univ., 1999
- 3 徐永安. 约束 Delaunay 三角化的关键问题研究与算法实现及应用: [博士学位论文]. 杭州: 浙江大学, 1999
- 4 赵建军. 非结构化网格生成与压缩技术的研究: [博士学位论文]. 武汉: 华中科技大学, 2002
- 5 Rajan V T. Optimality of the Delaunay triangulation in R^d . Discrete & Computational Geometry, 1994, 12: 189~202
- 6 Watson D F. Computing the Delaunay tessellation with application to voronoi polytopes. Computer Journal, 1981, 24(2): 167~172
- 7 Bowyer A. Computing dirichlet tessellations. Computer Journal, 1981, 24(2): 162~166
- 8 Lewis R W, Yao Z, Gethin D T. Three dimensional unstructured mesh generation: Part 3. Volume meshes. Comput. Methods Appl. Mech. Engrg., 1996, 134: 285~310
- 9 Staten M L, Jones N L. Local refinement of three-dimensional finite element meshes. Engineering with Computers, 1997, 13: 165~174
- 10 Zhao J J, Wang Q F, Zhong Y F. Delaunay triangulation method of curved surfaces based on Riemannian metric. Chinese Journal of Mechanical Engineering, 2003, 16(1): 91~93
- 11 赵建军, 钟毅芳, 胡于进, 等. 基于面结构的三角化新方法. 计算机学报, 2000, 23(9): 1 003~1 008
- 12 Joe B. Construction of three-dimensional improved-quality triangulations using local transforms. SIAM J. of Sci. Compt., 1995, 6: 1 292~1 307
- 13 Lawson C L. Software for C_1 surface interpolation. Mathematical Software III, 1977, 5: 161~194
- 14 Cavalcanti P R, Mello U T. Three-dimensional constrained Delaunay triangulation: A minimalist approach. In: Proceedings 7th International Meshing Roundtable, Michigan, 1998: 273~284
- 15 Gill B, Matthew D. On triangulating three-dimensional polygons. Computational Geometry: Theory and Appli-

- cations, 1998, 10(3): 155~170
- 16 Ruppert J, Seidel R. On the difficulty of triangulating three-dimensional non-convex polyhedra. *Discrete Computational Geometry*, 1992, 7: 227~253
- 17 Krysl P, Ortiz M. Generation of tetrahedral finite element meshes: variational Delaunay approach. In: *Proceedings of 7th International Meshing Roundtable*, Michigan, 1998: 273~284
- 18 Scott C, Joseph T, Matthew S. An approach to combined Laplacian and optimization-based smoothing for triangular, quadrilateral, and quad-dominant meshes. In: *Proceedings of 7th International Meshing Roundtable*, Michigan, 1998: 421~436

STABLE TETRAHEDRAL MESH GENERATION ALGORITHM WITH BOUNDARY CONSISTENCY

Zhao Jianjun Wang Qifu
(*Mechanical Science and Technology School,*

Huazhong University of Science and Technology,
Wuhan 430074)

Abstract: A stable tetrahedral mesh generation algorithm is presented, which improves the present three dimensional Delaunay tetrahedral mesh generation methods with the following techniques: Random perturbation and lazy inserting techniques are introduced to improve the robustness with less speed decrease; Constrained segments restoration and constrained facet restoration algorithm are improved to deal with curved boundary and carry out boundary consistency; Mesh optimization combining the facet flipping and vertex insertion methods and Sliver elimination method are proposed to improve the mesh quality. Several examples are given to validate the proposed algorithm.

Key words: Mesh generation Tetrahedral mesh
Delaunay Boundary consistency

作者简介: 赵建军, 男, 1973 年出生, 博士。主要研究方向为网格生成、CAD 和计算几何。

E-mail: zhaojj@hustcad.com