

# TetGen

一款优良的四面体网格生成和 3D Delaunay 三角剖分器

版本 1.5

用户手册

2012 年 11 月 5 日

Hang Si

[si@wias-berlin.de](mailto:si@wias-berlin.de)

<http://www.tetgen.org>

© 2002 – 2012

周顺 陈强译

E\_mail: shunzhou@foxmail.com

2014 年 1 月

## 译者序

三角形为三维建模中较为基础图元,任何面状图形都可以分割为多个三角形的形式进行表达。四面体作为体的基础单形,任何体即可分割为多个四面体的形式进行表达。因此,三角剖分在三维建模中具有重要的作用,广泛应用于几何造型、仿真、虚拟现实、医学、数字城市、矿山、水文、气象等领域。四面体剖分已被有限元分析、计算机图形学、科学计算可视化、生物医学和地理信息系统等领域广泛使用。四面体格网生成的效率、质量是影响这些领域应用的较为关键的技术。

TetGen 为德国计算数学与科学计算应用分析和随机指数维尔斯特拉斯研究所(WIAS) Hang Si 编写。2013 年 11 月 5 日发布了 TetGen 1.5.0 版本。可使用在所有主流的操作系统上,如: Unix/Linux,Mac,Windows 等。TetGen 可以认为是美国加州大学克利分校计算机科学部 Jonatan Richard Shewchuk 编写的 Triangle ([www.cs.cmu.edu/~quake/triangle.html](http://www.cs.cmu.edu/~quake/triangle.html)) 二维格网生成和三角剖分软件的高维推广。TetGen 提供了较为丰富的命令,能够满足较多领域的应用,在开源四面体剖分软件中较为容易学习和掌握。

译者出于学习四面体剖分知识和深入研究的 TetGen 的使用方法的目的,对 TetGen 1.5.0 使用手册进行了翻译。由于英语水平和专业知识有限,译文中难免出现不足之外,敬请读者给予批评指正!

周顺

# 目 录

<b>1 引言 .....</b>	<b>2</b>
1.1 点集的三角剖分.....	3
1.1.1 Delaunay 三角剖分,Voronoi 图 .....	3
1.1.2 权重 Delaunay 三角剖分,权重图.....	4
1.1.3 算法.....	5
1.2 四面体剖分.....	6
1.2.1 分段线性复合形(PLC, Piecewise Linear Complexes).....	6
1.2.2 边界一致性,Steiner 点 .....	7
1.2.3 约束 Delaunay 四面体剖分.....	8
1.2.4 格网矢量,四面体形状判定.....	9
1.2.5 格网尺寸,格网自适应性.....	11
1.2.6 格网优化.....	11
1.2.7 算法.....	11
<b>2 入门 .....</b>	<b>12</b>
2.1 编译.....	12
2.1.1 Unix/Linux/MacOSX.....	13
2.1.2 Windows9.x/NT/2000/XP .....	13
2.2 测试.....	14
2.3 可视化.....	17
2.3.1 TetView.....	17
2.3.2 Medit.....	17
<b>3 使用 TetGen.....</b>	<b>18</b>
3.1 命令行语法.....	18
3.2 命令行开关.....	18
3.2.1 四面体格网化一个点集(-w, -v).....	19
3.2.2 分段线性复合体(PLC)四面体格网化(-p, -Y).....	21
3.2.3 生成优良格网 (-q).....	23
3.2.4 分配范围属性(-A).....	24
3.2.5 体积约束(-a).....	25
3.2.6 应用格网尺寸函数(-m).....	25
3.2.7 重建四面体格网(-r) .....	25
3.2.8 插入附加点列表(-i).....	26
3.2.9 格网优化(-O).....	26
3.2.10 设置容差(-T) .....	26
3.2.11 格网统计(-V).....	27
3.2.12 其他选项.....	27
<b>4 文件格式.....</b>	<b>27</b>
4.1 TetGen 文件格式.....	27
4.1.1 .node 文件.....	28
4.1.2 .poly 文件.....	28
4.1.3 .smesh 文件.....	30
4.1.4 .ele 文件 .....	31

4.1.5 .face 文件 .....	32
4.1.6 .edge 文件 .....	33
4.1.7 .vol 文件 .....	33
4.1.8 mtr 文件 .....	33
4.1.9 var 文件 .....	33
4.1.10 .neigh 文件 .....	34
4.1.11 .v.cell, .v.face, .v.edge .....	34
4.2 支持的外部文件格式 .....	34
4.2.1 .off 文件 .....	34
4.2.2 .ply 文件 .....	35
4.2.3 .stl 文件 .....	35
4.2.4 .mesh 文件 .....	35
4.3 文件格式示例 .....	35
4.3.1 两个边界标识的 PLC .....	35
4.3.2 带有两个范围的 PLC .....	38
<b>5 示例 .....</b>	<b>40</b>
5.1 四面体格网生成(-q, -V) .....	40
5.2 格网重建选项(-a, .var 文件) .....	40
<b>参考文献 .....</b>	<b>40</b>

## 摘要

**TetGen**是一款四面体格网生成器。并以科学实验和工程应用为目标。它可以作为一个独立的程序使用，或者作为一个组件的形式嵌入到其他软件中。

本文档简短地说明了生成格网的程序，详细地讲解了**TetGen**的使用方法。读者将学会如何使用输入文件通过命令行生成四面体格网。此外，讲解了如何使用其他程序调用**TetGen**的程序接口。

关键字：四面体格网生成，Delaunay四面体剖分，约束Delaunay四面体剖分，常规三角剖分，格网质量，格网细化，自适应格网

# 1 引言

TetGen是一款生成四面体格网和Delaunay四面体剖分的C++程序。

如果输入的仅仅是一些3D点集，TetGen生成Delaunay四面体和这些点集的Voronoi图。如果是一组加权的点集，也就是说每个点分配一个实数作为它的权重，TetGen生成权重Delaunay四面体和这些带有权重点集的权重图。

如果输入的一个大概的3D空间表面网格边界，TetGen生成这个表面网格围成的空间的四面体格网。用户可以决定这个输入的表面应该是不变的（不被修改）还是可以改变的。如果是可修改的，这个选项可以帮助创建一个较为优良的四面体格网。另外，设置这个输入的表面为不可修改的，这个选项可以有益于局部重构，格网组合和并行构网等。TetGen提供灵活的格网细化选项以改善格网质量和生成适合用户目的的格网。

TetGen可以拿预先生成好的四面体格网作为输入和在此基础上执行各种操作，例如：局部格网细分，优化格网，线性插值等。

在剩下的章节中简短地描述TetGen中三角剖分和构网问题，以及执行的算法。然而，这些信息对大多数用户来说不是重要的。你可以选择跳过这些章节，但是在1.2.1和1.3.1中包含了使用TetGen成功生成一个合适格网的要点。

## 基础名词

如果一个点集  $V \subset \mathbb{R}^d$  包含的每条线段的端点都在这个点集内，则称该点集是一个凸面体(convex)。点集  $V$  的凸面体有无穷多个。其中，称  $V$  的最小凸面体为  $V$  的凸包(convex hull)。 $V$  的凸包维度(dimension)与  $V$  的仿射空间维数相同。

一个  $K$  维的单纯形( $k$ -simplex)  $\sigma$  是  $k+1$  个仿射无关点的凸包。例如：0-、1-、2-和3-单纯形分别为顶点，边界线，三角形和四面体。设  $V$  为  $k+1$  个顶点，其凸包为  $k$ -单纯形  $\sigma$ 。 $\sigma$  的面(face)是任何子集  $S \subseteq V$  的凸包，当然，它也是一个单纯形，其维数小于等于  $\sigma$ 。如果  $S \subset V$  则该面是一个真面(proper face)。 $\sigma$  的边界 (boundary)是所有  $\sigma$  真面的并集。 $\sigma$  的内部(interior)即  $\sigma$  的点集减去它的边界。

一个单纯复形(simplicial complex)  $S$  是单纯形的一个有限集，例如：(i)单纯形  $\sigma \in S$  的任意一个面也是  $S$  的一个面。(ii)任意两个单纯形  $\sigma, \tau \in S$  的交集，要么为空要么为两个单纯形的公共面。 $S$  的维度是它们单纯形的最大维度。 $S$  的基础空间(underlying space)为  $S$  所有单纯形的并集。注意， $S$  所有单形体的内部分解了  $S$  的基础空间。

### 1.1 点集的三角剖分

三角剖分是基本的几何结构。一个点集  $V$  的三角剖分(triangulation)是一个单纯复形  $S$ ，它的顶点集合等同于  $V$  或是  $V$  的子集， $S$  的基础空间是  $V$  的凸包。同一点集存在许多种剖分方法。它们当中，Delaunay三角剖分和权重Delaunay三角剖分是最值得关注的。它们具有许多优良的数学性质。最优算法可用来高效的构建它们。

TetGen生成3D Delaunay和权重Delaunay三角剖分以及它们的对偶图：Voronoi图 and 权重图。

#### 1.1.1 Delaunay 三角剖分,Voronoi 图

设  $V$  为  $\mathbb{R}^d$  空间的点集， $\sigma$  为  $k$ -单纯形( $0 \leq k \leq d$ )，单纯形的顶点存储在  $V$  中。 $\sigma$  的外接球(circumsphere)经过了  $\sigma$  的所有顶点。如果  $k = d$ ， $\sigma$  有唯一的外接球，否则， $\sigma$  有无穷多个外接球。如果  $\sigma$  存在一个外接球，外接球内没有  $V$  的顶点，那么我们认为  $\sigma$  是具有Delaunay特性的。

$V$  的Delaunay三角剖分  $D$  是一个单纯复形，所有的单纯形都具有Delaunay特性， $D$  的基础空间为  $V$  的凸包[11]。图1(a)说明了一个二维Delaunay三角剖分。在三维空间中，它也可以称为Delaunay四面体剖分(Delaunay tetrahedralization)。

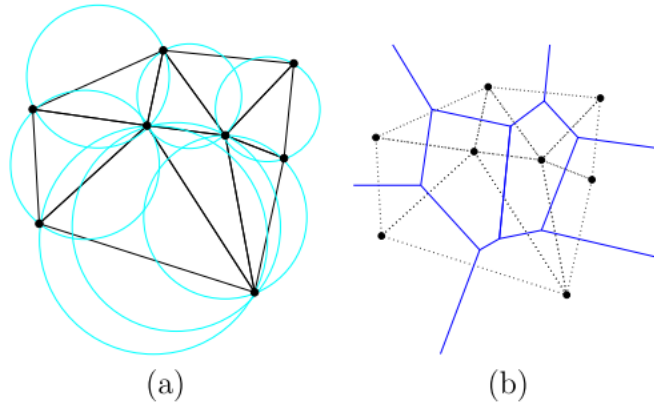


图 1:二维点集的 Delaunay 三角剖分(a)以及它的对偶 Voronoi 图(b)

如果  $V$  在常规位置(*general position*)，那么  $V$  的Delaunay三角剖分是唯一的，也就是说，

在  $V$  中不存在  $d+2$  个点共处于同一个球上。如果存在，我们认为  $V$  包含简并 *Delaunay* (*Degeneracies*)，也就是说，它们在  $V$  中有共球点。简并 *Delaunay* 可以通过在  $V$  点集的坐标上应用任意小的扰动值来移除掉。

*Delaunay* 三角剖分的对偶为相同点集定义的 *Voronoi* 图(见图1(b))。对于任何顶点  $p \in V$ ， $p$  的 *Voronoi* 单胞(*Voronoi cell*)是到  $p$  的距离不比到  $V$  中其他顶点的距离大一个点集，( $p$  与周围点连线的垂直平分线组成的区域) 也就是：

$$cell(p) = \{x \in R^d; \|x - p\| \leq \|x - q\|, \forall q \in V\}$$

其中， $\|\cdot\|$  表示欧几里德距离。 $V$  的 *Voronoi* 图是将  $R^d$  空间细分为 *Voronoi* 胞(有一些是无限大的)和它的面[46]。它是  $d$ -维多面复形。如果点集  $V$  处于常规位置，它们在 *Delaunay* 三角剖分的  $k$ -维单纯形与 *Voronoi* 图的  $(d-k)$  多面体之间是一一对应的，其中  $0 \leq k \leq d$ 。在  $R^3$  域中，*Voronoi* 图的顶点就是 *Delaunay* 四面体剖分的四面体的外接球心。

在  $R^d$  域的 *Delaunay* 三角剖分和  $R^{d+1}$  域中的凸包之间具有很好的关系。对于任意点  $p = (p_0, p_1, \dots, p_{d-1}) \in R^d$ ，定义提升点 (*lifted point*)  $p^+ = (p_0, p_1, \dots, p_{d-1}, p_d) \in R^{d+1}$ ，其中  $p_d = p_0^2 + \dots + p_{d-1}^2$ 。对于任何的点集  $V \in R^d$ ，定义  $V^+ = \{p^+; p \in V\} \subset R^{d+1}$  为  $V$  的提升点集。在  $V^+$  中所有的点都位于  $R^{d+1}$  域的抛物面上(见图2(a))。 $V^+$  的凸包是  $(d+1)$  维凸多胞形  $P$ 。

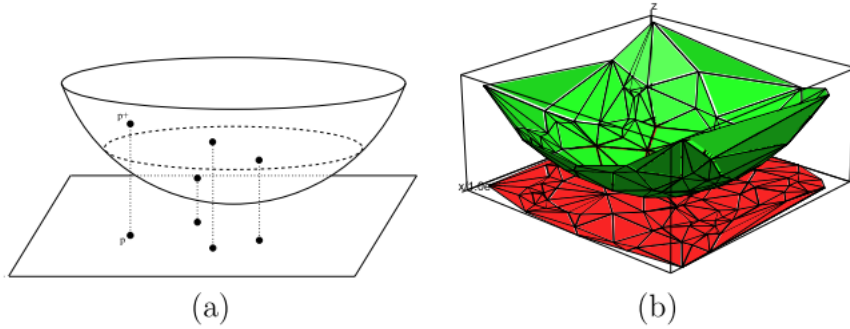


图2  $R^d$  域的 *Delaunay* 三角剖分和  $R^{d+1}$  域中的凸包(这儿  $d=2$ )之间的关系。(a)一些二维点和他对应的三维提升点。(b)二维点集的 *Delaunay* 三角剖分和它的3维提升点的降维凸包。

$P$  的低维面(*lower face*)是位于  $P$  的底侧(借助  $V$  集中点显示出来)的一个面， $V$  的 *Delaunay* 三角剖分是从  $P$  的低维面集合向  $d$  维空间的投影。图2说明了当  $d=2$  时这一关系。当且仅当在  $R^{d+1}$  域中存在一个超曲面经过  $\sigma$  所有提升点，这样在  $V^+$  域中没有其他提升点位于超曲面下方，此时我们认为单纯形  $\sigma$  是 *Delaunay* 单纯形。类似的， $V$  的 *Voronoi* 图是一个凸多胞形  $Q \subset R^{d+1}$  的低维面的投影，这样  $P$  和  $Q$  正好是相反的[48]。

*Delaunay* 三角剖分和 *Voronoi* 图有许多有利的和优良的性质，见[2,30,14]。它们在很多应用中都被广泛使用。

### 1.1.2 权重 *Delaunay* 三角剖分,权重图

权重 *Delaunay* 三角剖分是 *Delaunay* 三角剖分借助权重距离的推广。一个权重点 (*weighted point*)  $p' = (p, p^2) \in R^d \times R$  可以理解为一个以  $p$  为球心  $p$  为半径的球。 $p'$  和  $z'$  之间的加权距 (*weighted distance*) 为：

$$\pi_{p',z'} = \sqrt{\|p - z\|^2 - (p^2 + z^2)}$$

在  $R^d$  空间的点可以认为是权重值为0的点。如果俩个权重点  $p'$  和  $z'$  的权重距离为0，也



就是  $\|\mathbf{p} - \mathbf{z}\|^2 = (p^2 + z^2)$ ，那么  $\mathbf{p}'$  和  $\mathbf{z}'$  是正交的(orthogonal)；当两个权重点权重距离为正时，认为这两个权重点是远离正交的(further than orthogonal)；当两个权重点距离为虚数时，那么这两个权重点是接近正交的(closer than orthogonal)。

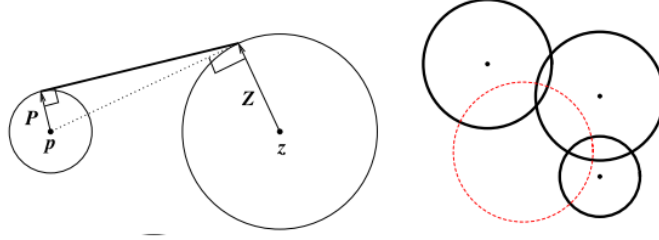


图3 左图：两个权重点  $(p, P^2)$  和  $(z, Z^2)$  的权重距离。右图：三个权重点的正交球(orthosphere)(图来自文献 [18])

一般来说，在  $R^d$  域中  $d+1$  个点确定一个经过这些点的唯一的外接球。类似的，在  $R^d$  域  $d+1$  个权重点确定唯一公共的正交球(orthosphere)。当所有的点权重都为0，它们的正交球(orthosphere)就是它们的外接球。图3(右)给出了一个二维域中正交球(orthosphere)的示例。

设  $V' \subset R^d \times R$  是权重点有限的集合。如果  $V'$  中所有的权重点都是远离正交的，那么我们认为这个球为空。 $V'$  的权重Delaunay三角剖分是单纯复形  $D'$ ，这样每个单纯形都有一个空的正交球。 $D'$  的基础空间是  $V'$  的凸包。显然，如果所有的点都有相同的权重，那么权重Delaunay三角剖分就跟常规Delaunay三角剖分一样。注意，一个权重Delaunay三角剖分不必要包含所有的  $V'$  中的点。

权重Delaunay三角剖分的对偶是权重Voronoi图(weighted Voronoi diagram)，也可以称为权重点集  $V'$  的权重图(power diagram)[2,16]。权重图类似于Voronoi图的定义，使用权重距离而不是欧几里德距离。如果  $V'$  中不存在  $d+2$  个权重点共享一个公共的正交球，即它在一个常规位置上，那么权重Delaunay三角剖分的单纯形和权重图的单胞是一一对应的。在  $R^3$  中，权重图的顶点是权重Delaunay四面体剖分的四面体的垂心。

一个权重Delaunay三角剖分  $V \subset R^d$  也是凸多胞形  $P \subset R^{d+1}$  降维面集合的投影。在  $\mathbf{p} = \{p_0, \dots, p_{d-1}\} \in V$  中的任意点提升到点  $\mathbf{p}' = \{p_0, \dots, p_{d-1}, p_d\} \in R^{d+1}$ ，其中  $p_d = p_0^2, \dots, p_{d-1}^2, p - p^2$  ( $p$  是  $\mathbf{p}$  的权重)。对于  $p \neq 0$ ， $\mathbf{p}'$  不在  $R^{d+1}$  空间中的抛物面上，但  $p^2$  会使其垂直向下移动。要使一个单纯形属于  $V$  的权重Delaunay三角剖分 (即它有一个空正交球)，当且仅当存在一个超平面通过这些单纯形的提升权重点而且  $V$  中没有位于超平面以下的提升权重点。

权重Delaunay三角剖分和权重图都被称为点集的“规则分割(regular subdivisions)” [48]。规则分割有较好的组合结构。在高维凸多胞形[48,10]中它们是重要的研究对象之一。

### 1.1.3 算法

在计算几何中对常规 Delaunay(以及权重 Delaunay)四面体剖分算法已有很好的研究，参考文献列表中[13,14]。TetGen 实现了两种算法，Bowyer-Watson 算法[5,17]和增量换边算法(incremental flip algorithm)[16]。这两种算法都是逐点插入。最坏的情况下两者的运行时间都为  $O(n^2)$ (因为 3D Delaunay 四面体剖分的复杂度[24])。在大多数实际应用中，它们通常都是非常快速的。如果这些点均匀分布在  $[0,1]^3$  域中，这些算法的预期运行时间为  $O(n \log n)$ 。

点的定位非常影响增量算法的速度。当  $n$  为较大时，简单的随机 walk 算法[12]和改善的 jump-walk 算法[26]会很慢。TetGen 使用了空间排序体制[23,4]以改善点的定位。它的优点在于不需要添加任何数据结构。排序这些点以至临近的空间点有临近的索引号。这些点首先在不同的组进行随机排序[1]。然后每个组中的点沿 Hilbert 曲线排序。以此顺序插入点，简单随机 walk 算法每个点运行的时间几乎不变。

为了执行 Orient3D, InSphere, 和 Orient4D 测试, TetGen 使用了 Shewchuk 的精确几何判断[34]。它们足够保证常规 Delaunay 和权重 Delaunay 四面体剖分的数值健壮性。使用一种简化的象征性的扰动方案[15, 39]来去除退化的三角形或者四面体。

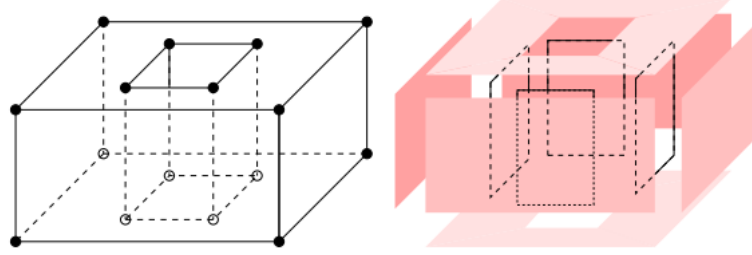


图 4: 一个三维的凹多面体, 左图表示多面体的顶点(0 维)和边(1 维), 右图表示多面体的所有面(2 维)。

## 1.2 四面体剖分

四面体网格是一个三维单纯复形, 是三维连续空间的离散表达, 在拓扑结构和几何结构上都有所体现。TetGen 为三维空间中的指定类型构建四面体网格, 使用分段线性复合形(章节 1.2.1)为其建模。TetGen 的重点在于几何体(边界)的表达及其网格的质量。

为了实现这些目标, TetGen 会生成四面体网格不同的类型。下面分不同的章节来介绍。

### 多面体与面

多面体(polyhedron) $P$ 是凸多面体的并集, 并且在 $P$ 的空间是连续的。它不一定非是凸多面体。 $P$ 的维度是由包含 $P$ 的最小仿射空间的维度来决定的。图4左图说明了一个三维多面体。 $P$ 的内部是个点集表示为  $\text{int}(P)$ , 每个点  $p \in \text{int}(P)$  都有邻接元素(比如以这个点为中心的球), 该邻接元素是 $P$ 的一个子集。 $P$ 的边界是一个点集  $bd(P) = P - \text{int}(P)$ 。

$P$ 的一个面(face) $F$ 是 $P$ 边界上的连续点集的闭包, 而且 $F$ 上的所有点都有包含 $F$ 的相同的最小维度仿射空间。它也是一个多面体, 其维度是决定 $F$ 的仿射空间的维度。 $P$ 的0维, 1维, 2维单纯形被称之为 $P$ 的顶点, 边, 面。就像图4右图阐明的一样, 每个面都是一个多边形区域, 可能不是凸多边形, 也可以包含洞。

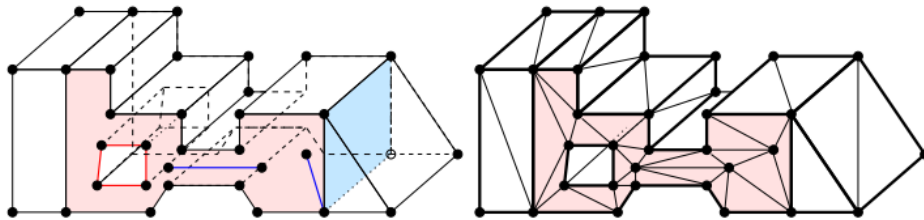


图5: 左图: 一个三维分段线性复合形, 是集点、边、面、多面体为一体的。左边阴影部分表示一个非凸的而且包含洞的面, 也有游离的边和顶点。右边阴影部分表示分离两个多面体(子区域)的内部面。右图: 表示左图PLC的四面体网格剖分。

### 1.2.1 分段线性复合形(PLC, Piecewise Linear Complexes)

三维几何域比多面体更为复杂。TetGen采用了更加通用的输入, 被称为分段线性复合

形(PLC)，由Miller等人[25]首先引入该概念。三维PLC  $X$  是一个集顶点，边，面和多面体(不一定为凸多面体)为一体的，统称为体元(cells)，满足以下属性：

- (1)  $X$  中的每个体元边界都是  $X$  中体元的结合。
- (2) 如果两个不同的体元  $f, g \in X$  相交，它们的交集是  $X$  中体元的结合。

PLC的定义不允许它的单体非法相交，比如，两个边的相交只能在一个公共的顶点处，并且该点位于  $X$  内；T的两个面的相交只能是顶点和线段的结合，并且位于  $X$  内。

在描述3维几何特征方面，PLC比多面体复合形更加灵活。比如，PLC允许面、边、点浮动在多面体区域内，或是边和点浮动在面上。图5左图就是一个例子。这些漂浮单体的一个目的就是限制PLC的四面体化，以至于边界条件可在这些单体中应用。

设  $X$  是一个三维PLC。PLC  $X$  底层空间(underlying space)表示为  $|X|$ ， $\bigcup_{f \in X} f$  是要被三角化的区域。 $X$  的四面体剖分(tetrahedralization)是一个三维单纯复形  $T$ ，有：(1)  $X$  和  $T$  具有相同的顶点；(2)  $X$  中的每个体元都是  $T$  中单纯形的并集；(3)  $|T| = |X|$ 。图5的右图展示了一个例子。

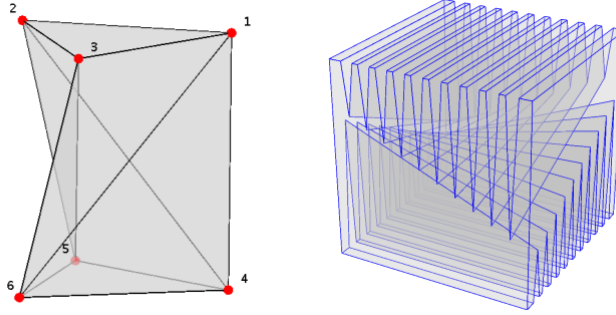


图 6：没有 Steiner 顶点多面体不能被四面体化。左图：Schonhardt 多面体。右图：Chazelle 多面体。

### 1.2.2 边界一致性,Steiner 点

不是每个三维PLC(或三维多面体)都能四面体剖分的。有时多面体四面体剖分时所需的顶点(称为Steiner点)必须比其本身[33,6]要多，请参照图6的两个例子。然而，如果Steiner点允许，使一个PLC四面体剖分是可以的。PLC  $X$  的Steiner四面体剖分(或是限定四面体剖分)是  $X \cup S$  的四面体剖分， $S$  是Steiner点的一个有限集(与  $X$  中的点不相交)。

TetGen可为任意的PLC生成Steiner四面体网格。必要的时候TetGen会添加Steiner点。当没有指定其他选择(比如网格细化或网格优化)时，TetGen会限制添加Steiner点的数量。根据添加的Steiner点的位置，由TetGen生成的(Steiner)四面体网格可被分为以下几类：

- 约束Delaunay四面体剖分(在1.2.3章节讲解)。它可能包含Steiner点，一些Steiner点位于PLC的边界上，比如，PLC的边界三角剖分是原始剖分的一个细化。包含的每个四面体都满足约束Delaunay的条件(参见1.2.3章节)。它有一些属性类似于Delaunay四面体剖分。
- 限定Delaunay四面体剖分。可能包含Steiner点。一些Steiner点位于PLC的边界上，比如，PLC的边界三角剖分是原始剖分的一个细化。另外，它是Delaunay四面体剖分的一个子复形体，即每个四面体都是Delaunay四面体。
- 约束四面体剖分。可能包含Steiner点。但是所有的Steiner点必须位于PLC空间内部，PLC的边界三角剖分是被保留的。这个四面体剖分可能既不是限定Delaunay四面体剖分也不是约束Delaunay四面体剖分。

默认情况下，TetGen生成PLC的约束Delaunay四面体网格。在TetGen中无论是限定Delaunay四面体剖分还是约束四面体剖分选项设置都是可用的。

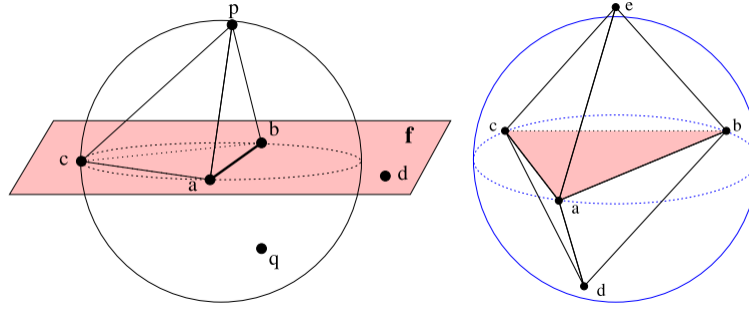


图 7: 左图: 阴影部分表示  $P$  的一个多边形  $f$ , 包括点  $a, b, c, d$ 。点  $p$  和  $q$  位于多边形  $f$  的两侧, 两者被分隔。 $S_t$  是四面体  $t(abc p)$  的外接球, 它包括了点  $q$  但不含点  $d$ — $t$  是约束 Delaunay。

### 1.2.3 约束 Delaunay 四面体剖分

设  $X$  是一个三维 PLC。 $X$  顶点的 Delaunay 四面体剖分不一定包含  $X$  的边和面, 约束 Delaunay 四面体剖分 (CDT, *constrained Delaunay tetrahedralization*) 是 Delaunay 四面体剖分的变异, 只是被迫遵从  $X$  的边和面。一个平面内的 CDTs 是由 Lee 和 Lin[20] 介绍的。Shewchuk[34,40] 把 CDTs 推广到三维或更高维。

下面, 我们给出两个约束 Delaunay 四面体化的等价定义。

有平面  $f \in X$  使得  $p$  和  $q$  位于这个平面的两侧, 平面包含  $f$ , 线段  $pq$  与这个切面相交 (参见图 7), 两个点  $p, q \in |X|$  相互是不可见的。判定四面体  $t$  顶点在  $X$  内是否符合约束 Delaunay (*constrained Delaunay*) 的条件是其外接球不包含  $X$  的点, 这些点对四面体  $t$  相对内部的任何点是可见的 (参见图 7 左图)。

判定一个四面体剖分  $T$  是  $X$  的约束 Delaunay 四面体剖分的条件是  $X$  的四面体剖分和  $T$  中每个四面体都是约束 Delaunay 规则。

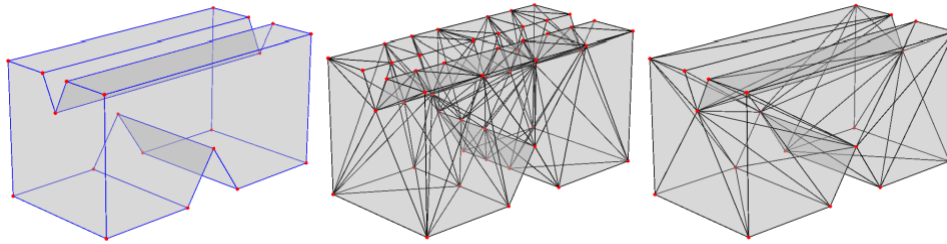


图 8: 对于一个简单的非凸多面体 (左图) 符合 Delaunay 四面体剖分 (中图) 和约束 Delaunay 四面体剖分 (右图) 的比较。

假定  $s$  是  $X$  中四面体网格  $T$  的一个三角形。如果  $s$  属于  $T$  的唯一的一个四面体, 或  $s$  恰好是两个四面体  $t_1$  和  $t_2$  的一个面, 并且  $s$  的外接球不包含  $t_1$  和  $t_2$  的顶点, 那么我们称  $s$  是局部 Delaunay (*locally Delaunay*), 也就是说,  $t_1$  的外接球不包含  $t_2$  中的顶点, 反之亦然 (参见图 7 右图)。要是  $T$  内的每个三角形都没包含在  $P$  的任意多边形内, 而且每个三角形都是局部 Delaunay, 则  $P$  的四面体网格剖分属于约束 Delaunay 四面体剖分 (CDT)。

Delaunay 四面体剖分和约束 Delaunay 四面体剖分的定义几乎是一样的。不同之处在于对 CDT 而言我们对边界多边形内的三角形去掉了局部 Delaunay 的要求。因此, 约束 Delaunay 四面体剖分保留了很多 Delaunay 四面体剖分的有利属性, 参考 [40,42]。注意约束 Delaunay 四面体剖分内的单纯形 (四面体, 三角形和边) 不一定符合是 Delaunay 规则的。

任意一个PLC  $X$ 的约束Delaunay四面体剖分(CDT)可能是不存在的, 参考[40]。Steiner点是必需的, 以确保CDT的存在。 $X$ 的一个Steiner约束四面体Delaunay剖分(CDT)是  $X \cup S$  的CDT, 这里  $S \subset |X|$  是一个Steiner点集。

与限定Delaunay四面体剖分相比, (Steiner)CDT生成的Steiner点通常会更少一点, 参见图8的例子。特别是当PLC包含一些锐化特征, PLC的几个单体会在一个很小的角度(或是二面角)相交时, 限定Delaunay四面体剖分是难以生成的, 而且可能需要大量的Steiner点。而CDT可以很容易地处理锐化特征。

#### 1.2.4 格网矢量,四面体形状判定

这里术语“网格质量”没有唯一的定义, 这是由应用程序和使用的数值方法决定的。作为一般的指导方针, 应该避免有非常小或非常大的角度(和二面角)的元素, 因为这通常会在数值方法的精度和性能上引发不良影响。

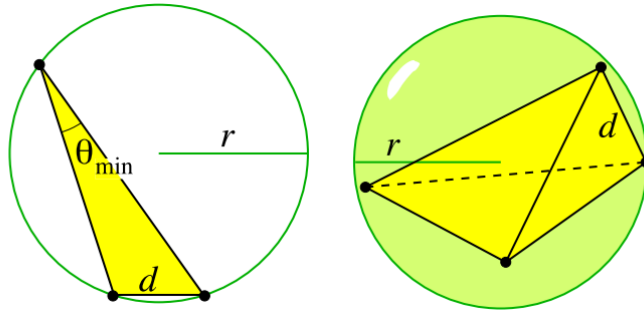


图 9: 三角和四面体的半径-边缘比

四面体形状判定(tetrahedron shape measure)是通过一个实数来评估四面体形状连续函数。目前已经提出不同的四面体形状判定方法, 有些是等价的, 参见[22,38]。

对于一个单纯形形状判定最普通的方法是使用长宽比。四面体  $T$  的长宽比  $\eta(T)$  是最长边的长度  $l_{max}$  与最小高度  $h_{min}$  之间的比值, 即  $\eta(T) = l_{max}/h_{min}$ 。当比值大于  $\sqrt{2}/\sqrt{3}$  时, 长宽比用来检测一个四面体的“圆形度”。长宽比越小意味着形状越好。长宽比也有其他类型的定义, 比如外接圆半径与内接圆半径的比值。如果四面体由一个比值限制意味着另一个也会被限制, 那么在这种意义下, 这些定义是等价的。

TetGen 并不是用长宽比, 而是选择面角(两条边的夹角)和二面角(两个面的夹角)作为四面体的形状判定。通过与 TetGen 中使用的 Delaunay 细化算法一起来发挥作用。而且, 它们的结合跟长宽比一样实现相同的目标。

四面体  $T$  的半径-边缘比(radius-edge ratio)  $\rho(T)$  是外接球半径  $r$  与四面体最短边的长度  $d$  的比值, 即:

$$\rho(T) = \frac{r}{d} \geq \frac{1}{2 \sin \theta_{\min}}$$

这里  $\theta_{\min}$  是四面体  $T$  最小的面角, 参见图 9。限制半径-边缘比等价于限制四面体最小的面角。半径-边缘比  $\rho(T)$  至少是  $\sqrt{6}/4 \approx 0.612$ , 这是由常规四面体剖分实现的。大多数形状不好的四面体会有一个比较大的半径-边缘比值(比如,  $>2.0$ ), 除了那种没有小边但是接近 0 体积的扁平四面体薄片。薄片有最小值  $\sqrt{2}/2 \approx 0.707$ 。因此半径-边缘比并不等同于长宽比(因为薄片)。除此之外, 半径-边缘比是一种有效的形状判定方法。如果四面体网格有一个限制所有



四面体的半径-边缘比，那么网格的点集则被完全分隔开，而且网格的每个节点都有限制度[45,42]。

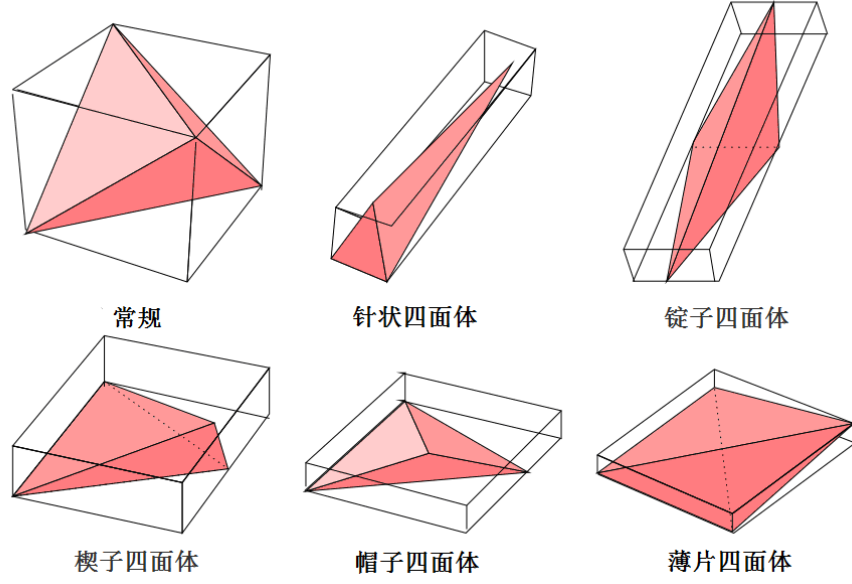


图 10: 由 Bern 等人在[3]中分类的不同类型的四面体

	常规	针状四面体	锭子四面体	楔子四面体	帽子四面体	薄片四面体
方向比	0.75	4.74	70.47	7.74	8.83	8.15
半径-边比	0.612	4.10	26.27	6.74	3.57	0.707
最小二面角	70.53°	60.12°	22.15°	5.77°	15.79°	4.98°

表 1: 长宽比( $\eta$ ), 半径-边缘比( $\rho$ ), 及其图 10 中显示的四面体最小二面角( $\phi_{\min}$ )

四面体 $T$ 六条边中的每一条都被两个面环绕，给定一个边两个面间的二面角(*dihedral angle*)就是两个面的交集和垂直于这条边的一个平面之间的角度。 $T$ 内的二面角范围是 $[0^\circ, 180^\circ]$ 。四面体 $T$ 最小的二面角 $\phi_{\min}(T)$ 是TetGen中使用的四面体形状判定方法。

图10展示了在[3]中分类的六种不同类型的四面体。每个四面体的三种形状判定值在表1中记录。

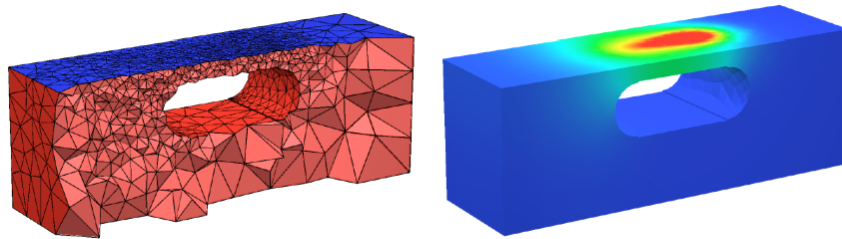


图 11: 一个自适应四面体网格(左图)和热传导问题的计算数值解决方案(右图)

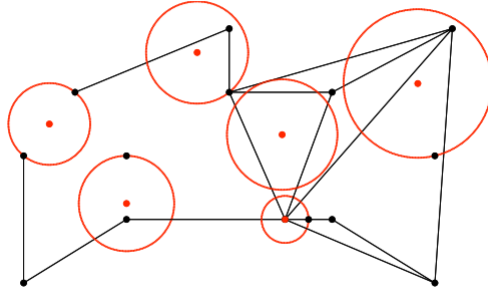


图 12: 区域内各点处的局部特征尺寸

### 1.2.5 网格尺寸, 网格自适应性

网格尺寸就是有网格元素的尺寸(如边长)决定的网格元素的总数。质量最佳的网格生成(或自适应网格生成)目标就是由少量网格元素来实现所需网格质量。在数值方法中, 这些网格很好的平衡了解决方案的时间和精确度。参见图11的例子。

这儿没有唯一的方式来预先决定网格尺寸。这是由实际的应用程序和使用的数值方法来确定的。作为自适应网格生成的常规指南, 我们假定网格尺寸函数(mesh sizing function)是可用的。设 $X$ 为一个三维PLC, 网格尺寸函数 $H: |X| \rightarrow \mathbb{R}$ 是使每个点 $p \in |X|$ 映射为一个正值 $H(p)$ , 用来指定点位置 $p$ 所需的边长。

在点 $p \in |X|$ 处的局部特征尺寸(local feature size)[32],  $\text{lfs}(p)$ 定义为以点 $p$ 为中心的最小球的半径, 其相交于两个没有拓扑关联的 $X$ 的单胞。 $\text{lfs}$ 基于其边界信息在 $|X|$ 上定义了一个默认的距离域。另外, 这是一个Lipschitz函数, 即 $\text{lfs}(x) \leq \text{lfs}(y) + \|x - y\|$ 。图12说明了在一个二维多边形区域内几个点的本地尺寸特征。

如果边长不会随着点 $p$ 的方向的变化而变化那就说明 $H$ 是各向同性的(isotropic), 否则就是各向异性的(anisotropic), 参见[17]。此时, TetGen仅仅使用各向同性的网格尺寸函数。一个理想的尺寸函数是 $C^\infty$ ,  $\forall p \in |X|$ 。然而大多数情况下,  $H$ 是由 $|X|$ 的某些点处指定的离散函数来逼近的。在 $|X|$ 上的其他点的尺寸是由插值法来获取的。

### 1.2.6 网格优化

为了进一步提高网格质量进行网格优化是必要的。通常用来做为网格生成后的后处理步骤。典型的就是它可以改善一个或几个网格质量上的目标函数。例如长宽比, 最小或最大的二面角, 等等。

### 1.2.7 算法

构造三维约束Delaunay四面体剖分的算法首先是由Shewchuk提出的。在[35]中, 他给出了PLC(或多面体)生成CDT的充分条件。如果PLC  $X$ 的边存在外接球, 那么边符合强Delaunay(strongly Delaunay)边的条件, 而且 $X$ 中没有其他顶点在外接球内或外接球上。判定三维PLC  $X$ 的每条边都是强Delaunay, 那么可以说是 $X$ 的边保护(edge-protected)的条件。任意边保护的PLC都有一个不含Steiner点的CDT。这个条件就意味着只能在PLC的边上添加Steiner点。在这个条件的基础上, 提出了几个构造Steiner CDT的算法[37,39,44,43]。

TetGen的CDT算法来自于Si和Gartner [44, 43]。这个算法在输入PLC  $X$ 的非Delaunay边上插入Steiner点, 得到一个新的PLC  $X'$ (包含 $X$ 的Steiner点)使得 $|X| = |X'|$ , 且 $X'$ 的所有边都是Delaunay边。用这样的方式选择Steiner点是为了消除不必要的短边被创建。使用简单象征性

的干扰是为了维护  $X'$  点集的常规位置。因此  $X'$  满足Shewchuk的条件。实用面插入算法被用于构造  $X'$  的CDT。

在生成三维PLC的符合Delaunay四面体剖分方面已经提出了多种算法[29,27,9,19]。TetGen使用改进的Delaunay细化算法来是一个CDT转变为限制Delaunay四面体剖分。描述如下。

生成高质量四面体网格的基本算法是Shewchuk[36]的Delaunay细化算法，这是三维中Ruppert算法[32]的概括。此算法反复添加一些恶劣四面体的外接球心，直到不存在这些四面体。这个算法生成高质量的Delaunay四面体，四面体的半径-边缘比没有超过2.0的(也就是说面角没有少于14.5°的)。一个短距离范围内对由小到大对四面体的尺寸进行分级。TetGen实现了这个算法提高了PLC的CDT网格质量。实际中，此算法生成的网格通常都会超过理论边界，且很有效的消除了含有过大或过小二面角的四面体。

基本Delaunay细化算法存在两个理论方面的问题。第一，因为使用半径-边缘比作为四面体形状判定的唯一方法，所以并没有消除薄片的存在。第二，此算法可能不会终止要是输入的PLC  $X$  包含锐化特征，也就是说， $X$  的两个边的夹角是锐角或是 $X$  的两个面的二面角是锐角。这些问题已经激发了一些薄片移除算法[7,21]和一些能处理锐化特征[8,28,31]的基本Delaunay细化算法的改进。

TetGen使用四面体的最小二面角作为Delaunay细化算法的第二个形状判定方法。因此通过这个算法能发现那些薄片，并通过相同的Delaunay细化迭代次数来移除薄片。因为TetGen进行约束Delaunay四面体剖分(CDT)时，它可以提前检测到CDT中所有的锐化特征然后开始Delaunay细化过程。具有锐化特征接近锐化的四面体是永远不会被移除的。TetGen中改进的算法结束这种现象。当一些畸形四面体接近锐化时可能还会幸存下来。

TetGen使用网格尺寸函数来控制网格尺寸。若没给出网格尺寸函数，TetGen使用默认的局部特征尺寸(由输入PLC定义)。它是通过基本的Delaunay细化算法[32,36]来隐式调用的。如果给出了该函数，TetGen会根据输入的网格尺寸函数生成一个自适应四面体网格，这个算法是改进的Delaunay细化算法，在[41]中有所描述。

## 2 入门

TetGen源码使用C++分布式编写的。最新版本在<http://www.tetgen.org>官方网站获得。当前版本包含了25000行源代码和7000行注释行。

TetGen仅仅使用了标准C++库。它可以在所有的32位和64位计算机上，使用流行的C++编译器上编译。章节2.1说明了在Unix/Linux/MacOSX和Windows系统下分别使用GNU C++编译器和Microsoft C++编译器，将如何编译TetGen为可执行程序或者一个库。

TetGen编译成功之后，得到可执行文件“tetgen”(或者window下“tetgen.exe”)。你可以使用在章节2.2的教程中的示例文件开始测试TetGen。

TetGen没有图形用户界面接口(GUI)。TetView程序可用于TetGen输入和输出的可视化。其它流行的格网视景软件支持TetGen输入输出的可视化，见章节2.3。

### 2.1 编译

下载到的文档包括下面文件

README            常规信息

LICENSE           版权条款声明



tetgen.h	TetGen头文件
tetgen.cxx	TetGenC++源文件
predicates.cxx	几何判断C++源文件
makefile	编译TetGen Makefile文件
example.poly	示例文件

predicates.cxx 是 Shewchuk 使用 C++ 编写的，较为健壮的几何判断程序，可参照 <http://www.cs.cmu.edu/~quake/robust.html>。

编译 TetGen 时，不同系统使用不同的编译器。如 GNU's g++，或者 MS Windows 系统下的 Microsoft C++。TetGen 可以编译为可执行程序，或者提供为其他程序使用的库文件。

### 2.1.1 Unix/Linux/MacOSX

编辑和使用“makefile”文件能够较为容易的编译 TetGen。在编译之前，输入所有源文件(tetgen.h, tetgen.cxx, and predicates.cxx)和在同一目录下的makefile文件(通常情况他们是在一个目录下面的)。阅读makefile文件可以对一些选项做出相应的修改。

应该至少指定一款 C++ 编译器和优化级别。GNU C++ 编译器默认是使用的，并且它没有使用优化。

当使用“make”方式编译 TetGen 为一个可执行程序或者使用“make tetlib”方式编译 TetGen 为一个库文件，可执行文件“tetgen”或者库文件“tetgen.a”将出现与makefile文件相同的目录下。

除了上述方法之外，也可以不使用makefile文件编译 TetGen。假设使用 g++，首先编译文件 predicates.cxx 为目标文件：

```
g++ -c predicates.cxx
```

编译 TetGen 为可执行文件使用下面的命令：

```
g++ -o tetgen tetgen.cxx predicates.o -lm
```

编译 TetGen 为一个库文件时，需要指定 TETLIBRARY 符号：

```
g++ -DTETLIBRARY -c tetgen.cxx  
ar r libtet.a tetgen.o predicates.o
```

一个额外的设置可以得到较为高效的 TetGen 版本：

- 使用 C++ 编译器的最优化选项。通常使用 TetGen 优化的版本比没有优化版本快两倍。
- TetGen 源码包含一些声明。使用它们的目的是为了捕获程序的 bugs，即方便调色程序而生。但是它们的出现降低了 TetGen 程序的性能。你可以通过在编译命令行中添加“-DNDEBUG”符号废弃与调试相关的代码。

下面为使用 GNU C++ 编译器生成一个简单的 TetGen 优化版本的示例：

```
g++ -O3 -DNDEBUG -c predicates.cxx  
g++ -O3 -DNDEBUG -o tetgen tetgen.cxx predicates.o -lm
```

### 2.1.2 Windows9.x/NT/2000/XP

在 Win32 系统中，TetGen 将编译为控制台程序“tetgen.exe”或者静态库文件“tetgen.lib”。使用 Microsoft Visual C++ 6.0 (VC++)，测试通过。使用 VC++ 的 IDE (Integrated Development Environment，集成开发环境) 是一种方便编译 TetGen 的方式，创建“tetgen.exe”最简单的步

骤:

- 创建一个“Win32 console application”，命名为“tetgen”，
- 添加源码文件(tetgen.h, tetgen.cxx, 和 predicates.cxx)到工程，
- 生成工程。

创建库文件简单步骤:

- 创建一个“Win32 static library”，命名为“library”，
- 添加所有源码到工程中，
- 添加预编译宏“TETLIBRARY”，
- 生成工程

### Shewchuk健壮几何判断程序说明

TetGen使用Shewchuk健壮的浮点算术几何判断程序(“predicates.cxx”)。该算术是基于IEEE 74浮点数标准。然而一些处理器可能默认不使用这些浮点表达和算术的标准。如果是这样，需要配置，以正确的支持这些几何判断。见<http://www.cs.cmu.edu/~quake/robust.pc.html>的说明。

- 在构建Delaunay四面体时，如果TetGen出现“a segmentation fault”错误，大多数可能是没有正确配置几何判断程序。
- 在我特有的测试中(原文作者)，使用gcc/g++ 版本4.4在MacOSX系统酷睿2双核的处理器计算机上测试，这里不需要配置几何判断程序。使用了默认的几何判断程序配置。TetGen正确运行debug版本和优化版本中。使用gcc/g++版本4.6在一台Linux(Ubuntu)系统，Inter奔腾2.8GHz处理的计算机上得到相同的结果。
- 当我使用gcc/g++版本4.2在Linux系统Inter Xeon CPU处理器的计算机上，TetGen遭遇上述的错误。TetGen使用优化选项“-O3”编译。然而，当TetGen编译debug版本是使用“-g”选项，它可以正确运行。

## 2.2 测试

如果命令行参数为空，那么TetGen将打印出命令行选项列表(即，仅“tetgen”类型参数)。TetGen使用-h选项将打印出使用方法的简单描述:

**tetgen -h**

附加文件example.poly是一个简单的三维格网文件。尝试在TetGen中的使用:

**tetgen -p example**

使用“-p”选项，TetGen将读文件，即example.poly，并且生成图(右)中约束Delaunay四面体剖分。结果格网保存在三个文件中example.1.node, example.1.ele, 和example.1.face，分别保存了节点列表，四面体列表和边界面列表(注意，不是格网所有的面)。TetGen并将输出如下的报告信息:

```
Opening  example.poly.
Delaunizing  vertices...
Delaunay  seconds:    0.003834
Creating  surface  mesh  ...
Surface  mesh  seconds:    0.000839
Constrained  Delaunay...
```

```

Boundary recovery seconds: 0.000728
Removing exterior tetrahedra ...
Exterior tets removal seconds: 0.000214
Writing example.1.node.
Writing example.1.ele.
Writing example.1.face.
Writing example.1.edge.
Output seconds: 0.000875
Total running seconds: 0.006611
Statistics:
Input points: 56
Input facets: 30
Input segments: 82
Input holes: 0
Input regions: 0
Mesh points: 57
Mesh tetrahedra: 142
Mesh faces: 337
Mesh edges: 251
Mesh boundary faces: 106
Mesh boundary edges: 83
Steiner points in boundary edges: 1

```

上面的格网是相当粗糙的，并包含了许多恶劣形状的四面体(如：狭长的四面体)。现在做如下尝试：

‘-q’ 选项引起格网细化，一些新点添加到将移除的恶劣格网单元中。生成格网与上文相同包含三个文件。但是现在是一个优良的四面体格网，每个四面体的外接圆半径到四面体最短边的比率不超过2.0（TetGen默认值）。现在在做如下测试：

### **tetgen -pq1.2V example**

TetGen将重新生成一个优质格网（见图中间），它比先前的格网包含更多的点，所有的外接圆半径和边的比率不超过1.2，即，四面体图元比之前的格网中的图元更优良。另外，TetGen输出格网质量报告（-V选项作用），如下：

Mesh quality statistics:

Smallest volume:	0.0022168		Largest volume:	12.084
Shortest edge:	0.28004		Longest edge:	6.129
Smallest aspect ratio:	1.2658		Largest aspect ratio:	11.326
Smallest facangle:	25.194		Largest facangle:	127.1048
Smallest dihedral:	8.447		Largest dihedral:	164.7104

Aspect ratio histogram:

< 1.5	:	118		6 - 10	:	60
1.5 - 2	:	1537		10 - 15	:	6
2 - 2.5	:	1844		15 - 25	:	0
2.5 - 3	:	877		25 - 50	:	0

3 - 4	:	467		50 - 100	:	0
4 - 6	:	171		100 -	:	0
(A tetrahedron's aspect ratio is its longest edge length divided by its smallest side height)						
Face angle histogram:						
0 - 10 degrees:		0		90 - 100 degrees:		1598
10 - 20 degrees:		0		100 - 110 degrees:		490
20 - 30 degrees:		517		110 - 120 degrees:		190
30 - 40 degrees:		3190		120 - 130 degrees:		25
40 - 50 degrees:		4801		130 - 140 degrees:		0
50 - 60 degrees:		2315		140 - 150 degrees:		0
60 - 70 degrees:		2416		150 - 160 degrees:		0
70 - 80 degrees:		3702		160 - 170 degrees:		0
80 - 90 degrees:		2402		170 - 180 degrees:		0
Minimum input face angle is 81 (degree).						
Dihedral angle histogram:						
0 - 5 degrees:		0		80 - 110 degrees:		3331
5 - 10 degrees:		11		110 - 120 degrees:		678
10 - 20 degrees:		154		120 - 130 degrees:		456
20 - 30 degrees:		445		130 - 140 degrees:		250
30 - 40 degrees:		1243		140 - 150 degrees:		143
40 - 50 degrees:		1851		150 - 160 degrees:		85
50 - 60 degrees:		1170		160 - 170 degrees:		32
60 - 70 degrees:		206		170 - 175 degrees:		0
70 - 80 degrees:		105		175 - 180 degrees:		0
Minimum input facet dihedral angle is 90 (degree).						

替换 ‘-q’ 选项，我们使用 ‘-a’ 选项输入最大体积构建获得一个优质的格网。这样做，所得到的格网中没有任何四面体的体积大于该设定值。测试运行如下命令：

### **tetgen -pq1.2Va1 example**

现在得到的格网包含的顶点比之前的格网数量更多。除了 ‘-q’ 和 ‘-a’ 选项之外，TetGen 拥有其他控制格网大小和形状的选项。它们将第3章中讲述。

所有的格网和Delaunay四面体剖分可以借助下一章节介绍的程序显示。

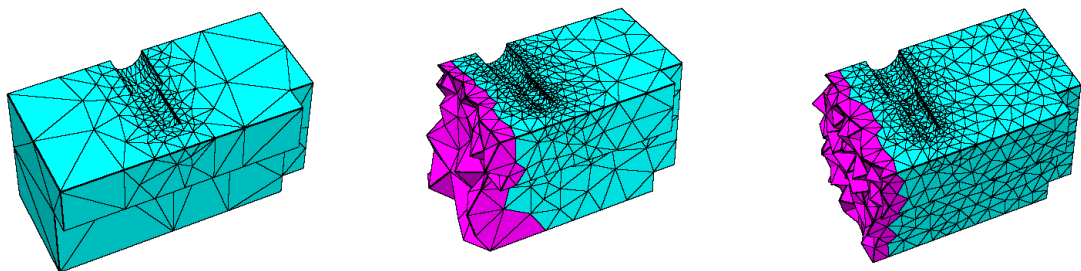
命令行选项和文件格式的详细描述将在第三章和第四章中阐述。

计算PLC点集的Delaunay四面体剖分和凸包，如下：

**cp example.poly example.node**

**tetgen example**

Delaunay 四面体剖分保存在 example.1.node 和 example.1.ele 中。凸包通过文件 example.1.face中的三角形表达。



图（左）命令tetgen -pq example;（中）命令tetgen -pq1.2V example;（右）命令tetgen -pq1.2Val example。

## 2.3 可视化

### 2.3.1 TetView

TetView是显示PLC（Piecewise Linear Complexes）和简单格网的图形界面程序。它可以读TetGen的输入和输出文件以及显示TetGen的对象，见图13快照。他可以显示其他信息，如：边界类型和材质。交互式接口提供用户操作（如：旋转，平移，放大/缩小，剪切，收缩等），方便模型使用鼠标和键盘操作。TetView可以保存当前窗口内容为高质量的被封装的PostScript格式文件。

TetView可在<http://www.tetgen.org/tetview.html>免费获得。可以找到在不同平台下的可执行程序。下载符合当前你的系统的可执行程序。

为了显示文件example.poly文件中的PLC，首先拷贝可执行文件到example.poly文件目录下，加载运行：

**tetview example.poly**

下面命令行将显示格网(在文件example.1.node,example.1.ele,和example.1.face中的格网):

**tetview example.1.ele**

在相关网站上可以使用TetView的使用教程。

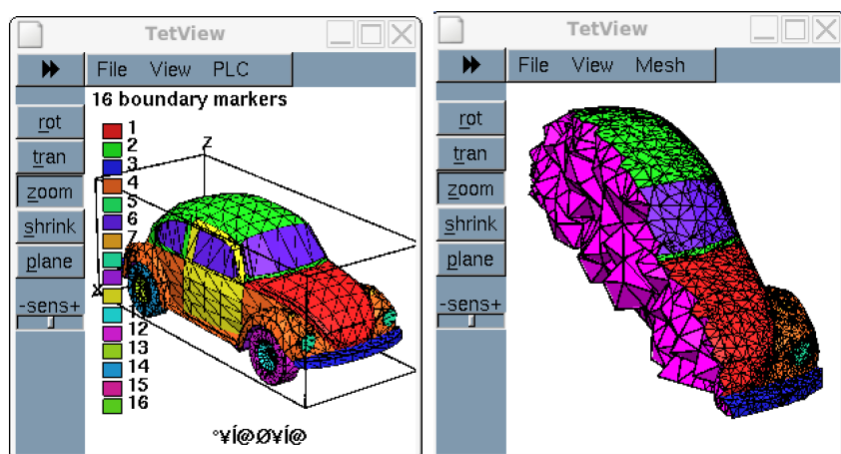


图13 TetView界面快照。左边：显示一个汽车表面格网。右边：显示一个汽车的四面体格网

### 2.3.2 Medit

TetGen可以导出四面体格网为Medit软件支持的.mesh文件格式。借助软件Medit展示格网，Medit可以在<http://www.ann.jussieu.fr/~frey/logiciels>中免费获得。

为了在Medit中显示格网，在使用TetGen生成格网时，命令行中添加‘-g’选项。TetGen将在TetGen软件目录中额外输出example.1.mesh文件。测试运行命令：

```
tetgen -pg example.poly
medit example.1
```

### 3 使用 TetGen

本章描述了如何使用独立的TetGen程序。它通过命令行设置程序中选项和输入文件名。选项用于控制TetGen的行为和指明输出文件。对于不同的选项，TetGen将生成Delaunay四面体格网或者约束Delaunay四面体格网或者自适应Delaunay格网等。

#### 3.1 命令行语法

```
tetgen [-pYq_Aa_mriO_S_T_XMwcdzfenvgKJBNEFICQVh] input_file
```

下划线表示某些选项后的这个数字可以是任意的。选项和数字参数之间不能有空格。这些开关的解释见章节3.2。

输入文件可以是不同格式的文件，这取决于你使用的选项。如果没有任何选项使用，那么输入文件中必须包含一个包含三维点列表的.node文件，使用这些点集生成Delaunay四面体格网。

如果使用-p选项，输入文件必须包括下面的格式的文件.poly,.smesh,.off,.stl和.mesh。这些文件都包含PLC(piecewise linear complex，一个表面格网)，使用这些对象以生成边界约束的Delaunay四面体分割。如果与-q开关联合使用，将生成自适应四面体格网。

如果使用了-r选项，现存的四面体格网将被读取，你必须提供.node文件和.ele文件，而.face文件、.edge和.vol文件可以不提供。输入文件没有后缀。联合-q开关，这个格网将遵循新的约束以改善格网质量。

在第四章中对文件格式做出了详细的介绍。

#### 3.2 命令行开关

表2是命令行参数列表和每个参数简短的说明。这个信息也可以通过调用TetGen程序块获得外部参数和输入文件。在下面小段详细的介绍了每个选项。

- p 使用一个PLC进行四面体分割。
- r 重建已有四面体格网。
- q 改善格网(改变格网的质量)。
- Y 保持输入表面格网(不修改它)。
- m 应用用户自定义格网大小功能。
- a 应用最大四面体体积约束。
- A 在不同范围内四面体指定属性。
- I 插入附加点到格网中。
- O 指定格网优化级别。
- S 指定添加点的最大数量。
- T 设置一个共面容差(默认为1e-8)。

- M 不合并共面的面。
- X 支持使用精准算法。
- w 生成权重(规律的)Delaunay四面体格网。
- d 检测PLC中面自身是否交叉。
- z 所有的输出项编号从零开始。
- f 输出所有面到.face文件。
- e 输出所有的边到.edge文件。
- n 输出四面体的邻接关系到.neigh文件。
- v 输出Voronoi图到文件。
- g 输出格网到Medit的.mesh文件中。
- K 输出格网到Paraview的.vtk文件。
- J 不丢弃未使用点。
- B 不输出边界信息。
- N 不输出.node文件。
- E 不输出.ele文件。
- F 不输出.face文件。
- I 不输出格网的迭代次数。
- C 检测格网的相容性。
- Q 不打印除错误以外的额外信息。
- V 打印详细信息。
- h TetGen简要帮助信息。

表2 TetGen命令行参数一览表

### 3.2.1 四面体格网化一个点集(-w, -v)

给定三维点集或者权重点集，TetGen将生成这些点集的Delaunay四面体格网或者权重Delaunay四面体格网。Voronoi图或者权重图通过它的对偶获得。

三维点集保存在.node文件中，使用命令行运行命令：

```
tetgen test.node
```

这个命令生成点集的Delaunay四面体格网(DT)。下面是TetGen输出信息：

```
Opening  test.node.
Delaunizing  vertices...
Delaunay  seconds:    0.003922

Writing  test.1.node.
Writing  test.1.ele.
Writing  test.1.face.

Output  seconds:    0.000982
Total  running  seconds:    0.004995
Statistics:

Input  points:  56
```

```

Mesh points: 56
Mesh tetrahedra: 233
Mesh faces: 491
Mesh edges: 313
Convex hull faces: 50
Convex hull edges: 75

```

图14显示了一个示例，输入的点集(100个顶点)和生成的DT和它的外包围盒。  
表3中列出了TetGen默认输出的三个文件。

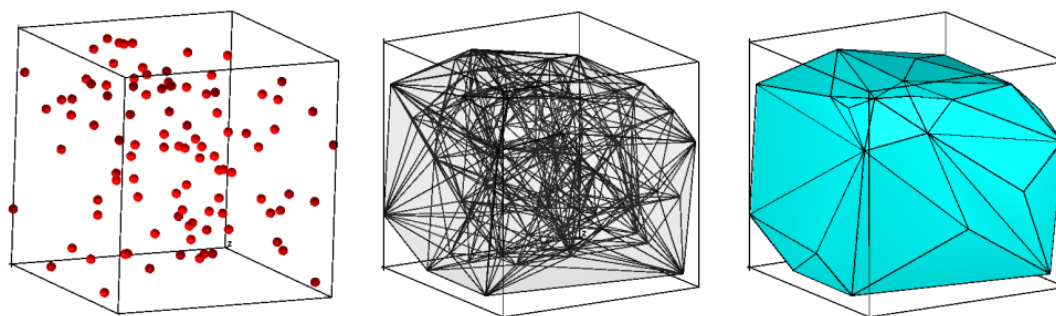


图14 从左到右：在单位立方体中的100个随机点集，Delaunay四面体格网和点集的外包围盒。

test.1.node	DT的顶点列表(与输入相同)
test.1.ele	DT的四面体列表
test.1.face	点集的外包面

表3 TetGen默认输出文件

DT表面和边可以通过添加输出参数-f(输出所有面)和-e(输出所有边)获得。例如：下面命令。

```
tetgen -fe test.node
```

TetGen输出4个文件，如下表4:

test.1.node	DT的顶点列表(与输入相同)
test.1.ele	DT的四面体列表
test.1.face	点集的外包面，包围盒上的面标识为1，内部面为0
test.1.edge	DT的边列表。外包边界有边界标识1，内部边界有标识0

表4 命令tetgen -fe test.node输出的文件

四面体列表的邻接关系可以通过添加-n命令参数获得。TetGen将输出一个附加文件test.1.neigh。neigh文件格式见第四章说明。

使用-w参数将由三维权重点生成权重Delaunay四面体格网。一个权重点定义为 $P' = \{p_x, p_y, p_z, p_x^2 + p_y^2 + p_z^2 - w\} \in R^4$ ，其中w是点 $P = \{p_x, p_y, p_z\} \in R^3$ 权重值(实数)。



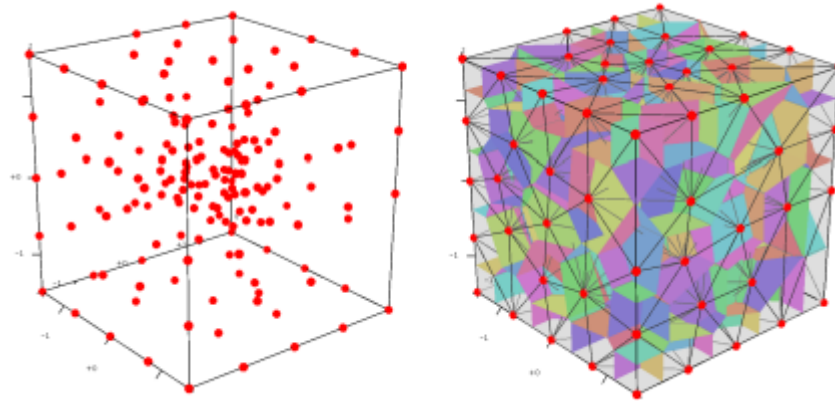


图15 左图：单位立方体内的164个随机点集。右图：Delaunay四面体格网(黑边显示)和Voronoi图(彩色面表示)。

保存权重点集到.node文件中，在.node文件中的这些点集中必须包括一个属性值，并且每个点的第一个属性是它的权重。由这些点生成权重DT，输入下面命令运行TetGen：

```
tetgen -w test.node
```

权重Delaunay四面体格网和它的包围盒保存在与表3同样的文件中。test.1.node文件中的一些点可能没有被任何四面体所使用。

Voronoi图或者点的权重图是通过通过生成Delaunay和权重Delaunay四面体格网的对偶获得。见图15示例。

添加-v参数命令，TetGen输出Voronoi图或者权重图，四个文件如下表5：

test.1.v.node	Voronoi顶点列表(垂心)
test.1.v.edge	Voronoi边列表
test.1.v.face	Voronoi面列表
test.1.v.cell	Voronoi胞列表

表5 Voronoi图或者权重图输出文件

\*.v.node文件与.node文件格式相同。\*.v.edge，\*.v.face和\*.v.cell文件格式在第四章有详细的描述。

注意-w和-v参数仅仅适用于点集情况。

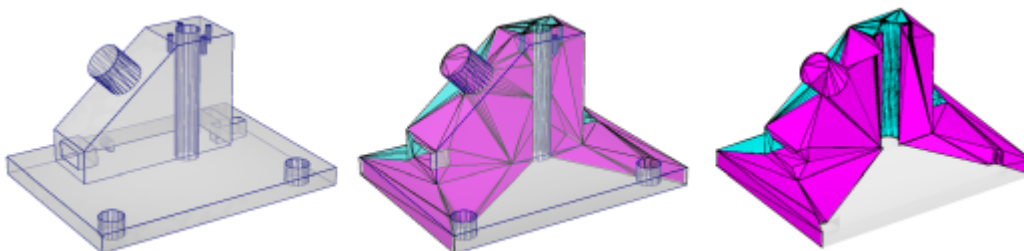


图16 输入PLC(camila.poly,左图),由位于PLC边界上的Steiner点集生成的Steiner CDT(中间图,使用'-p'选项)。

和由位于PLC内部的Steiner点集生成的约束四面体格网(右图,使用'-pY'选项)。

### 3.2.2 分段线性复合体(PLC)四面体格网化(-p, -Y)

使用-p选项将读取保存在.poly和.smesh文件中的PLC信息，以生成PLC四面体格网。

TetGen默认生成一个PLC约束的Delaunay四面体格网(CDT)。CDT可能包括不是PLC顶点的Steiner点。此外，Steiner点位于PLC的边界边和面上,见图16(中)。

如果-Y选项与-p选项一起使用。Steiner点(如果存在的话)仅仅在PLC的内部空间出现，见图16(右)。当一定要保存格网边界以致它符合一些邻接的格网时，-Y选项就会很有用处。用test.poly文件来描述一个三维PLC。运行下面的命令行：

**tetgen -p test.poly**

这将会产生PLC的约束Delaunay四面体网格，下面是TetGen的屏幕输出：

```
Opening test.poly.
Delaunizing vertices...
Delaunay seconds: 0.004033
Creating surface mesh ...
Surface mesh seconds: 0.000832
Constrained Delaunay...
Boundary recovery seconds: 0.000765
Removing exterior tetrahedra ...
Exterior tets removal seconds: 0.000221
Writing test.1.node.
Writing test.1.ele.
Writing test.1.face.
Writing test.1.edge.
Output seconds: 0.001051
Total running seconds: 0.007055
Statistics:
Input points: 56
Input facets: 30
Input segments: 82
Input holes: 0
Input regions: 0
Mesh points: 57
Mesh tetrahedra: 142
Mesh faces: 337
Mesh edges: 251
Mesh boundary faces: 106
Mesh boundary edges: 83
Steiner points on boundary edges: 1
```

TetGen默认的输出就是表6列出的四个文件：

test.1.node	CDT的顶点列表(包括Steiner点)
test.1.ele	CDT的四面体列表
test.1.face	CDT的边界面列表
test.1.edge	CDT的边界边列表

表6: 命令tetgen -p test.poly的输出文件

通过添加命令其他可用的输出选项：'-f'(输出所有面)，'-e'(输出所有边)，'-n'(输出四面

体的邻接图形)

- 如果没有指定文件扩展名, TetGen会自动地寻找有扩展名.poly或.smesh的文件并且使用其中的任意一个。例如, “tengen -p xxx” 打开xxx.poly的文件(如果该文件不存在, 就会打开xxx.smesh), 也可能打开xxx.node文件和xxx.edge(包含了一系列不属于PLC任何面的孤立的边); 在整个PLC中读取数据; 生成一个CDT结果是三个文件: xxx.1.node, xxx.1.ele与xxx.1.face。
- 其他多边形文件格式也可以用作输入文件。TetGen通过文件扩展名来识别文件格式。到目前为止, 支持这几种文件格式: .off, .ply, .stl, .vtk, .mesh。
- TetGen与-q命令的组合使用, 将会生成PLC的一个高质量的四面体网格。要是-Y命令与-q命令一起使用(保存了边界信息并且提高了网格质量), 产生的四面体质量主要受边界网格质量的影响。
- -p命令与-r命令不兼容, 不应该与其他命令一起使用。

### 3.2.3 生成优良格网 (-q)

-q命令添加新点来提高网格质量。它能跟'-p'一起使用(细化CDT), 或与'-r'一起使用(细化之前生成的网格), 或与'-m'一起使用(以符合网格尺寸的函数)。

TetGen执行四面体上的两个质量约束条件: 最大的半径-边缘比约束和最小的二面角约束。默认情况下, 这两个约束条件的值分别为2.0和5°。

供选择的质量约束条件可以在命令'-q'后指定。两个约束条件用斜线 '/' 分割开:

- 第一个约束条件是允许的半径-边缘比的最大值;
- 第二个约束条件是任意一个四面体允许的二面角的最小值。例如, '-q1.2'表示半径-边缘比的最大值为1.2; '-q1.2/10'表示半径-边缘比的最大值为1.2的同时二面角的最小值为10°; '-q/7'表示半径-边缘比使用默认值2.0, 但是二面角限值为7°。'-q/'表示采用默认半径-边缘比界限值2.0和默认二面角限值5°。

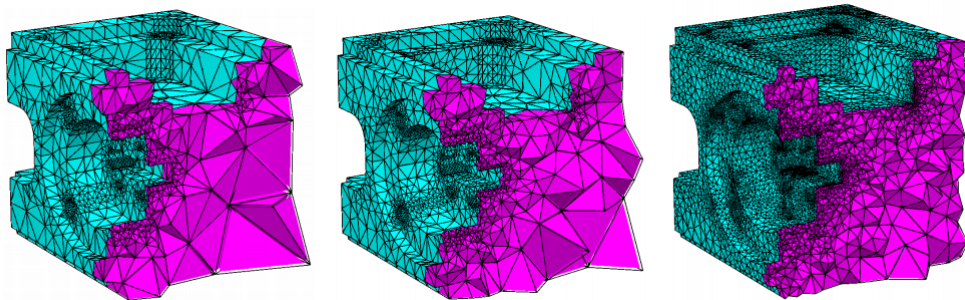


图17: 由命令-pq2/0, -pq1.4/0, 和-pq1.1/0生成的PLC(the part.smesh)的高质量四面体网格

图17说明了通过应用不同的半径-边缘比限值生成了PLC的三个高质量四面体网格。

默认的输出文件(四个文件)与表6列举的一样。

- 在命令行添加'-V'命令, TetGen会在屏幕上打印生成的四面体网格的一份质量报告(包括高宽比, 半径-边缘比, 二面角等信息)。

```
Mesh quality statistics:
Smallest volume: 0.016741 | Largest volume: 125.77
Shortest edge: 0.30902 | Longest edge: 12.189
Smallest asp.ratio: 1.2927 | Largest asp.ratio: 16.964
Smallest facangle: 15.352 | Largest facangle: 141.8279
Smallest dihedral: 5.587 | Largest dihedral: 163.9413
Aspect ratio histogram:
< 1.5 : 5 | 6 - 10 : 33
1.5 - 2 : 105 | 10 - 15 : 4
2 - 2.5 : 228 | 15 - 25 : 1
2.5 - 3 : 215 | 25 - 50 : 0
3 - 4 : 321 | 50 - 100 : 0
4 - 6 : 150 | 100 - : 0
```

(A tetrahedron's aspect ratio is its longest edge length divided by its smallest side height)

Face angle histogram:

0 - 10 degrees: 0	90 - 100 degrees: 637
10 - 20 degrees: 122	100 - 110 degrees: 131
20 - 30 degrees: 556	110 - 120 degrees: 101
30 - 40 degrees: 700	120 - 130 degrees: 44
40 - 50 degrees: 1273	130 - 140 degrees: 5
50 - 60 degrees: 1085	140 - 150 degrees: 1
60 - 70 degrees: 1129	150 - 160 degrees: 0
70 - 80 degrees: 871	160 - 170 degrees: 0
80 - 90 degrees: 506	170 - 180 degrees: 0

Minimum input face angle is 81 (degree).

Dihedral angle histogram:

0 - 5 degrees: 0	80 - 110 degrees: 1675
5 - 10 degrees: 10	110 - 120 degrees: 228
10 - 20 degrees: 141	120 - 130 degrees: 149
20 - 30 degrees: 362	130 - 140 degrees: 92
30 - 40 degrees: 487	140 - 150 degrees: 77
40 - 50 degrees: 762	150 - 160 degrees: 32
50 - 60 degrees: 770	160 - 170 degrees: 7
60 - 70 degrees: 812	170 - 175 degrees: 0
70 - 80 degrees: 768	175 - 180 degrees: 0

Minimum input dihedral angle is 90 (degree).

- 如果在PLC中没有表现出锐化特征，则TetGen中使用的Delaunay细化算法可以确保终止掉半径-边缘比不小于2.0的四面体，其中二面角的最小值没有界限。在实践中，该算法应用性更好，比如，在半径-边缘比为1.2最小二面角为18°的情况下该算法通常都能成功。
- 如果在PLC中表现出锐化特征，TetGen则会确保大多数四面体需要的质量约束条件，但是在最终的网格中会留下一些质量差的四面体。通常，它们会接近锐化特征。

### 3.2.4 分配范围属性(-A)

对于每个能识别出属于哪个有界面区域的四面体，-A命令能为其分配额外的属性(一个整数)，在输出网格中，同一区域的所有的四面体都会给出一个非零属性值。

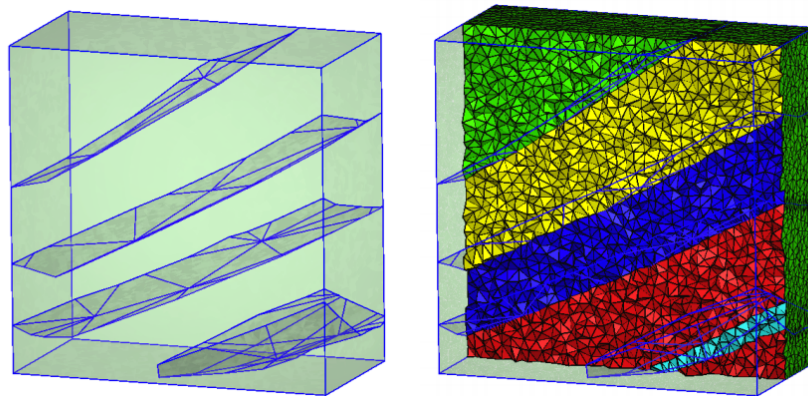


图18: 由命令: -pqAale-12生成的一个PLC(ts80305\_nd32\_cell834.off)的四面体网格

图18显示了包含有几个子域的PLC四面体网格的例子。

- 区域属性是一个整数值，或正或负。但一定不能为0。它用于PLC的外部。
- 用户定义的属性通过.poly或.smesh文件(在区域部分)分配至区域。如果一个区域没有被.poly或.smesh文件来明显标注，则那个区域的四面体会自动分配一个非零属性。
- 默认的，区域属性编号为1,2,3,...如果有用户分配的区域属性(通过.poly或.smesh文件)，区域属性编号就会变为i，即i+1,i+2,i+3,...,i或者为0，或者为用户分配的区域属性最大整

数值。

- 只有当使用-p命令但不使用-r命令时，-A命令才会起作用。

### 3.2.5 体积约束(-a)

-a命令使用在网格细化中(与-q选项一起使用)。它在所有四面体上强加了一个最大体积约束条件。如果'-a'后紧跟一个数值，则就不会生产体积大于这个数值的四面体。图18为一个例子。

- 我们可以通过调用两次-a命令强加一个固定和一个变化的体积约束条件。前者后面跟一个整数值，后者不用。每个指定的体积可能都包含一个小数点。
- 如果没有指定数值但是使用了-r命令，则需要一个.vol文件，它对于每个四面体都包含不同的体积约束条件。要基于后验误差估计来细化有限元或有限体积网格是.vol文件的用处很大。

除了最大的体积约束，TegGen也支持其他的约束条件如分别强加在面片和线段上的最大面积约束和最大边长约束。这些约束条件都是通过使用一个'.var'文件来施加的。

参见'-m'选项以了解如何为了网格细化应用一个网格尺寸函数。

### 3.2.6 应用网格尺寸函数(-m)

-m命令在网格细化中使用(也就是，与'-q'选项一起使用)。-m命令应用用户定义的网格尺寸函数，该函数指定了最终网格中的期望边长。目的在与创建一个边长符合这个函数的自适应网格。目前，仅支持各向同性的网格尺寸函数。

TetGen假定网格尺寸函数被指定在一个离散点集上，该离散点集的凸包覆盖了网格区域(也就是，PLC的基础空间)。区域内任何一点的网格尺寸都会由它的邻接点线性插值来自动计算。

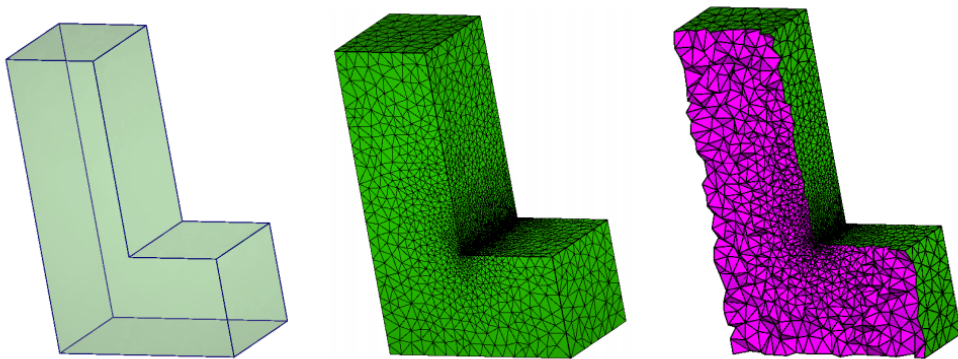


图19: 由命令: -pqm生成的PLC(L.mesh)的四面体网格。在PLC的节点上应用了一个尺寸函数(L.mtr)

当使用'-m'选项时，TetGen会读取一个存储着节点网格尺寸的.mtr文件。有两种方式来指定尺寸函数：

- 网格尺寸定义在输入PLC(使用-p命令)的节点上或输入网格(使用-r命令)的节点上。这种情况下，文件名为xxx.mtr，其中xxx是输入PLC或输入网格的基本文件名，见图19的例子。
- 网格尺寸定义在背景网格的节点上。此情况下，背景网格通过文件xxx.b.node，xxx.b.ele和网格尺寸文件xxx.b.mtr给出。

### 3.2.7 重建四面体网格(-r)

-r命令重构一个存在的四面体网格。通常，使用这个命令的目的是细化网格以提高其质量，也就是，跟-q命令一起使用。-r命令也可能用以其他用处，比如插入额外的点(-i命令)，网格自适应性(-m命令)以及线性函数插值(-m命令+一个背景网格)。



- 四面体网格从.node和.ele文件读取。一定要提供这两个文件。
- 如果存在一个.face文件，TetGen会读取并使用它来寻找四面体网格中边界面。注意：只有非零边界标注的面才会当作是边界面。不论哪种情况，TetGen都会自动识别网格区域外部的面，并视为边界面。边界面内部通过比较两个邻接四面体的属性也会被识别。
- 如果存在.edge文件，TetGen会读取并使用它来寻找网格中的边界边。注意：只有非零边界标识的边才会当作边界边。TetGen也将会从已被识别的边界面中识别出边界边。
- 重建后的网格以不同的迭代次数在初始网格中区别开来。例如，“tetgen -r xxx.1”从文件xxx.1.node, xxx.1.ele及xxx.1.face和xxx.1.edge(如果存在的话)读取网格；重建网格；把它输出到三个文件中xxx.2.node, xxx.2.ele, xxx.2.face。现在，在上述命令中xxx.2可被用作输入，结果就是另外一个网格保存文件xxx.3.node等。网格迭代次数允许您你创建一个连续细分格网的序列。
- -r不能与-p和-I一起使用。

### 3.2.8 插入附加点列表(-i)

-i命令表示在已存在的四面体网格中插入一系列附加点。附加的节点列表从文件xxx.a.node中读取，xxx代表了输入文件的名称(即xxx.poly或xxx.mesh或是xxx.ele,...)。此命令在使用用户定义的点列表来细化有限元或有限体积网格时是很有用处的。

- 位于网格区域的外部点会被丢弃。
- TetGen使用相对容差(通过-T设置)来检测一个点是否位于区域边界上。

### 3.2.9 格网优化(-O)

-O选项指定网格优化的级别，选择操作。两者都是整数并由斜线’/’分开。

网格优化级别是由整数0至10划分的，0表示没有实行网格优化。级别越大，将会执行越多的网格优化迭代次数，而且TetGen可能会运行很慢。默认的网格优化级别是2。

为了优化网格TetGen提供了三种可用的局部操作，分别是：

- 边/面转换；
- 顶点平滑化；
- 点的插入/删除。

选择操作的整数范围是从0至7。0表示没有选择操作(因此没有进行网格优化)。每个操作都是通过设置这个整数值对应的位元来开启/禁止的。

- 第一个位元(最低有效位元)开启或禁止边/面的转换；
- 第二个位元开启禁止顶点平滑化；
- 第三个位元开启禁止顶点的插入或删除；

默认情况下，这个整数值是7，也就是这三项操作都是开启的。

例如，选项 ‘-O2/7’ 表示优化级别为2并且使用所有的优化操作。这也是TetGen中默认的选项。选项 ‘-O/1’ 表示只选择边/面转换的操作并且使用默认的优化级别。

**下面的选项是暂时的(将来可能会改变)**

TetGen要优化的当前目标函数是为了减小四面体的最大二面角。默认情况下这个值是165°。我们可以在 ‘-o/’ 之后设置这个值。例如， ‘-o/150’ 表示设置最大二面角为150°。

以后将会支持更多的目标函数。

### 3.2.10 设置容差(-T)

TetGen 中的许多计算要用到容差，用户定义的容差由-T 命令设置。容差的默认值为1e-8。

按照规则，用来确定 PLC 的一个面片的顶点确切来说是共面的。但是在实践中是很难实现的，这是因为计算机使用的浮点数格式的不相容性质。

TetGen 接受的面片中顶点并不是完全共面的而是“近似共面”。只要  $v/l^3$  的比值比给定的容差小那么四个顶点就被认为是共面的，其中  $v$  和  $l$  分别表示四面体的体积和平均边长。

根据输入点集来选择一个合适的容差通常会减少添加点的数量且提高网格质量。

### 3.2.11 格网统计(-V)

对于 TetGen 正完成的工作-V 命令会给出详细的信息。‘V’ 越多细节量就越多。

特别地，‘-V’ 会给出算法进度的信息和更加详细的统计数据包括一个粗糙网格质量报告。为了得到一个已存在网格的统计数据，请在该网格上运行 TetGen，使用 ‘-rNEF’ 命令读取网格并打印统计数据，无需编写任何文件。

‘-VV’ 给出了算法上的更多细节并且执行速度变慢，然而 ‘-VVV’ 仅仅用于调试。

### 3.2.12 其他选项

**-c** -c 命令可保持四面体网格的凸包。默认情况下，TetGen 会删除不在 PLC(区域)内部的所有四面体，该 PLC 可能具有任意形状和拓扑结构，也就是说，可能是非凸的也可能包含洞。如果使用了 ‘-c’ 命令，PLC 外部的四面体不会被删除，网格元素的并集是一个拓扑球。

**-S** -S 命令指定 Steiner 点(不在输入数据中点)的最大数量，Steiner 点是通过网格细化添加的目的在于提高网格质量。默认情况下 Steiner 点数量不受限。

**-z** -z 命令给所有的输出项从零开始编号。这个命令在另一个程序中调用 TetGen 时很有用处。

**-o2** 使用-o2 命令，TetGen 输出的带有二次元素的四面体网格，每个四面体将会有 10 个节点，每个三角面片将会有 6 个节点，每个边将会有 3 个节点。每个元素中这些额外节点的位置在图??中有所显示。

**-C** -C 命令表示指示 TetGen 检查完成网格的连续性。如果它被指定两次，也就是，‘-CC’，TetGen 还会检查该网格的约束 Delaunay(-p 命令)或符合 Delaunay(-q,-a 或 -i)属性。

**-I** -I 命令没有使用迭代数，它抑制输出的.node 文件，所以你的输入文件不会被覆盖。它不能与-r 命令一起使用，因为这将会覆盖你的.ele 输入文件。如果我们正使用.node 文件作为输入则它不应该与-q 命令一起使用，因为不会编写.node 文件，所以也不会有添加 Steiner 点的任何记录。

## 4 文件格式

### 4.1 TetGen 文件格式

表7列出TetGen使用的所有格式。所有文件都是ASCII码形式，可能包含有以#为前缀的注释。点，四面体，面，洞和最小体积约束编号必须是连贯的，开始编号为1或者0。然而，所有的输入文件必须是一致的。当读.node (.poly,.smesh) 时，TetGen自动检测你选择的文件。当其他程序调用TetGen时，如果你想要对象的编码从0开始，那么使用-z命令。

.node	input/output	节点列表文件
.poly	input	PLC.
.smesh	input/output	PLC.
.ele	input/output	四面体文件
.face	input/output	三角形面文件
.edge	input/output	边界
.vol	input	最大体积列表
.var	input	约束列表
.neigh	output	A list of neighbors.

表7 TetGen文件格式一览表

### 4.1.1 .node 文件

第一行: <点个数><维数(3)><属性个数><边界标识(0或者1)>

剩下的行为点列表:

<点编号><x><y><z>[属性][边界标识]

一个.node文件包括三维点列表, 每个点有三维坐标(x,y,z), 或者有1个或者多个属性和边界标识。 .node文件可以作为输入或者输出文件以表示PLC点集, 或者一个格网的点集, 或者嵌入到格网中的附加点集(使用-i参数)。

.node的文件格式示例:

```
# 节点格式, 3 维, 没有属性, 没有边界标识
8 3 0 0
# 节点编号, 节点坐标
1 0.0 0.0 0.0
2 1.0 0.0 0.0
3 1.0 1.0 0.0
4 0.0 1.0 0.0
5 0.0 0.0 1.0
6 1.0 0.0 1.0
7 1.0 1.0 1.0
8 0.0 1.0 1.0
```

属性为与点关联的浮点型物理量值(例如: 质量、导电性等), 并不加修改的复制到格网中。如果-p,-q,-a或者-i是选择的, 每个Steiner点添加到格网中默认属性值为0。

如果指明了'-w'(权重Delaunay四面体剖分)项, 第一个属性为相应点的权重。

如果第一行的第四部分值为1, 文件中最后一列为边界标识。边界标识用于识别边界点(点在PLC面上), 除了TetGen禁用了-B了开关, 否则TetGen生成的.node文件包含边界标识。边界标识在输出文件.node文件是选择如下:

- 如果在输入文件中点指定了非零边界标识, 那么它在输出文件中将有相同的标识。
- 另外, 如果节点在PLC表面上带有一个非零边界标识, 那么这个表面上的节点将指定相同的标识。如果节点在若干的表面上, 节点的边界标识在这些表面标识中任意选择。
- 如果节点在格网的边界上, 那么节点的标识指定为1。
- 其他点分配标识为0

**TetGen可以决定那些点在边界上, 输入边界标识为0(或者没有使用标识)将结果在输出所有点的边界标识为1。**

### 4.1.2 .poly 文件

一个.poly文件表示一个PLC(piecewise linear complex)作为附加信息。尽管这儿没有限制PLC的表面。TetGen需要PLC表示的完整的面边界格网范围。

.poly文件格式包含了四部分, 点列表, 面列表, 洞(体)点集列表和范围属性列表。前三部分是必须的, 第四部分是可选的。

这四部分详细描述如下。在这一章的最后, 给出了单位立方体的示例.poly文件。

**第一部分 节点列表** 所有节点的列表和它与.node具有完全相同的个数。<点个数>可以设置为0, 指明这些点集存放在分离的.node文件中。



第一行: <点个数> <维数 (3)> <属性个数> <边界标识 (0 or 1)>

下面为点的列表:

<点编号> <x> <y> <z> [属性] [边界标识]

...

**第二部分 面列表** 每个面包含一个共面直线图(PSLG, planar straight line graph), 它是一个可以包含线段, 单点和洞的多边形。多边形(polygons)列表包含面。每个多边形有n个角点,  $n \geq 1$ 。它可以退化, 当 $n=1$ 时表示点, 当 $n=2$ 时表示线段。格式如下:

某一行: <多边形个数s> [洞个数] [边界标识]

下面是多边形列表:

<角点个数> <角点 1> <角点 2> ... <角点 #>

...

下面是洞列表

<洞编号> <x> <y> <z>

...

每个多边形指明了角点的个数, 其后这些角点的列表, 它的顺序(逆时针或者顺时针)并不重要。每个多边形目录行不是任意长度的, 因为TetGen读每行数据的最大字符为1024。那么角点列表将打断为几行。

面上的洞(不同与下面体中的洞)用于识别洞内点。洞点集的列表(连续的)位于多边形列表下面。

如果边界标识是1, TetGen将为.face文件中的每个面生成额外的边界标识。

如果边界标识为0, TetGen将自动为.face文件中边界上的面(属于PLC的面)添加标识1。

你可以使用-B参数不允许输出边界标识到.face文件中。

面的边界标识用于识别四面体格网的面属于哪个PLC。因此, 识别哪个面存在四面体格网的边界表面上。常见用途决定不同的边界条件类型应用到格网中。

**第三部分 洞(体)列表** 在体中的洞通过标记一个洞内部点指明。

通常约束Delaunay四面体格网是用于成形的。TetGen通过移除四面体来创建洞。因此可以推出为什么TetGen需要闭合PLC边界。PLC没有闭合的情况下, 全部的四面体格网将被“吃掉”(译者注: 不会生成任何四面体)。如果相邻的两个四面体被移除, 那么它们公用的面也将被删除。

孔洞点放置在一个范围的里面, 根据大致的误差来决定面的哪一边将转化为洞。

某一行: <洞个数>

下面为洞的列表:

<编号> <x> <y> <z>

...

**第四部分 范围属性列表** 这一部分是可选的, 用于分配范围内所有四面体的属性值, 范围约束在最大四面体体积。分别在使用了-A参数或者-a参数TetGen将读取这一段数据。地域属性和体积约束与洞相同的方式传播。

写在x,y,z坐标之后的两个值, 前一个是范围属性(但是仅仅用于-A被选择时), 后一个值表示范围体积约束(仅仅应用于-a被选择时)。如果在坐标之后仅仅就一个数值, 它可以表

示属性和体积约束任意一个，这取决于参数的选择。一个负最大体积约束允许使用-A参数，并且-a参数在外部强加一个空间范围内的体积约束。

下面是一个使用.poly文件，表示单位立方体数据：

```
# 第一部分 - 点列表
# 点的总个数，维数3，没有属性，没有边界标识
8 3 0 0
# 点索引，点坐标
0.0 0.0 0.0
2 1.0 0.0 0.0
3 1.0 1.0 0.0
4 0.0 1.0 0.0
5 0.0 0.0 1.0
6 1.0 0.0 1.0
7 1.0 1.0 1.0
8 0.0 1.0 1.0
# 第二部分 - 面列表
# 面的总数，没有边界标识
6 0
# 面
1# 1 多边形，没有洞，没有边界标识
4 1 2 3 4 # 前
1
4 5 6 7 8 # 后
1
4 1 2 6 5 # 下
1
4 2 3 7 6 # 右
1
4 3 4 8 7 # 上
1
4 4 1 5 8 # 左
# 第三部分 - 洞列表
0# 没有洞
# 第四部分 - 范围属性列表
0# 没有范围属性
```

### 4.1.3 .smesh 文件

.smesh为专门用于表示一个PLC的文件-表面格网文件。.smesh文件格式与.poly文件格式类似，每个小面仅仅为一个共面多边形，内部没有洞，没有线段和点。它比.poly格式更为灵活，当使用其他程序创建.smesh格式的表面格网时，十分简单。

与.poly文件相同，.smesh文件格式包含四部分，分别为点，小面，洞和范围。仅仅第二部分小面的描述与.poly文件不同。它的描述如下：

第二部分 小面列表 每个小面包含一个共面多变形。每个小面的角点列表描述为点索引号的列表。每个小面的边界标识选项房子角点列表的最后。

第一行: <面个数> <边界标识 (0 or 1)>

下面为面的线列表:

```
<角点个数> <角点1> ... <角点 #> [边界标识]
...
```

下面实例为单位立方体小面部分布局：

```
# 第二部分 - 面列表
# 面个数, 没有边界标识
6 0
# facets
4 1 2 3 4 # 前
4 5 6 7 8 # 后
4 1 2 6 5 # 下
4 2 3 7 6 # 右
4 3 4 8 7 # 上
4 4 1 5 8 # 左
```

#### 4.1.4 .ele 文件

.ele文件保存了四面体列表。每个四面体有四个角点(如果使用了-o2命令将有10个角点)。记录了对应与.node文件的顶点索引号。开始的四个节点为角点顶点。如果开启了'-o2'那么余下的六个节点为四面体边界的中间点。图20显示了他们角点和二阶节点的位置。

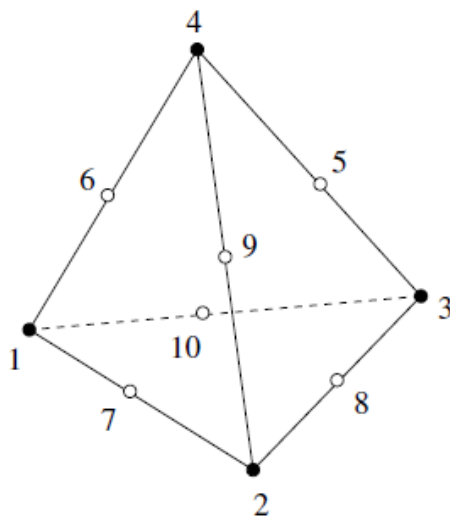


图20 一个四面体的角点和二阶节点的位置编号

```
First line: <# of tetrahedra> <nodes per tet. (4 or 10)>
<region attribute (0 or 1)>
Remaining lines list # of tetrahedra:
<tetrahedron #> <node> <node> ... <node> [attribute]
...
```

如果在第一行的范围属性为'1'，每个四面体的最后一列中额外添加范围属性(一个integer类型)。四面体的范围属性是主要用于标识四面体剖分中四面体关联的那个PLC小面边界范围,在.poly文件和.smesh文件中的第四部分。范围属性不能超出这些小面，在同一范围内的所有的四面体具有相同的范围属性。通常范围属性的使用决定四面体的材质。

TetGen生成了一个格网或者四面体格网，默认输出.ele文件。然而,它可以使用-E禁止.ele

文件的输出。如果使用了-r选项,TetGen读取.ele文件,以重建四面体格网。

下面是.ele文件格式示例:

```
154    4    0
      1    4   107    3    50
      2    4   108    3   107
      3    9    97   95    94
      4    4   107   50    93
      5   56     1   50    47
      6   94    98   97    95
      7   97     9   95    55
      ...
```

#### 4.1.5 .face 文件

.face文件包含三角面片的列表。TetGen默认输出该文件, 那些可能的边界面列表(如果使用了-p或者-r选项)或者外包面列表。如果-f选项被使用,它包含了所有的四面体格网的三角面(包括内部三角面),这个文件可以通过-F选项禁止它的输出。

基本的形式为每个三角面有三个角点,或者包含一个边界标识。记录与.node文件中相关联的编号。

```
First line: <# of faces> <boundary marker (0 or 1)>
Remaining lines list # of faces:
<face #> <node> <node> <node> ... [boundary marker] ...
...
```

如果在首行的边界标识为1,每个三角面最后一列有一个额外的边界标识(整型)。小面的边界标识是定义在.poly或者.smesh文件中。它们主要用于识别四面体格网的三角面与PLC那些小面是相关联的。通常作为确定不同的边界的条件。边界标识选择列可以通过-B命令禁止读取它。

如果-o2是选择的,每个三角面有三个角点和三个位于边界中心的二阶节点。图21显示了三角面的角点和三角面的二阶节点的位置编号。

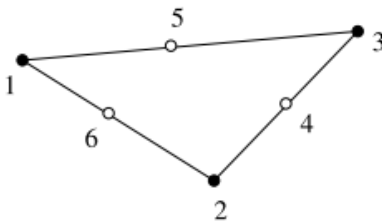


图21 三角面的角点和三角面的二阶节点的位置编号

如果-nn选项是使用的, 每个三角面包含两个与.ele文件相应的附加编号(位于[边界标识]之后)。它们是这个面相邻的四面体。'-1'表示在这个边没有相邻的四面体。它是一个外部面。

TetGen默认仅仅输出边界三角面或者外包三角面到.fcae文件。如果'-f'被使用,TetGen输出所有四面体格网的三角面(包含内部三角面)。在这种情况下,每个内部三角面边界标识通常为0。

如果'-r'选项被使用,TetGen将读取.face文件用于为一个重建的格网指定边界三角面。

#### 4.1.6 .edge 文件

```
First line: <# of edges> <boundary marker (0 or 1)>
Remaining lines list # of edges:
<edge #> <endpoint> <endpoint> ... [boundary marker] ...
...
```

.edge文件包含了四面体格网的边界列表。当'-p'或者'-r'选择被使用时,TetGen默认输出该文件。默认情况下,它仅仅包含一个四面体格网的边界边列表(线段)。如果'-e'选项被使用,它是一个包含四面体格网所有边(包括内部边)。这个文件也可以通过'-F'选项禁止它的输出。

#### 4.1.7 .vol 文件

```
First line: <# of tetrahedra>
Remaining lines list # of maximum volumes:
<tetrahedron #> <maximum volume>
...
```

.vol文件与每个四面体关联一个最大体积值用于格网的细化。'-r'选项被使用时TetGen将读取.vol文件。

与其他文件一样,每个四面体必须被描述,它们编号必须是连贯的。四面体可以不被约束,那么需指定一个负数的最大体积值。

#### 4.1.8 mtr 文件

```
First line: <# of nodes> <size of metric>
Remaining lines list # of point metrics:
<value 1> <value 2> ...
...
```

...

#### 4.1.9 var 文件

```
One line: <# of facet constraints>
Remaining lines list # of facet constraints:
<constraint #> <boundary marker> <maximum area>
...
One line: <# of segment constraints>
Remaining lines list # of segment constraints:
<constraint #> <point1> <point2> <maximum length>
...
```

.var文件允许在线段和面上指定不同的约束条件。与最大体积约束相似,每个面可以有一个最大面积限制。在小面的每个子面的面积不会大于这个限制。每个线段有一个最大长度限制,因此,线段的子线段长度将不会超过限制。

小面约束是设置在与.poly和.smesh文件相关联的指定了整型边界标识的小面上。线段约束是应用在指定了两端点的编号线段上。

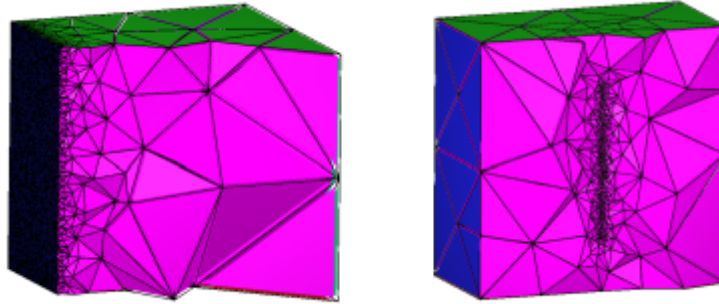


图 22 面约束和线段约束应用实例(.var文件)

图22 展示了应用在一个小面上和一条线段的约束的实例结果。

#### 4.1.10 .neigh 文件

```
First line:  <# of tetrahedra> <# of nei. per tet. (always 4)>
Following lines list # of neighbors:
<tetrahedra #> <neighbor> <neighbor> <neighbor> <neighbor>
...
```

.neigh文件关联每个与.ele文件中相应的四面体的邻接四面体编号。-1表示没有邻接四面体(因为四面体位于格网的边界处)。四面体的第一个邻接四面体i位于四面体的第一个角点的对面,其他邻接四面体以此对应。当-n选项被使用时，TetGen将输出该文件。

#### 4.1.11 .v.cell, .v.face, .v.edge

...

## 4.2 支持的外部文件格式

TetGen完美的支持一些多面体文件格式。表8列出了支持的文件格式。TetGen通过文件后缀识别文件格式。

.off	输入	GeomView多面体文件格式
.ply	输入	多面体文件格式
.stl	输入	立体格式
.mesh	输入/输出	Medit格网文件格式

表8: 支持文件列表

#### 4.2.1 .off 文件

.off是GeomView (<http://www.geomview.org>) 的一种格式。用于Unix/Linux程序3D显示的接口。它通过顶点索引表现共面多边形的连接关系,以此方式描述多面体。多边形可以是凹多边形,但是多边形不能包含孔洞。

.off格式的说明可以在网上找到。下面是.off格式的简单描述:

```
OFF  numVertices  numFaces  numEdges
x  y  z
x  y  z
...  numVertices  like  above
NVertices  v1  v2  v3  ...  vN
MVertices  v1  v2  v3  ...  vM
...  numFaces  like  above
```

注意：顶点编号从0开始(不是1),numEdges始终为0。

#### 4.2.2 .ply 文件

.ply文件格式是对简单对象的描述,为了研究多面体模型的方便提供一种格式。该文件的早期版本用于斯坦福大学和UNC Chapel Hill。

格式的描述是简单的,PLY格式的可以在 <http://astronomy.swin.edu.au/~pbourke/geomformats/ply> 找到。

#### 4.2.3 .stl 文件

.stl或者stereolithography格式是一种广泛应用于制造业的ASCII或者二进制格式。它通过三角形表面列表描述计算机常规实体模型。这是最快速的机械设计模型的标准输入。

.stl 格式的描述可以在网上找到。详细描述参考 <http://www.sdsc.edu/tmf/Stl-specs/stl.html>。下面为简单示例：

```
solid
...
facet normal 0.00 0.00 1.00
  outer loop
    vertex 2.00 2.00 0.00
    vertex -1.00 1.00 0.00
    vertex 0.00 -1.00 0.00
  endloop
endfacet
...
endsolid
```

#### 4.2.4 .mesh 文件

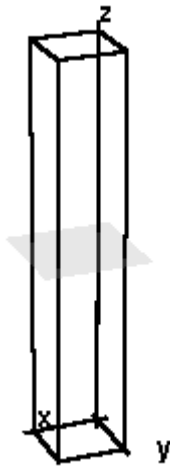
.mesh 为 Medit 格式文件 —3D 格网显示程序交互文件 <http://www.ann.jussieu.fr/~frey/logiciels/medit.html>。该文件格式在Medit文档中有描述。利用INRIA Free 3D Meshes(下载地址<http://www-rocq1.inria.fr/gamma>)可以将3D模型存储在该文件中。

### 4.3 文件格式示例

本章给出两个示例。它们用于学习使用TetGen。使用TetGen的.poly文件格式描述PLC凭借命令行项构造不同的质量的格网。

#### 4.3.1 两个边界标识的 PLC

图显示了矩形条几何体。这个条有8个节点和6个面组成(即所有矩形)。额外,有两个边界标识码(-1和-2),分别标记了左边的面和右边的面。这个简单模型有其自身的物理意义。它可以认为是典型的热传递问题。目的是计算在条上加热端与不加热端之间的热的传递情况。两个边界标识可以描述为两个不同的边界约束,一个是高温面和另一个低温面。下面为输入文件“bar.poly”,来描述矩形条：



```

# Part 1 - the node list.
# A bar with 8 nodes in 3D, no attributes, no boundary marker.
8 3 0 0
# The 4 leftmost nodes:
1 0 0 0
2 2 0 0
3 2 2 0
4 0 2 0
# The 4 rightmost nodes:
5 0 0 12
6 2 0 12
7 2 2 12
8 0 2 12
# Part 2 - the facet list.
# Six facets with boundary markers.
6 1
# The leftmost facet.
1 0 -1 # 1 polygon, no hole, boundary marker (-1)
4 1 2 3 4
# The rightmost facet.
1 0 -2 # 1 polygon, no hole, boundary marker (-2)
4 5 6 7 8
# Other facets.
1
4 1 5 6 2 # bottom side
1
4 2 6 7 3 # back side
1
4 3 7 8 4 # top side
1
4 4 8 5 1 # front side
# Part 3 - the hole list.
# There is no hole in bar.
0
# Part 4 - the region list.
# There is no region defined.

```



命令行选择如下：首先是构建PLC(-p),然后强制质量约束(-q)。这将得到三个文件：bar.1.node, bar.1.ele, bar.1.face。

> tetgen -pq bar

下面是输出文件”bar.1.node”。它包含47个点。TetGen自动将添加满足矢量的额外的点。

```

47    3    0    0
1      0    0    0
2      2    0    0
3      2    2    0
4      0    2    0
5      0    0   12
6      2    0   12
7      2    2   12
8      0    2   12
9      1.0000469999999999    0    0
10     0    0.999668    0
11     0    0.999445000000000003    12
12     1.000594    0    12
...
# Generated by tetgen -pq bar

```

下面是输出文件”bar.1.ele”,包含83个四面体。

```

83    4    0
1 18    33    20    34
2 9      2      3    25
3 17     18     20    34
4 43     32     18    37
5 19     20     30    33
6 14     41     13    42
7 12     26      7     6
8 10     28      1     9
9 28     33     18    34
10 35     41     38    45
11 10      9     25    28
12 3      25     19    30
...
# Generated by tetgen -pq bar

```

下面是输出文件”bar.1.face”包括90个面。面1和面2是属于最左边的面有标识-1;面3和面4有标识2属于最右边的面。其他面有默认标识0。

```

90    1
1 3      4      10     -1
2      10      9      3     -1
3 7      12     11     -2
4 7      11      8     -2
5      18     37     43      0
6      24     46     39      0
7      26      6      7      0
8      35     22     24      0
9      29      7      8      0

```

```

10      39      7      29      0
11      29      11     27      0
12      23      45     38      0
...
# Generated by tetgen -pq bar

```

然而，格网可以用有限元或者有限体方法粗略建立数学模型。使用-q或者-a其中一个选项，或者同时使用俩个选项将得到更为稠密的格网：

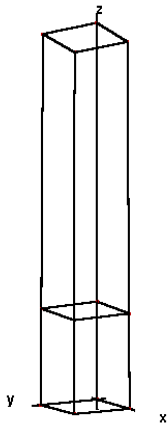
```
> tetgen -pq1.414a0.1 bar
```

TetGen生成一个包含330个点和1092个四面体的格网。添加点是由于使用-q和-a选项得到的。参看一个格网的报告：

```
> tetgen -rNEFV bar.1
```

#### 4.3.2 带有两个范围的 PLC

在这个示例中,我们添加了内部边界面(如图),那么创建俩个范围(在新的添加面上分开)在矩形条上。图显示修改的几何体。这个矩形条包括12个点和7个面(注意:它们其中的一些不在一个多变形或者多个多边形)。另外,它们有两个范围定义,其中属性分别为-10和-20。可以认为两个不同材质的物体进行了连接。保留了前一个示例的边界标识(-1和-2)。下面是输入文件”bar2.poly”,描述修改后的矩形条：



```

# Part 1 - the node list.
# The model has 12 nodes in 3D, no attributes, no boundary marker.
12 3 0 0
# The 4 leftmost nodes:
1 0 0 0
2 2 0 0
3 2 2 0
4 0 2 0
# The 4 rightmost nodes:
5 0 0 12
6 2 0 12
7 2 2 12
8 0 2 12
# The 4 added nodes:
9 0 0 3
10 2 0 3
11 2 2 3
12 0 2 3

```

```

# Part 2 - the facet list.
# Seven facets with boundary markers.
7 1
# The leftmost facet.
1 0 -1 # 1 polygon, no hole, boundary marker (-1)
4 1 2 3 4
# The rightmost facet.
1 0 -2 # 1 polygon, no hole, boundary marker (-2)
4 5 6 7 8
# Each of following facets has two polygons, which are
# one rectangle (6 corners) and one segment.
2
6 1 9 5 6 10 2 # bottom side
2 9 10
2
6 2 10 6 7 11 3 # back side
2 10 11
2
6 3 11 7 8 12 4 # top side
2 11 12
2
6 4 12 8 5 9 1 # front side
2 12 9
# The internal facet separates two regions.
1
4 9 10 11 12
# Part 3 - the hole list.
# There is no hole in bar.
0
# Part 4 - the region list.
# There are two regions (-10 and -20) defined.
2
1 1.0 1.0 1.5 -10 0.1
2 1.0 1.0 5.0 -20 -1

```

输入命令行“tetgen -pqA bar2”生成了文件“bar2.1.ele”。下面列出了文件中8行数据。与“bar.1.ele”不同,每条记录添加了范围属性。

```

431 4 1
1 32 57 50 60 -20
2 51 23 50 49 -20
3 88 138 116 149 -10
4 76 96 95 36 -20
5 29 55 56 52 -20
6 132 138 88 139 -10
7 65 138 132 139 -10
8 16 54 53 15 -20
...
# Generated by tetgen -pqA bar2

```

(借助TetView或者其他工具)可视化显示格网可以看出,属性为-10的范围内比其他范围格网密度大。这是由于定义在文件“bar2.poly”中体积约束(0.1)和使用了’-aA’选项的结果。

## 5 示例

本章借助多种命令行选项介绍了TetGen输入和输出的少量示例。本章的一些输入的解释可以通过TetGen网站获得。

假设已经有编译好的TetGen。编译方法参照章节2.1。

### 5.1 四面体格网生成(-q, -V)

生成一个三维PLC的优良四面体格网,需要下面步骤:

(1) 保存PLC到.poly或者.smesh文件中(test.poly或者test.smesh)。

(2) 使用下面命令运行TetGen:

```
tetgen -pq test.poly
```

图为TetGen输出结果在屏幕上的显示。两个相邻四面体在TetGen通过半径-边比例和最小两面角测定,默认值分别为2和5°。可以指定它们的值在命令选项'-q'之后使用/分开以获得高质量的格网。图17显示三中不同半径-边比率四面体格网。

命令行中添加'-V'选项,TetGen将输出创建四面体格网的报告。如图??所示。

### 5.2 格网重建选项(-a, .var 文件)

TetGen提供了重建四面体格网的多种选项。它允许指明最大四面体体积和最大边界面面积和最大边界线段长度来生成想要的格网尺寸。这些选项可以同时使用。

使用简单的PLC模型来说明格网重建选项的使用方法。PLC模型为一个单位立方体,其内部有一个边界。见图23左。它由俩个输入文件来描述它: cube-e.smesh,其中包含顶点和PLC面,cube-e.edge,其内部包含PLC内部的线。

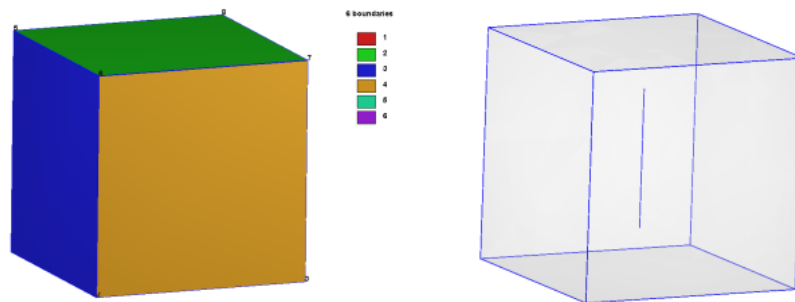


图23:输入PLC(cube-e.smesh,cube-e.edge)。它是一个其内部有线段的单位立方体。

使用强制所有四面体最大体积这种最简单的方式,改善四面体格网为较多小尺寸的格网(意思为有更多的Steiner点)。它可以借助带有希望的体积值的'-a'选择来完成。见图??左示例。

## 参考文献

[1] N. Amenta, S. Choi, and G. Rote. Incremental construction con BRIO.In Proc. 19th ACM Symposium on Computational Geometry, pages 211–219, 2003.

[2] F. Aurenhammer. Voronoi diagrams – a study of fundamental geometric data structure. ACM Comput. Surveys, 23:345–405, 1991.

- [3] M. Bern, P. Chew, D. Eppstein, and J. Ruppert. Dihedral bounds for mesh generation in high dimensions. In Proc. 6th Symp. Discrete Algorithms, ACM and SIAM, pages 186–196, January 1995.
- [4] J.-D. Boissonnat, O. Devillers, and S. Hornus. Incremental construction of the Delaunay triangulation and the Delaunay graph in medium dimension. In Proc. 25th Annual Symposium on Computational Geometry, 2009.
- [5] A. Bowyer. Computing Dirichlet tessellations. *Comp. Journal*, 24(2):162–166, 1987.
- [6] B. Chazelle. Convex partition of polyhedra: a lower bound and worst-case optimal algorithm. *SIAM Journal on Computing*, 13(3):488–507, 1984.
- [7] S.-W. Cheng, T. Dey, H. Edelsbrunner, M. A. Facello, and S.-H. Teng. Sliver exudation. *J. Assoc. Comput. Mach.*, 47:883–904, 2000.
- [8] S.-W. Cheng, T. K. Dey, E. A. Ramos, and T. Ray. Quality meshing for polyhedra with small angles. *International Journal on Computational Geometry and Applications*, 15:421–461, 2005.
- [9] D. Cohen-Steiner, E. C. De Verdière, and M. Yvinec. Conforming Delaunay triangulation in 3D. In Proc. 18th Ann. Symp. on Comput. Geom., 2002.
- [10] J. A. de Loera, J. Rambau, and F. Santos. *Triangulations, Structures for Algorithms and Applications*, volume 25 of *Algorithms and Computation in Mathematics*. Springer Verlag Berlin Heidelberg, 1 edition, 2010.
- [11] B. N. Delaunay. Sur la sphère vide. *Izvestia Akademii Nauk SSSR, Otdelenie Matematicheskikh i Estestvennykh Nauk*, 7:793–800, 1934.
- [12] O. Devillers, S. Pion, and M. Teillaud. Walking in triangulation. *International Journal of Foundations of Computer Science*, 13(2):181–199, 2002. INRIA Tec. Report No. 4120, 2001.
- [13] H. Edelsbrunner. *Algorithms in combinatorial geometry*. Springer-Verlag, Heidelberg, 1987.
- [14] H. Edelsbrunner. *Geometry and topology for mesh generation*. Cambridge University Press, England, 2001.
- [15] H. Edelsbrunner and M.P. Mücke. Simulation of simplicity: A technique to cope with degenerate cases in geometric algorithm. *ACM Transactions on Graphics*, 9(1):66–104, 1990.
- [16] H. Edelsbrunner and N. R. Shah. Incremental topological flipping works for regular triangulations. *Algorithmica*, 15:223–241, 1996.
- [17] P. L. George and P. Frey. *Mesh Generation - Application to Finite Elements*. Hermes Science Europe Ltd, Oxford, UK, 2000. ISBN 1-903398-00-2.
- [18] Damrong Guoy. *Tetrahedral Mesh Improvement, Algorithms and Experiments*. PhD thesis, Computer Science in University of Illinois at Urbana-Champaign, 2001.
- [19] B. Hudson, G. Miller, and T. Phillips. Sparse Voronoi refinement. Technical Report CMU-CS-06-132, School of Computer Science, Carnegie Mellon University, 2006.

- [20] D. T. Lee and A. K. Lin. Generalized Delaunay triangulations for planar graphs. *Discrete and Computational Geometry*, 1:201–217, 1986.
- [21] X.-Y. Li and S.-H. Teng. Generating well-shaped Delaunay meshes in 3D. In *Proc. 12th ann. ACM-SIAM Symp. on Disc. Algo.*, pages 28–37, 2001.
- [22] A. Liu and B. Joe. Relationship between tetrahedron shape measures. *BIT Numerical Mathematics*, 34:268–287, 1994.
- [23] Y. Liu and J. Snoeyink. A comparison of five implementations of 3D Delaunay tessellation. In J. E. Goodman, J. Pach, and E. Welzl, editors, *Combinatorial and Computational Geometry*, volume 52, pages 439–458. MSRI publications, 2005.
- [24] P. McMullen. The maximum number of faces of a convex polytope. *Mathematika*, 17:179–184, 1970.
- [25] G. L. Miller, D. Talmor, S.-H. Teng, N. J. Walkington, and H. Wang. Control volume meshes using sphere packing: Generation, refinement and coarsening. In *Proc. 5th Intl. Meshing Roundtable*, 1996.
- [26] E. P. Mücke, I. Saías, and B. Zhu. Fast randomized point location without preprocessing in two- and three-dimensional Delaunay triangulations. In *Proc. 12th annual Symposium on Computational Geometry*, pages 274–283, 1996.
- [27] M. Murphy, D. M. Mount, and C. W. Gable. A point-placement strategy for conforming Delaunay tetrahedralizations. In *Proc. 11th annual ACM-SIAM Symposium on Discrete Algorithms*, pages 69–93, 2000.
- [28] S. E. Pav and N. J. Walkington. Robust three dimensional Delaunay refinement. In *Proc. 13th International Meshing Roundtable*. Sandia National Laboratories, 2004.
- [29] P. Pebay and P. J. Frey. A priori Delaunay-conformity. In *Proc. 7<sup>th</sup> International Meshing Roundtable*, pages 321–333. Sandia National Laboratories, 1998.
- [30] V. T. Rajan. Optimality of the Delaunay triangulation in  $\mathbb{R}^d$ . *Discrete and Computational Geometry*, 12:189–202, 1994.
- [31] A. Rand and N. Walkington. Collars and intestines: practical conforming Delaunay refinement. In B.W. Clark, editor, *Proc. 18th International Meshing Roundtable*, pages 481–497, 2009.
- [32] J. Ruppert. A Delaunay refinement algorithm for quality 2-dimensional mesh generation. *Journal of Algorithms*, 18(3):548–585, 1995.
- [33] E. Schönhardt. Über die zerlegung von dreieckspolyedern in tetraeder. *Mathematische Annalen*, 98:309–312, 1928.
- [34] J. R. Shewchuk. Adaptive precision floating-point arithmetic and fast robust geometric predicates. *Discrete and Computational Geometry*, 18:305–363, 1997.
- [35] J. R. Shewchuk. A condition guaranteeing the existence of higher-dimensional constrained Delaunay triangulations. In *Proc. 14th Ann. Symp. on Comput. Geom.*, pages 76–85, 1998.

- [36] J. R. Shewchuk. Tetrahedral mesh generation by Delaunay refinement. In Proc. 14th Ann. Symp. on Comput. Geom., pages 86–95, 1998.
- [37] J. R. Shewchuk. Constrained Delaunay tetrahedralizations and provably good boundary recovery. In Proc. 11th International Meshing Roundtable, pages 193–204. Sandia National Laboratories, 2002.
- [38] J. R. Shewchuk. What is a good linear element? interpolation, conditioning, and quality measures. In Proc. 11th International Meshing Roundtable, pages 115–126, Ithaca, New York, September 2002. Sandia National Laboratories.
- [39] J. R. Shewchuk. Updating and constructing constrained Delaunay and constrained regular triangulations by flips. In Proc. 19th Ann. Symp. on Comput. Geom., pages 86–95, 2003.
- [40] J. R. Shewchuk. General-dimensional constrained Delaunay and constrained regular triangulations, i: combinatorial properties. *Discrete and Computational Geometry*, 39:580–637, 2008.
- [41] H. Si. Adaptive tetrahedral mesh generation by constrained delaunay refinement. *International Journal for Numerical Methods in Engineering*, 75(7):856–880, 2008.
- [42] H. Si. Three dimensional boundary conforming Delaunay mesh generation. PhD thesis, Institut für Mathematik, Technische Universität Berlin, Strasse des 17. Juni 136, D-10623, Berlin, Germany, August 2008. Available online: <http://opus.kobv.de/tuberlin/volltexte/2008/1966/>.
- [43] H. Si and K. Gaertner. 3d boundary recovery by constrained Delaunay tetrahedralization. *International Journal for Numerical Methods in Engineering*, 85:1341–1364, 2011.
- [44] H. Si and K. Gaertner. Meshing piecewise linear complexes by constrained Delaunay tetrahedralizations. In Proc. 14th International Meshing Roundtable, pages 147–163, 2005.
- [45] D. Talmor. Well-spaced points for numerical methods. PhD thesis, Department of Computer Science, Carnegie Mellon University, Pittsburgh, Pennsylvania, 1997. Available as Technical Report CMU-CS-97-164.
- [46] G. Voronoi. Nouvelles applications des paramètres continus à la théorie de formes quadratiques. *Reine Angew. Math.*, 133:97–178, 1907.
- [47] D. F. Watson. Computing the n-dimensional Delaunay tessellations with application to Voronoi polytopes. *Comput. Journal*, 24(2):167–172, 1987.
- [48] G. M. Ziegler. *Lectures on Polytopes*, volume 152 of Graduate Texts in Mathematics. Springer-Verlag, New York, second edition edition, 1997.

## 附件 A TetGen 剖分彩图

