算法思想

在截面环的求交算法选择上，首先想到通过若干组平行线和模型相交，以获取截面环。但这样的算法不仅效率较低，而且在许多部位上可能会出现漏点的情况，最终导致截面环成型困难。

而相对的，用平面对模型截交，不仅在效率和准确度上更高，在数据的存储上可以顺序存储从而避免了后面排序成环的过程，降低复杂度。

算法的思路是通过点法式（法向量和平面上一点）确定平面，并将其与模型求交，由于平面和三角面片相交，有且仅有两条边与截面相交，通过其中一条可以找到另一条，另一条再去另一个平面找下一条，通过循环便可形成截面环。考虑到截面和物体相交可能不止一个环，例如对竖放的葫芦做垂直截面，会出现上下两个环。则重复这一过程形成多个环。所以多个环有间隔点。

求一层截面环算法如下：（IntersectPlane）

函数输入：截面上的一点pt

截面对应的法向量pnorm

1. 设置一个和半边数量相等的flag[]，用于循环遍历时候标记使用，遍历所有的边，先将flag[]设置为false，有交点则为true。
2. 循环遍历所有边，依次取出边的两个点的坐标值，通过点法式计算两点和截面的位置关系【1】，如果[i]号边两个点分别在截面两侧或者有一个在面上，则该边一定穿过截面，所以标记flag[i] = true，并计算其中一点到pt的距离。选取所有距离中最短的距离为mind，记录其所在边的编号为ne。
3. a. 如果上述过程（2）没有发生，则直接表示没有相交，跳出循环（suc == false）。

b. 否则如果发生，则flag中记录了所有有交点的边的序号。ne边是离平面最短距离的边，标记ne为开始边

starte，计算并保存交点。 当全部轮廓交点找到时，suc = true。计算出的数据存储在coord中。

1. 从ne所在面nf出发，取nf的一条非ne且有交点的边ne1，计算交点并记录，再获得ne1的反向对边ne2,将ne2赋值给ne，所有记录交点操作后都将flag = false。重复上述过程，直到最后的ne边等于最初的starte边，所有的边均完成遍历。对应代码do…while…。[4]
2. 等循环结束后，再遍历所有边找还没有遍历过的边flag = true的边，如果有证明物体和截面有不止一个闭合环。先插入间隔点，再以该边为起点进行第二次循环遍历，重复过程3、4，如此以至所有flag都为flase，即所有有交点的边都记录过交点坐标。

相关知识点

1. 点到直线的距离。[1]

<https://blog.csdn.net/qq_23869697/article/details/82688277>

补充：平面方程是：

当平面是三个基本面（x-y、x-z、y-z）时，p（x1,y1,z1）到平面距离就是：

这里面的法向必须是单位向量。

1. 通过相似三角形计算ne与截面的交点，x坐标的计算如下列公式，y，z原理也相同。(v1,v2是ne的两个端点，d1是v1和平面的距离，同理d2)。

P.x =

1. OpenMesh的迭代器，命名规律：

（1）如果只是迭代整个mesh的一个元素，则只带一个名字，如：

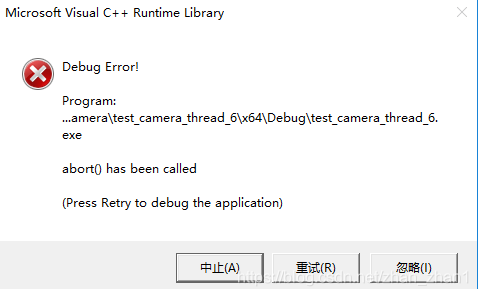
for (MyMesh::VertexIter v\_it = mesh.vertices\_begin(); v\_it != mesh.vertices\_end(); ++v\_it)

迭代整个mesh的所有顶点。

（2）如果是迭代某个元素的相邻元素，则[名字名字迭代器]，如：

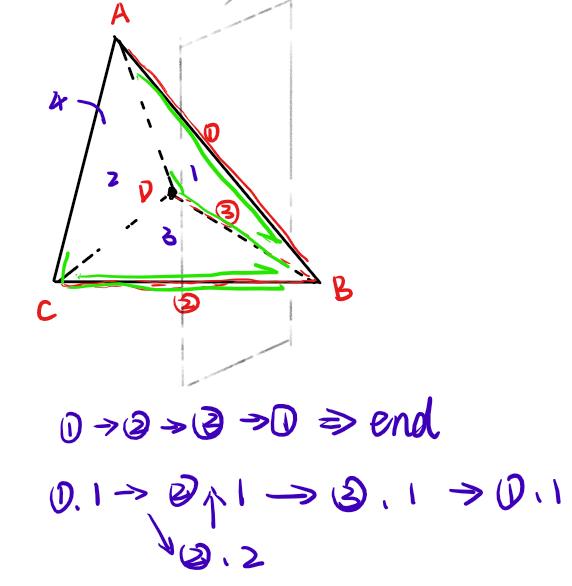
for (MyMesh::VertexFaceIter vf\_it = mesh.vf\_iter(\*v\_it); vf\_it.is\_valid(); ++vf\_it)//这个顶点所带有的面迭代器

for (MyMesh::FaceHalfedgeIter fh\_it = mesh.fh\_begin(\*vf\_it); fh\_it != mesh.fh\_end(\*vf\_it); ++fh\_it)//面的半边迭代器，一个面有3个半边



1. 基于openmesh半边结构的实现：

上述的算法描述中第4步要考虑半边结构的特点，openmesh采用半边结构绘制网格。如右图所示，在模型中边①属于面ABD和面ABC，但是在半边结构中边①实际上是有两条不同的半边1.1和1.2组成的，边1.1属于面ABD，边1.2属于面ABC。



循环开始，在边ne找到交点并记录，通过face\_handle循环找到ne1，并找到交点。这时通过找ne1找对边ne2，因为在图形上ne1、ne2是同一条边，所以只需要记录一次交点。但是找到ne2相当于找到了另一个有交点的面，所以把ne2赋值给ne，判断如果ne不是起始边starte，则继续face\_handle循环。



算法待改进：

1. 多个环可以是独立的连个环，也可以是内外嵌套的环，处理方法不一样。
2. 因为有的模型数据量很大，所以容易发生数组越界。可以全部改用vector，并且时刻注意清除。参考test\_slice.cpp，这个cpp应该去掉大多数多余的参数，比现在的切片要简洁一些，但是因为时间关系没有整合在一起。