# 三维非结构网格的生成及自适应技术

## 概述

非结构网格的基本单元是三角形和四面体。对二维而言，只有三角形能充满任意的二维区域；对三维情形，只有四面体能充满任意的三维区域。非结构网格的生成方法有许多，如Delaunay方法、前沿推进法、叉树法、规则划分法等，其中最常用的是Delaunay方法和前沿推进法。

在非结构网格的生成方法中，前沿推进法提供了一种很好的布点方式，而Delaunay方法则是一种很好的结点连接方法。本文结合两者的优势，提出一种生成三维非结构网格的混合技术——采用前沿推进的方式布点，然后用Delaunay三角化方法对结点进行连接。在生成非结构网格时，首先将边界点用Delaunay方法连接生成原始网格，然后将边界作为最初的前沿，按照背景网格的信息从前沿向内部推进产生新的结点，新的结点再用Delaunay方式引入网格中，再将前沿中原有面用包含它们的新单元的新生成的面代替，继续推进产生新的结点，生成新的网格结构，直到前沿中不再有三角形面存在为止。由于对新生成的结点不是采用传统的前沿推进法的直接连接方式连入网格中，而是用Delaunay方法连入网格结构中，因而在推进时，可以一次推进一层产生一组新的结点。这样不仅能按给定尺度分布生成元素过渡光滑的网格，而且网格的生成过程稳定，生成效率高，生成速度快。为了提高网格质量，本文采用Laplacian光顺迭代技术，对迭代过程中会出现负体积的情况本文采取对局部区域用Delaunay方法进行重连的办法来处理，结果表明这种优化效果较好。

## Delaunay方法和前沿推进法介绍

### Delaunay方法

这种方法要求事先知道区域内点的分布，然后用这些已知的点为顶点，根据区域中任意点到这些顶点的距离，将区域分为与顶点数相同的小区域，每个顶点对应一个区域。其原则（Dirichlet棋盘化原则）是处于一个顶点的区域内的点到该顶点的距离比到其它任意顶点的距离都要小。如果两个顶点的小区域相邻，那么便用一条线段将此二顶点连接起来——对给定的一组点而言，这种连接方式是最优的。这样整个区域便作了三角剖分（二维）或四面体剖分(三维)。这种方法的优点是方便、生成速度快、生成过程稳定、并且网格的质量也比较好。但需要保证物面的完整性，而且网格结点需另外给定。

Delaunay三角化方法是对给定点集的一种最优化的连接方式，其理论依据是Dirichlet棋盘化原则：对于平面上给定的一组点，可以将平面划分为一系列区域，使得每个区域都包含点集中的一个点，且该区域中的任一点与它所包含的这点的距离比与中其它点的距离近。这些区域都是凸多边形的，即所谓Voronoi多边形（图1.1）。容易看出，Voronoi多边形的每一条边都是以它为公共边的两个区域所对应的点的中垂线，这样的两个点称为一个点对，将所有这样的点对相连，则整个平面就被三角化了。这就是Delaunay三角化的基本原理。



图1.1 Voronoi多边形及Delaunay三角化单元

由上面的原理可以推出，用Delaunay三角化法连成的单元有一个重要的性质：每一个三角形单元的外接圆中都不包含其它的点，在三维情形下则是每个四面体单元的外接球中都不包含其它点。这是Delaunay方法的算法基础。关于Delaunay三角化网格生成过程有多种算法，目前大家公认比较好的方法是Bowyer算法。

### 前沿推进法

这种方法是从已知边界的基础上，根据给定的网格尺度分布，向区域内部推进生成网格。二维区域的边界由一组线段构成的环路组成（三维则是由一组三角形面构成的闭合面）。在初始时，所有的边界构成一“前沿”，前沿上的线段（或面）都可用来作为网格新单元的边（或面）。推进过程永远从前沿向区域内部进行。从边界开始，计算新结点的位置，不断的向前沿中加入新的边（或面），删除旧的边（或面）,不断更新的前沿将在区域内部相遇、融合。当前沿中不再有边（或面）存在时，内部区域就被彻底的三角化了。这种方法的优点是不需要进行额外处理以保证物面的完整性，与Delaunay方法相比它可以自己生成结点而不需要另外引入，并且在网格尺度的分布控制上有其方便和优越之处。但是每次推进只生成一个新单元，效率很低，而且每生成一个新单元时，都要对新单元进行可行性测试，计算量非常大，在三维情形就更是如此，必须采取很多数值技巧，采用更合理的数据结构。

## 本文方法

三维非结构网格的生成方法目前最常用的有两种：一是Delaunay三角化方法，它是把空间预先分布的点按Dirichlet棋盘化原则连成四面体。其优点是生成效率高，生成过程稳定，生成的网格质量也比较好。另一种是前沿推进法，它是先生成表面网格，然后以此为初始的前沿向区域内部推进，按背景网格提供的信息自动生成空间点，当前沿内的面为零时，生成也就完成了。与Delaunay方法相比，它可以自己生成结点而不需要引入，而且在网格尺度的分布控制上有其方便和优越之处。但是网格生成过程不稳定，生成效率也很低，尤其是查找要花费大量的时间。

从以上的分析可以看出，前沿推进法提供了一种很好的布点方式，而Delaunay方法则是一种很好的连接方法。基于这一特点，本文在生成非结构网格时，结合两者的优势，采用前沿推进的方式布点，而用Delaunay三角化方法对结点进行连接。在国外有人在二维情形下做过类似的工作。

本文生成非结构网格的具体步骤如下：

（1）将边界上给定的结点用Delaunay方法连接生成原始网格；

（2）对原始网格进行整理，保证边界面的完整性，消除边界外的单元；

（3）整理后的网格，边界面为初始前沿；

（4）从前沿向区域内部推进布点，此时采取层推进方式，一次产生多个结点；

（5）对上述结点进行判断，剔除不合理的点，并将剩下的点引入Delaunay结构中，生成新的网格，前沿中原有的三角形面在推进后被包含它们的新单元的新生成的面代替；

（6）重复（4）、（5）直到前沿中不再包含任何面为止，则网格生成完毕。

下面本文将对以上各点进行详述。

## 三维背景网格

本文对内部结点的生成，采用的是前沿推进法的方式，因而必须有背景网格来提供空间的尺度分布信息。背景网格应覆盖整个计算区域。当生成初网格时，我们可以选取一些有代表性的点，并定好各点上的尺度参数，然后将它们用Delaunay方法连接起来作为背景网格。当生成自适应网格时，则可以用初网格作为背景网格。

当背景网格生成好以后，对域内任一点的控制参数的获得是通过线性插值而得到的，具体的作法是：1)找出包含该点的背景网格单元，其标识是该点与所找到的四面体的四个面所构成的体积值均为正或这四个四面体的体积的绝对值之和不大于该背景网格四面体单元的体积；2）对该点进行线性插值得到所需的尺度参数,插值公式是

 （1.5.1）

其中，为插值点P的尺度参数，、、、为背景网格单元的四个顶点A、B、C、D上的尺度参数，为该背景网格单元体积，、、、为P与该背景网格单元的四个面所构成的四面体体积。

## 由边界点生成原始网格

离散化的三维区域边界应是由一个个的三角形面组成的闭合曲面，边界点包括物面点和远场点。

按照Bowyer算法，在将边界点用Delaunay方法连接起来之前，必须首先给定一个凸外壳包住整个计算区域，并用人工方式将该外壳分为几个四面体，作为最初网格。

对最初的凸外壳，我们选用长方体，并将之分为六个四面体单元。为与正式结点的编号区别，我们对该凸壳的顶点编号采用负数（图1.2）。

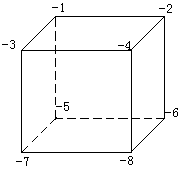
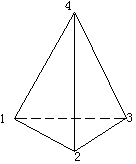
 

图1.2 凸壳结点编号 图1.3 四面体单元的结点局部编号

按照Bowyer算法，每插入一个点都要找出其外接球包含插入点的四面体。因此对各单元的存储通过如下的数据结构来进行：

{外心坐标,外接球半径，组成单元的四个结点，单元的四个相邻单元编号}

对组成每个单元的四个结点，我们按照图1.3的顺序进行局部编号，按此顺序当采用公式。计算四面体体积时，得到的V值是正数，否则得到的就是负体积值。这样，在引入边界点以前，最初的网格单元有六个(相邻单元编号为0表示该方向无单元与其相邻)：

表1 最初网格单元的数据结构

|  |  |  |  |
| --- | --- | --- | --- |
| 单元编号 | 外心及外接球半径 | 组成的点 | 相邻的单元 |
| 1 | （x,y,z）,r | -1,-2,-3,-5 | 3,0,0,0 |
| 2 | （x,y,z）,r | -7,-6,-5,-2 | 0,3,6,0 |
| 3 | （x,y,z）,r | -2,-3,-5,-7 | 0,2,5,1 |
| 4 | （x,y,z）,r | -6,-7,-8,-4 | 0,0,6,0 |
| 5 | （x,y,z）,r | -4,-3,-2,-7 | 3,6,0,0 |
| 6 | （x,y,z）,r | -4,-6,-7,-2 | 2,5,0,4 |

人工给定最初网格后，下面的步骤就是插入边界结点生成原始网格。按照Bowyer算法，将点插入网格中的过程如下：

1. 按随机找一个四面体单元，看它的外接球是否包含待插点，如果不包含，继续搜寻，直到找到一个外接球包含该点的单元为止；
2. 以找到的四面体为基点，搜寻其相邻的单元，判断它们的外接球是否也包含待插点，如果不包含，则停止这个方向的搜索，转向另一个方向；如果包含，则以此四面体为基点开始下一轮的搜寻，直到各方向都搜寻完为止；
3. 搜寻到的这些四面体单元都将被打破，而留下一个由一系列三角形面组成的空腔，即所谓Delaunay空腔；
4. 把Delaunay空腔的每一个面都与待插点连接起来，生成新的四面体单元；
5. 对新单元进行整理，加入到上述数据结构中，则该点插入完毕。

重复以上步骤，将所有边界点都加入网格中后，接下来就要对该网格进行整理。整理的过程包括保证物面的完整性以及消除边界外的四面体。整理的步骤是：

1. 判断每一个边界面的三个点是否包含于网格的某一个单元中，如是则该表面单元存在，否则就不存在；
2. 对不存在的边界单元在其中心处插入一个点加入网格中，同时该单元也进行分裂；
3. 重复上述步骤，直到所有的边界面都存在；
4. 经过上述处理，每个边界面都应为两个单元所包含，很容易区分出这两个单元谁在边界外，谁在区域内，我们将前者用0标记，将后者用1标记。将所有这些单元都标记完毕后，通过这些单元向四周搜寻，凡是与标记为0的单元相邻且不包含边界面的单元，都用0标记，而凡是与标记为1的单元相邻且不包含边界面的单元，都用1标记，直到所有单元标记完毕，这样，我们就将所有单元以边界面为界限，分为了两大类；
5. 所有标记为0的单元都是在区域外的，将它们都从网格中删除，并重新整理网格结构；
6. 将第2步在物面上新插入的点剔除，将包含该点的单元重连，恢复物面。

这样，就得到了由边界点组成的原始网格，且所有单元都在计算区域内部。在下面的过程中，当引入内部结点时，如果某个包含边界面的单元要被删除，则该边界面会自动作为Delaunay 空腔的一个面，参与组成新的四面体单元，因而边界的完整性不会被破坏了。

## 内部结点的生成及三角化

### tuijin.bmp推进生成内部结点

前沿中的每个面都可用于推进生成新结点、新

单元，我们称之为“活跃的面”。

从前沿中选择一个三角形面作为推进的基面，

设为面ABC(如图1.4所示)，（在前沿推进法中，

图1.4

通常要选择最小的面，这样是为了使得前沿提前

相遇的可能性降到最小，以避免生成过多的畸形单元）。设点O是面ABC的外接圆圆心，r是其外接圆半径，是该面的指向前沿所包围的区域内部的单位法向向量，用背景网格插值，可以得到点O上的尺度参数，从O点作有向线段，使其长度h满足：



之所以对h作出这样的限制，是为了避免单元过于畸形。

按照传统的前沿推进法，由面ABC推进生成新结点P的同时，点P将直接与面ABC的三个顶点连接而形成一个新的四面体单元PABC，当然，对于这个新结点和新单元，还要进行可行性判断，如果新单元和前沿相交或者该单元中包含前沿点，又或点P与前沿点距离太近，那么点P就不能参与构成新单元，而应用相应的前沿点来代替它构成新单元。对于三维问题而言，这些工作的计算量非常大。而且，由于前沿包围的空间是一个尚未被离散化的空腔，这也给判断前沿是否相遇，以及判断网格是否生成完毕增加了困难。

本文由于只是采用前沿推进的方式来生成结点，不是采用传统的前沿推进法连接结点的方式来生成新单元，而是随后将新结点用Delaunay方法连入网格中，因此处理方式比之原始的前沿推进法更加灵活。在由前沿向前推进时，不必选取前沿中最小的面作为起始面来推进，而是可以同时从前沿中的每一个面向前推进生成新的结点，一次推进一层，既提高了效率，又避免了对前沿中的面按大小进行排序。当然，为了保证网格的质量，对新生成的结点还要进行一些处理：如图1.4，从面ABC推进生成点P后，通过背景网格插值，可求得该点上的尺度，以P为圆心，0.5为半径作球，查找是否有结点位于该球内，若有，则不要点P，若无，则再对四面体PABC进行判断，查找是否有结点位于其外接球内，如果有，则也不要点P，否则点P将作为新的结点引入到外接球中。如果点P在边界附近，还要判断它是否与某个边界相距过小，如果小于0.5，则P也不能加入网格。由于前面在由边界结点生成初始网格时，采用Delaunay方式进行连接，因而整个计算区域的空间都被四面体充满了，这样，再按照本文前面的数据结构，进行这些查找和判断的工作量并不大。

### 内部结点的三角化及前沿的更替

当从前沿推进一次，生成一批新的结点后，就要将它们用Delaunay方式加入到三角化网格结构中，形成新的结构。内部结点的插入过程与1.5节中边界点插入网格中的过程一样，这里就不再论述了。

当新的结构形成后，前沿也要进行更替了。由于我们在推进时，一次向前推进一层，前沿中的每个活跃的面都参与了本次的推进，因此这些面都应作为不活跃的面从前沿中去掉，这些不活跃的面以后不能再次纳入前沿中。新的前沿将由包含不活跃的面的新单元的新生成的面来组成。当然，如果某个单元不是该次推进后更新三角化结构生成的新单元，但是它有部分面为不活跃的面，那么剩下的面也将作为活跃面加入新的前沿中。但是有几种情况需要注意：1）如图1.5a所示，如果我们确定新前沿时，有三个活跃的面DAB、DBC、DCA属于同一个四面体单元DACB，而这个单元的另一个面不是不活跃的面，那么我们就规定这三个面将作为不活跃面（虽然它们并未参与上次的推进），而该四面体的另一个面ABC将作为活跃面加入新的前沿；2）如图1.5b，对于新结构中的两个单元ABCP与BDCP，若只有ABC与BDC是不活跃的面，那么我们在确定新的前沿时，BCP将被规定为不活跃的面而不加入前沿；3）按照我们前面的原则确定新前沿时，如果某个单元的四个面都作为了活跃的面加入前沿，这时就是所谓前沿相遇，这四个面也将被规定为不活跃面，从前沿中去掉。前沿不断更替，不断的生成新的结点插入网格中，直到前沿中不再有活跃的面为止，则推进完毕。

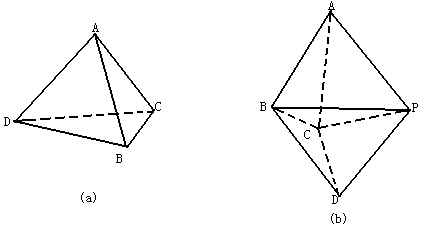


图1.5

### 网格的光滑处理

按照上面这种层推进，对于网格尺度变化剧烈的区域，难免会有一些因尺度过渡不够光滑而导致的畸形单元，对于这些地方还要做一些特殊处理。本文对于这些地方的畸形单元，采取在其外心处加入新的结点插入网格中的方法来使得网格的尺度过渡光滑化。

## Delaunay三角化时由于误差造成网格生成失败的处理

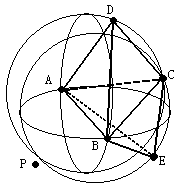
在实际的计算中，由于计算误差的存在，因而有可能造成搜寻其外接球包含待插点的四面体时发生判断错误，把本来其外接球不包含待插点的四面体搜寻了进来，或把本来其外接球包含待插点的四面体判断为外接球不包含待插点，这时一般是待插点刚好位于球面附近，因而发生了判断错误，这种情形在二维情况下很少见，但是在三维中却是经常出现的。这样就会导致出现下面这种情况：两个相邻的单元ABCD和ACBE，待插点P在它们两者的外接球附近，由于误差，判断时将ABCD搜寻了进来，而ACBE则判断为外接球不包含待插点，这样将形成一个新单元ABCP与ACBE相邻，而事实上点P和E在面ABC的同侧（图1.6），这就导致网格生成失败。

图1.6

对这类情况，本文的处理方式是：1）在判断一个单元的外接球是否包含待插点时，加一个正小量，将所有外接球包含待插点或待插点在其外接球球面附近的单元都搜寻进来；2）以待插点为基点，将搜寻到的四面体都打破，形成Delaunay空腔；3）对Delaunay空腔中的每个面，计算它们与待插点所构成的四面体的体积（按照2.3节的公式），如果体积为负值，就表明出现了上面所说的那种情况，4）找出出现这种情况的面是由哪个四面体被打破而行成的，并将该四面体从搜寻到的单元序列中释放出来，重新加入网格中，然后重新对序列中剩下的单元进行打破，得到新的Delaunay空腔。5）重复步骤3、4，直到Delaunay空腔中的每个面与待插点构成的四面体体积都为正值。

## 三维非结构网格的自适应

本文采用了重新生成网格的方法来实现自适应加密。生成自适应网格的过程与生成初网格的过程是完全一样的，生成自适应网格所用的背景网格是初网格。

现在问题的关键是要在背景网格即初网格的结点上给出尺度参数。如何根据初网格上的计算结果确定其结点上的尺度呢？直观的说，我们希望在解变化剧烈的地方，网格取得密些，而在变化平缓的地方，则取得疏些。为此，本文选取了密度的梯度的大小来作为标准，对于背景网格上的一个结点，其上的尺度h应满足

 (1.8.1)

显然，在均匀流动区，由于密度梯度小，计算的尺度h将很大，而在激波附近，由于密度的梯度大，计算的尺度h将很小。在实际应用中，对于尺度值给定了上界及下界，以避免生成过分畸形的单元。

## 网格的优化

对于已经生成好的四面体网格，如果对其中的部分单元形状或者结点分布不满意，可以对网格的结点坐标略作调整，这就是网格的优化技术。对于尺度变化剧烈的网格（例如自适应网格）而言，在网格生成过程中，会不可避免的形成一些畸形的单元，因而，对网格进行优化是必不可少的。

本文在对结点进行调整时，采用的是Laplacian光顺法：对于一个内部结点设与之相连的结点总数为N,则光顺迭代技术如下：

 （1.9.1）

其中，上标表示第次迭代结果，表示与相连的第个结点坐标。

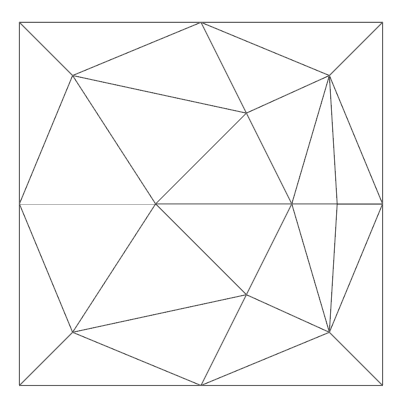
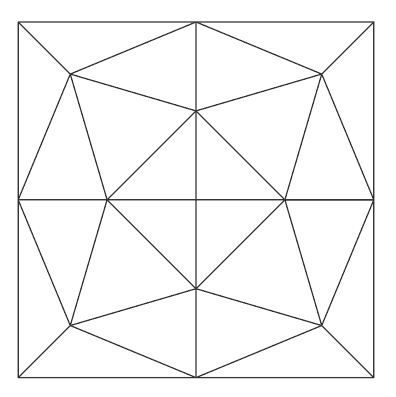
经过上面这种迭代，网格总体的质量会得到很大的改善。但是在三维情况下，每迭代一次，往往都会导致一些单元出现负体积。为解决这个问题，本文在每迭代一次后，都对出现负体积单元的区域的结点用Delaunay方法进行局部重连，以保证网格的正确性。对于优化前后网格质量的评估，本文引进了一个评估标准：四面体的每个面对应一个高h，该面的大小用它的外接圆半径r来衡量，定义参数hr＝h/r，对每个四面体而言，它有四个这样的参数，我们取其中最小的一个来作为评估该单元的标准，记为hrmin。这个值越大，就说明该单元越接近正四面体。表1.2是Hermes外形的简化航天飞机的自适应网格在光顺前后的网格单元质量比较。从这个比较可以看出，光顺过后结点的局部分布情况更为合理，网格的质量也有了明显的改善。

表1.2 Hermes外形自适应网格在光顺前后的网格单元质量比较

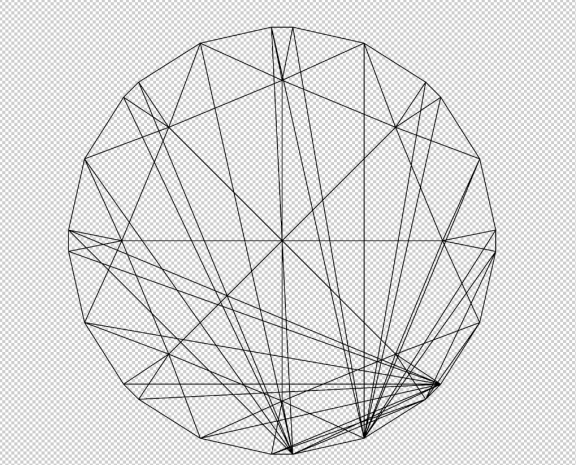
|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| *hrmin* | 1.2以上 | 1.0-1.2 | 0.8-1.0 | 0.65-0.8 | 0.5-0.65 | 0.5以下 |
| 光顺前 | 4.7% | 22.6% | 34.9% | 21.3% | 10.4% | 6.1% |
| 光顺后 | 9.6% | 42.5% | 35.4% | 9.6% | 2.0% | 0.9% |

**二维非结构网格的生成**

1. **背景网格**：提供空间的尺度分布信息，完全覆盖整个计算区域。当生成背景网格时，我们可以人工方式选取一些有代表性的点，并自定义规定好各点上的尺度参数或步长（保证最终的网格该密的地方密一点，该稀疏的地方稀一点），然后将它们用Delaunay方法连接起来生成背景网格。



图一 图二



图三

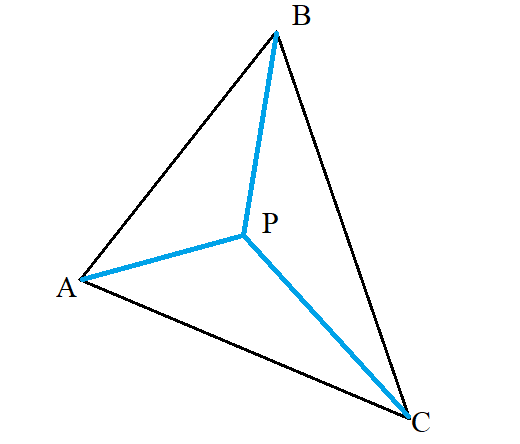
图一为jSR直径由600变为450，图二为RyR位置由 (0,0) 变为 (150,0)。（谢老师之前用的生成完背景网格后的第一步结果即离散化后的边界网格作为第一步结果（图三），这里直接用的delauary三角化后的背景网格）bggridt.dat 是自定义选取的坐标，bgstep.dat是尺度参数，或者步长。通过Delaunay方法进行三角化，生成两个文件：bgnod.dat，存储每个三角形单元由哪几个点组成；bgnoe.dat，存储每个三角形单元相邻的三个单元是哪几个。

按照Delaunay方法，在所有点用Delaunay方法连接起来之前，必须首先给定一个最初的外壳包住整个计算区域，我们选用长方形，并用人工方式先将该外壳对角连起来分为两个三角形，对外壳的顶点编号采用-1，-2，-3，-4编号。每插入一个点都要找出其外接圆包含插入点的三角形单元，将搜寻到的这些三角形单元打破，留下一个由三角形边组成的Delaunay空腔，将插入点与空腔每一个点连接起来形成新的单元。

重复以上步骤，将所有点都加入网格，并将外壳的四个负值的点以及该点连的边消除，生成的就是背景网格。

2. **边界点离散化**：选取边界上一点的坐标，例如半径为225的圆，选取(225,0)为第一个点，找出包含该点所在的背景网格单元，对该点进行线性插值得到所需的尺度参数，插值公式是：





小圆的边界点都是出流边界点，而大圆的边界点有壁面边界点和入流边界点。在对大圆进行边界点离散化时，将这个大圆根据入流点手工分为几段弧，每段弧上的属性都是一致的。人为定义第一个点作为起始点，一般按照模拟的场在左侧方向，所以大圆按照逆时针方向生成下一个点，小圆则按照顺时针方向生成下一个点，可以先小圆后大圆，也可以先大圆后小圆。

一条弧上的两个端点都是离散后会生成的点，当根据一个点的步长生成了下一个新点，而新点在一条弧终点附近时，就要对这个新点做判断：

（1）如果这个新点超出这条弧的区域，延伸到下一条弧时，就要舍弃这个新点，而且如果终点步长的1.5倍大于这段弧剩下的长度，直接用弧的终点作为这条弧上生成的最后一个点；

（2）该点还在原来的弧上，但是该点离弧终点(下一个弧起点)太近，取平均位置作为新点插入；

（3）要新加的点既没有超出该弧的区域，又没有离得终点太近，说明可以直接插入这个点；

（4）如果新加的点的步长的1.5倍还小于上个点的步长，那么不加这个新点，将这个新点的位置往回缩0.5倍的新点的步长，然后再做上述第（1）、（2）、（3）步的判断。

这里会用到的数学公式有：



其中是圆心角度数，是半径，是圆心角弧长。圆心在的圆的极坐标和直角坐标的转换：



每一个圆生成两个文件boungridt.dat 和bounpoch.dat，分别记录边界点的坐标和边界点的属性。

对于小圆圆心位置改变，可以仍然按照原来的方法依次生成下一个点，只是在每次在新生成一个点时，坐标分别加上与圆心位置的偏移量。

3. **插入边界点生成原始网格**：离散化的二维区域边界应是一组线段组成的闭合环。将两个圆的离散点的坐标和属性分别合并为sidegridt.dat和sidenpoch.dat，生成sideface.dat，用来记录边界上每条边两端的顶点序号和这条边上的属性。 载入背景网格和离散化的边界，和第1步生成背景网格相似，用外壳包住所有的边界点，用Delaunay方法依次连接起来生成原始网格。在所有边界点加入网格，要对网格进行整理，保证边界的完整性以及消除边界外的凸壳。

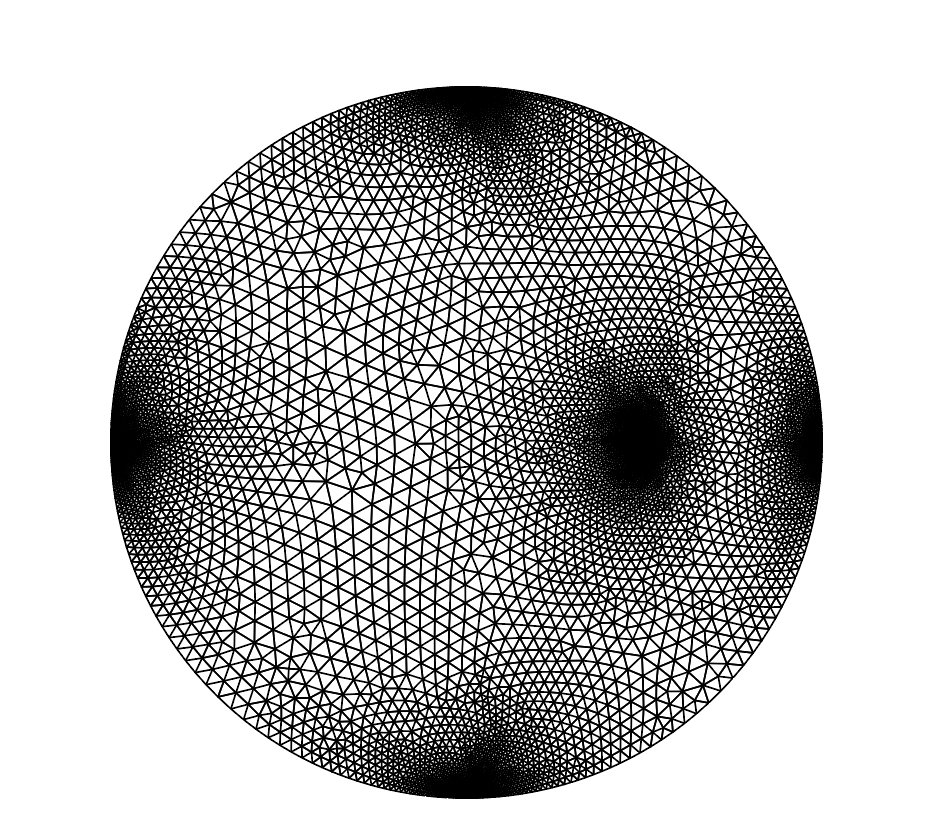
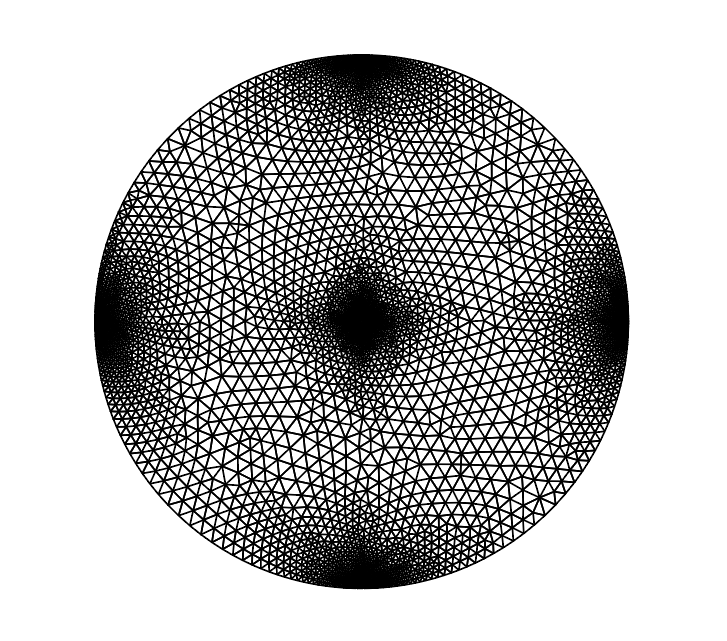
4. **内部结点的生成和三角化**：对于内部结点的生成，采用的是前沿推进法，使用背景网格来提供空间的尺度分布信息。选择边界上每条边的中点，根据该点在背景网格中的位置得出步长，沿该边法向量的方向向内延申距离生成新结点，一次从前沿推进生成多个内部结点，对每个点判断是否合适（如果与某个边界相距过小就要舍弃等等）。每从前沿推进一次，生成一批新的结点后，就要将他们依次用Delaunay方式加入到三角化网格结构中，形成新的结构。

5.**网格的优化：**生成完网格后，最后平滑网格，因为最后可能还存在部分的网格有点畸形，用包含这个点的多边形的坐标平均来代替这个点，即质心的位置。

最后生成的主要的文件有gridt.dat，nod.dat，noe.dat，npoch.dat，分别记录整个网格所有点的坐标，每个三角形单元的三个顶点的标号和每个三角形相邻的三个单元，以及每个点的属性。

最终生成的图像需要用到的文件为gridt.dat,nod.dat,以下为两种不同情况的图像，即一大圆半径由600变为450；二大圆圆心由（0，0）变为（150，0）

永肾：



尹帅：