第五次作业报告

PB21010362 汪兆辰

2023 年 4 月 15 日

1 算法介绍

对于开放的三维网格，可以找到其边界点并通过固定边界点将网格展开到二维区域上. 在这里我们将边界点 $u_{n+1}, \ldots, u_N$ 均匀固定到单位元上，从而知道了边界点坐标. 对于网格的内部点，利用方程解出：

$$u_i = \sum_{j=1}^{N} \lambda_{i,j} u_j, \quad i = 1, \ldots, n. \tag{1}$$

在这里使用均匀权重，即方程中令 $\lambda_{i,j} = \frac{1}{d_i}$，其中 d_i 为内部点 u_i 的邻接点数量. 由于方程右侧的 u 同时包含了已知量与未知量，将方程移项成如下形式:

$$u_i - \sum_{j=1}^{n} \lambda_{i,j} u_j = \sum_{j=n+1}^{N} \lambda_{i,j} u_j, \quad i = 1, \ldots, n. \tag{2}$$

这样就可以将方程组化为矩阵方程的形式:

$$Px = y \tag{3}$$

其中 P 为系数矩阵，x 为内部点构成的列向量，y 为边界点经方程右侧运算得到的列向量.

2 算法实现

为了得到给定三维网格的各点坐标，点之间的连接关系与边界点等信息，在 main.m 中调用 readObj 与 findBoundary 两个函数，并将所得的结果全部作为成员存储在结构体 obj 中.obj 的结构如图一，其中 vertices 存储了所有顶点的三维坐标；faces 用序号记录了每个三角形面的顶点，也即网格中各点的链接关系；boundary 存储了所有边界点的序号，虽然不懂它生成的原理，但实验表明 boundary 是保序的；inner 存储了所有内部点的序号；adj_mat 是在均匀权重的过程中计算出的邻接矩阵，将 faces 中的邻接关系转换为了矩阵的表示.

在实际计算的过程中，将方程 (2) 的 λ 全部扩大为 1，方便后续的计算与赋值. 考虑到 faces 中的三角形顶点未必都是按照顺时针或逆时针排列的，即如果按照相同的方向遍历每个三角形，可能存在

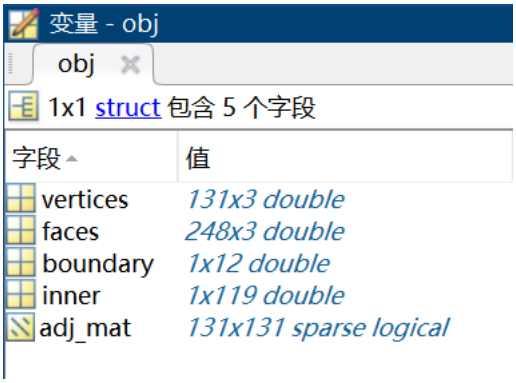


图 1: obj 的结构

某条边在相同方向上被重复遍历，而相反方向上未被遍历，这样将导致邻接矩阵生成错误. 直观来讲，如图二，若在 faces 中对两个三角形的记录分别是 [A B C],[B C D], 而在向邻接矩阵中赋值时，会将 (A,B),(B,C),(C,A),(B,C),(C,D),(D,B) 赋为 1，这就使得 (B,C) 被赋了两次，而 (C,B) 没有被赋值.

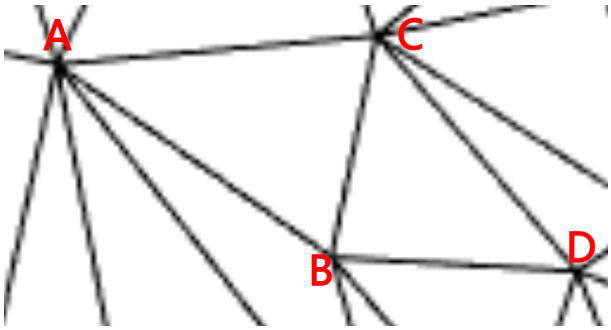
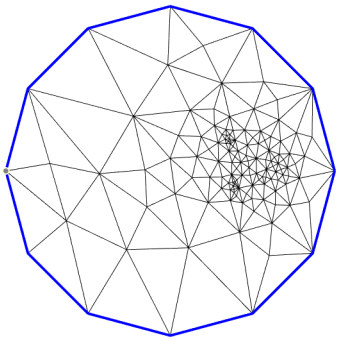


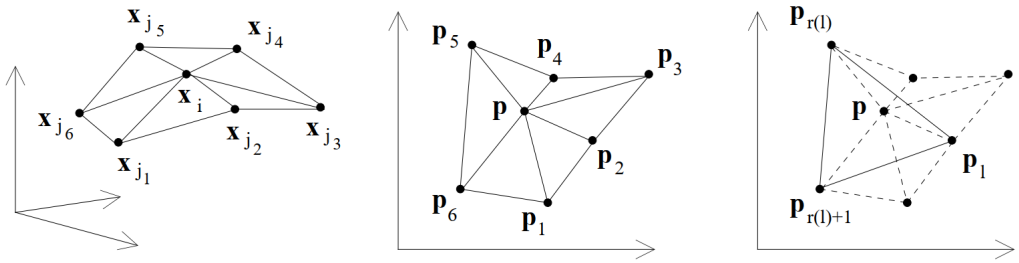
图 2: 示例图

为此，考虑分别由顺时针和逆时针得到两个矩阵 A_1, A_2，此时可以得到邻接矩阵 $A_0 = A_1|A_2$，将每行求和构成对角阵再减去邻接矩阵，就得到了需要的系数矩阵 A. 由此可以建立方程 (2) 并求解，得到结果如下：



3 Bonus:Floater 保型参数化

采用 Floater 保型参数化的方程构建与求解部分与上述完全相同，需要改变的是系数矩阵的赋值. 算法的大概思路是对每个内部点取其周围的邻接点，与其构成一个子图. 为了保型，也即保边长与角的比例关系，需要子图中各点与中心点的长度不变，各夹角之间的比例不变.



为了得到子图中每点的权重，考虑使用重心坐标. 对于每一个顶点 p_k，有且仅有一个顶点 $p_{r(k)}$，使得中心点 p 在由 $p_k, p_{r(k)}, p_{r(k)+1}$ 构成的三角形内部，从而此时中心点对三个顶点的重心坐标都非负. 对每个顶点都作上述操作，得到所有的均值坐标相加，再除 d_i 进行归一化，就得到了每个点的权重. 而对每个内部点都得到这样一个子图进行操作，就得到了要求的系数矩阵 A

在算法实现的过程中，需要将子图中的邻接点保序排列，实现的代码如下：

```
sorted_adj_ver=zeros(1,di);
sorted_adj_ver(1)=adj_ver(1);
for j=2:di
    adj_verj=find(obj.adj_mat(sorted_adj_ver(j-1),:));
    joint_ele=intersect(adj_ver,adj_verj);
    left_ele=setdiff(joint_ele,sorted_adj_ver);
    if ~isempty(left_ele), sorted_adj_ver(j)=left_ele(1); end
end
```

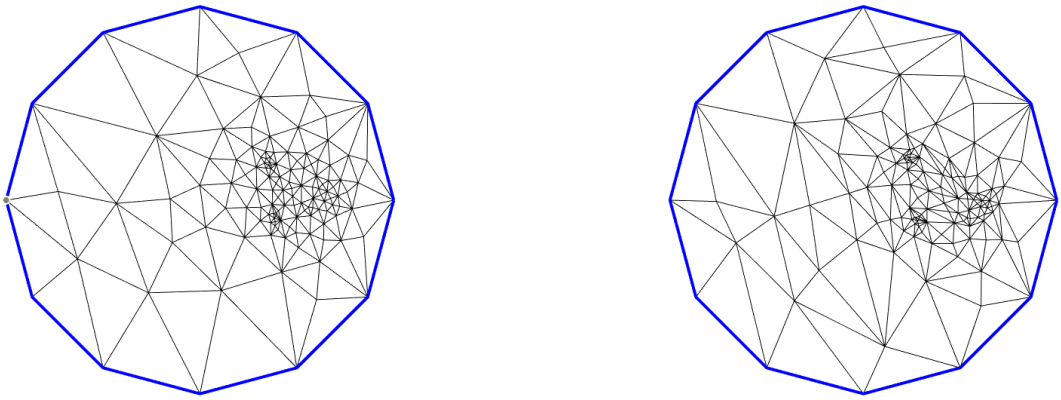
在这里 adj_ver 是按照序号的大小排序的邻接点，将其首位元素 p_{i_1} 作为排序后的首位元素，则第二位元素即在 p_i 与 p_{i_1} 的邻接元素中选取一个即可，重复操作，注意不能将已排过序的顶点再次放入排序后点列中，从而就得到了排序后的点列.

为了方便后续的操作，将子图放在复平面中，其中中心点位于原点处，只需要存储每个点的模长与辐角就可以确定每个点的位置，同时由辐角还可以方便的得出 r(k)，从而确定需要计算的均值坐标. 而均值坐标的计算即解方程：

$$\begin{pmatrix} 1 & 1 & 1 \\ x_1 & x_2 & x_3 \\ y_1 & y_2 & y_3 \end{pmatrix} \begin{pmatrix} \alpha \\ \beta \\ \gamma \end{pmatrix} = \begin{pmatrix} 1 \\ x_0 \\ y_0 \end{pmatrix} \tag{4}$$

在这里有 $x_0 = y_0 = 0$, 而三个顶点的坐标由模长与辐角容易得出.

在得到系数矩阵 A 之后求解过程与均匀权重相同，结果如下图 (b)：



(a) uniformly　　(b) Floater

参考文献

[1] Floater. Parametrization and smooth approximation of surface triangulations. CAGD1997