# 空间分析算法

# 实习报告

**班级： 地理信息科学18-3**

**姓名： 江盟**

**学号： 201801020812**

**提交日期： 2021.1.7**

|  |  |  |  |
| --- | --- | --- | --- |
| 实习时间： | 17周一周 | 实习地点： | J14-404等 |
| 实习评语：  指导教师签名：评阅时间： | | | |

# 实验任务和要求：

运用熟悉的编程语言（C#、C++、Python、Java、Matlab等）进行如下算法的设计与开发，要求程序运行稳定、运算结果可靠，并以图形化界面实现结果输出（实验报告中出现的内容这里不作要求，请删除）。

1. 空间实体量测算法设计

（1）请实现两个空间实体间的距离计算：点点距离、点线距离、点面距离、线线距离、线面距离、面面距离；

（2）请实现球面上两点间的距离计算；

（3）请实现面状实体的周长、面积、中心、重心、质心的计算，以及紧凑指数等形态参数的计算；

（4）请实现DEM表面的表面积以及指定高度的体积计算；

（5）请实现现状地物的曲率和弯曲度计算；

1. 线状数据的压缩算法

请运用道格拉斯普克算法进行线目标要素的数据压缩算法设计，以图形化方法实现压缩前、后的线状要素。

1. 不规则三角网（TIN）生成算法

基于离散点数据，运用Delaunay三角网生成准则，使用三角网生长法或逐点插入算法进行 TIN的生成。

1. 累计表面生成及应用算法设计

基于一幅50\*50的网格数据（每个格点值代表通过该网格的花费），随机设定一个网格作为起点，计算该起点到达其它网格的最小花费，进而生成累计表面矩阵。

1. 矢量数据叠加分析算法

（1）点到曲线最短距离计算；

（2）点在多边形内外的判定计算；（实验已做）

（3）曲线与曲线的求交运算；

（4）曲线与面的求交运算；

（5）面面的求交运算。

1. 栅格数据叠加分析算法

（1）进行栅格数据的局部变换、邻域变换、分带变换、全局变换等计算；

（2）基于地图代数，进行两幅栅格数据间的叠加分析计算。

1. 矢量数据的缓冲区分析计算

（1）进行单点、多点的缓冲区生成；（实验已做）

（2）分别基于角平分线法和凸角圆弧法进行线状要素的缓冲区生成；（实验已做）

（3）进行面状要素的缓冲区生成。

1. 基于DEM的地形特征提取算法

以某区域的DEM数据为基础，进行多种地形特征提取的算法设计，主要包括：地形因子计算（坡度、坡向、表面积、体积、坡度坡向变率、曲率、粗糙度、凸凹系数）、地形特征点的提取（山顶点、凹陷点、脊点、谷点、马鞍点、平地点）以及流长、汇水量、流域盆地等的计算网络分析中的最小代价生成树算法。

9. 网络分析中的最小代价生成树算法

基于一幅加权无向网络图，运用破回路算法，进行最小代价生成树的生成，并进行图形表达。

# 开发环境及系统运行环境

## 开发环境

Python 3.6.6

## 运行环境

运行平台：Visual Studio Code



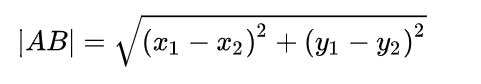
# 算法原理及实现步骤

## 空间实体量测算法设计

**3.1.1 多边形周长计算**

3.1.1算法原理

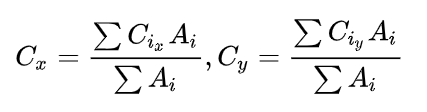
距离：根据坐标俩点A（x1，y1），B(x2,y2)进行，利用距离公式进行计算，公式为



面积：实现计算凸多边形面积，应该按顺序读入每一个点的坐标。然后它划分成若干个相邻的三角形，再分别计算每一个三角形的面积。最后把所有三角形面积的总和，累加起来就是所求的答案。三角形面积：使用海伦公式： 

中心：将多边形所有顶点x，y相加，分别取平均数，得到即为中心点坐标。

重心： A为每个小三角形的面积



**3.1.2实现步骤**

（1）获取鼠标点击画板的坐标点，存入列表中

（2）依次遍历点， 根据公式计算多边形的面积，边长，中心点，重心点等

（3）利用画图函数将结果显示在画板中，可视化表示。

## 不规则三角网（TIN）生成算法

**3.2.1算法原理**

先在点集重任取一点，找到与其相聚最短的点连接成为三角网的一条边，然后按照Delaunay的判别法（最大最小角，空外接圆等）找出包含此边的Delaunay三角形的另一个端点，依次处理所有生成的边，直至所有的边找不到形成合理Delaunay三角形的端点

**3.2.2实现步骤**

（1）在所有数据中取任意一点（一般从几何中心附近开始），查找距离此点最近的点，相连后作为初始基线

（2）在初始基线右边应用 Delaunay法则搜寻第三点3，形成第一个Delaunay角形

（3）并以此三角形的两条新边作为新的初始基线

（4）重复步骤（2）和（3）直至所有数据点处理完毕

## 矢量数据的缓冲区分析

**3.3.1算法原理**

缓冲区分析：

点的缓冲区构建：点的缓冲区构建原理比较简单，以点为圆心，以缓冲距离d为半径，作圆即可。

线的缓冲区构建：缓冲区在一般直线段位置处的构建很容易理解，就是以原始线段为基准，根据缓冲距离d，生成一条与原始线段长度相等、平行、相距d的缓冲线段。而在线段端点或者凹凸点处（也就是拐点处）的构建原理，一般情况下是不清晰的。重说明在线段端点或者凹凸点处（也就是拐点处）的构建原理：

在线段端点处，以端点为圆心，以缓冲距离d为半径，作半圆，与原始线段的缓冲线段相交，生成端点处的缓冲区在线段的拐点处，对于内拐点，也就是夹

角小于180度的方向，此处作角平分线，将原始线段的缓冲线段进行延伸与角平分线相交，此时就构建了内拐点的缓冲区。

面的缓冲区构建：面的缓冲区构建原理同线的缓冲区构建原理

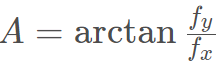
点缓冲区：给坐标点一个半径画圆即可。

线缓冲区：开始遍历每个顶点，若是开始点或者结束点，求该点以缓冲区半径向顺时针和逆时针选择90度的点，为圆弧的开始结束点。若为中间点，使用两个矢量的向量积来判断折点的凹凸性，若为内侧点，求出以转折点为原点，前后两个向量的方向角，对于凹侧和凸侧，分别使用不同的公式计算出平行线交叉点，若为外侧点，建立以转折点为原点，转折点到前进点为x轴的旋转坐标系，求出后面向量的方向角，对于凹侧和凸侧，在使用不同的公式求出圆弧的起始点和终止点。在弥合圆弧时，若转折点为凸点，顺时针弥合，若为凹点，逆时针弥合。

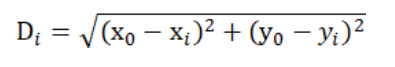
## 基于DEM的地形特征提取算法

**3.4.1算法原理**

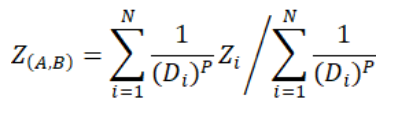
反距离加权法又称N-P法，是非规则分布点变成规则分布点常用的网格化方法之一。该方法的基本思想是离所估算的网格点距离越近的离散点对该网格点的影响越大，越远的离散点影响越小，甚至可以认为没有影响。在估算某一网格点的值时，假设离网格点最近的N个点对其有影响，那么这N个点对该网格点的影响与他们之间的距离成反比。坡度：坡度是法线与铅垂线之间的夹角利用坡度计算公式，坡向：坡向是法线在水平面上的投影与正北方向之间的夹角（顺时针度量）



 1.首先，需要计算所有离散数据点与所求网格点的距离，在二维平面空间，离散点（xi,yi）到网格（A,B）的距离Di为：



2.然后，需要找出离网格点（A,B）最近的N个离散点的距离，则网格点（A,B）上的估算值为：



注：其中，https://img-blog.csdnimg.cn/20190303144911230.png为离散点https://img-blog.csdnimg.cn/20190303144911264.png上的观测值，https://img-blog.csdnimg.cn/20190303144911236.png为网格点(A, B)上的估算值，https://img-blog.csdnimg.cn/20190303144911223.png为参与计算的样本个数，https://img-blog.csdnimg.cn/20190303144911233.png为插值点与第https://img-blog.csdnimg.cn/20190303144911229.png个站点间的距离，https://img-blog.csdnimg.cn/20190303144911254.png是距离的幂，一般取2。

坡度坡向：

1.导入DEM数据

2.将DEM数据处理为矩阵

3.可视化原始DEM

4.计算坡度

5.计算坡向

6.输出计算后的图形

## 网络分析中的最小代价生成树算法

**3.5.1算法原理**

对于一个给定的连通的无向图 G = (V, E)，希望找到一个无回路的子集 T，T 是 E 的子集，它连接了所有的顶点，且其权值之和为最小，我们把他称为最小生成树。Kruskal算法  
**俗称“加边法”，算法复杂度：O（ElogV）,其中E是边的数量，V是顶点数量，适合于边比较少的【稀疏图】。使用**Kruskal的避回路算法。Kruskal 算法提供一种在 O(ElogV) 运行时间确定最小生成树的方案。Kruskal 算法基于贪心算法（Greedy Algorithm）的思想进行设计，其选择的贪心策略就是，每次都选择权重最小的但未形成环路的边加入到生成树中。实现利用【并查集】合并和检查是否形成环。其算法结构如下：1、将所有的边按照权重非递减排序；2、选择最小权重的边，判断是否其在当前的生成树中形成了一个环路。  
如果环路没有形成，则将该边加入树中，否则放弃。

3、重复步骤 2，直到有 V – 1 条边在生成树中

**3.5.2实现步骤**

1,选择代价最小的边（A,C）；并保证AC不在同一颗树上然后合并AC

2,选择代价最小的边（D,F）并保证DF不在同一颗树上然后合并DF

3,选择代价最小的边（B,E）并保证BE不在同一颗树上然后合并BE

4, 选择代价最小的边（C,F）并保证CF不在同一颗树上然后合并CF所在的树5,选择代价最小的边A，D顶点AD在同一颗树上丢弃；选择最小的边（CD）顶点CD在同一颗树上丢弃；选择最小的边（BC）顶点BC不在同一颗树上加入此边然后合并BC所在的树此时所有顶点在同一颗树上，返回；

## 绘制Koch曲线

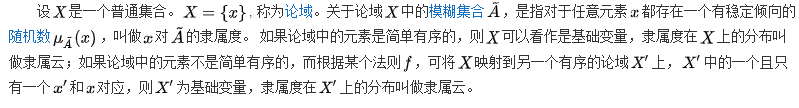
**3.6.1算法原理**

利用递归的思想，给每个线段找1/3点

**3.6.2实现步骤**

1.将给定线段(p1, p2)三等分。  
2.以三等分点s, t为顶点作出正三角形(s, u, t)。  
3.对线段(p1, s), 线段(s, u), 线段(u,t), 线段(t, p2)递归地重复进行上述操作

## 云模型



## K均值聚类

对于样本集。"k均值"算法就是针对聚类划分最小化平方误差：

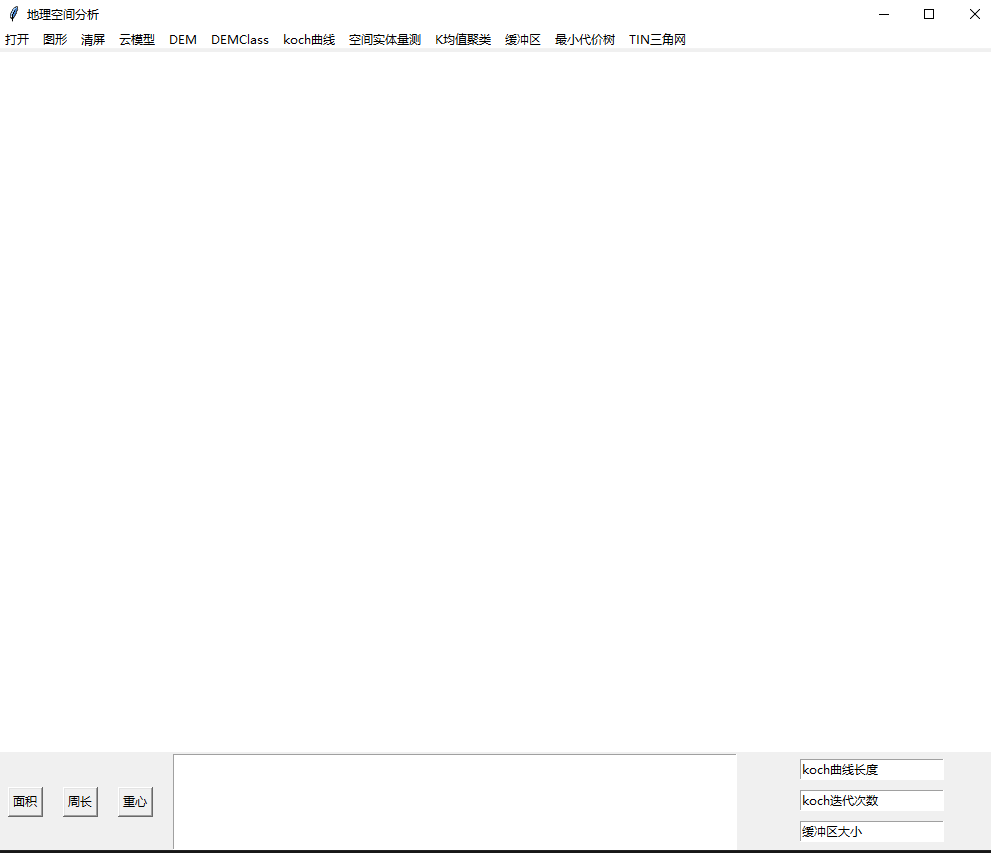
其中是簇Ci的均值向量。从上述公式中可以看出，该公式刻画了簇内样本围绕簇均值向量的紧密程度，E值越小簇内样本的相似度越高。

## 最小代价树

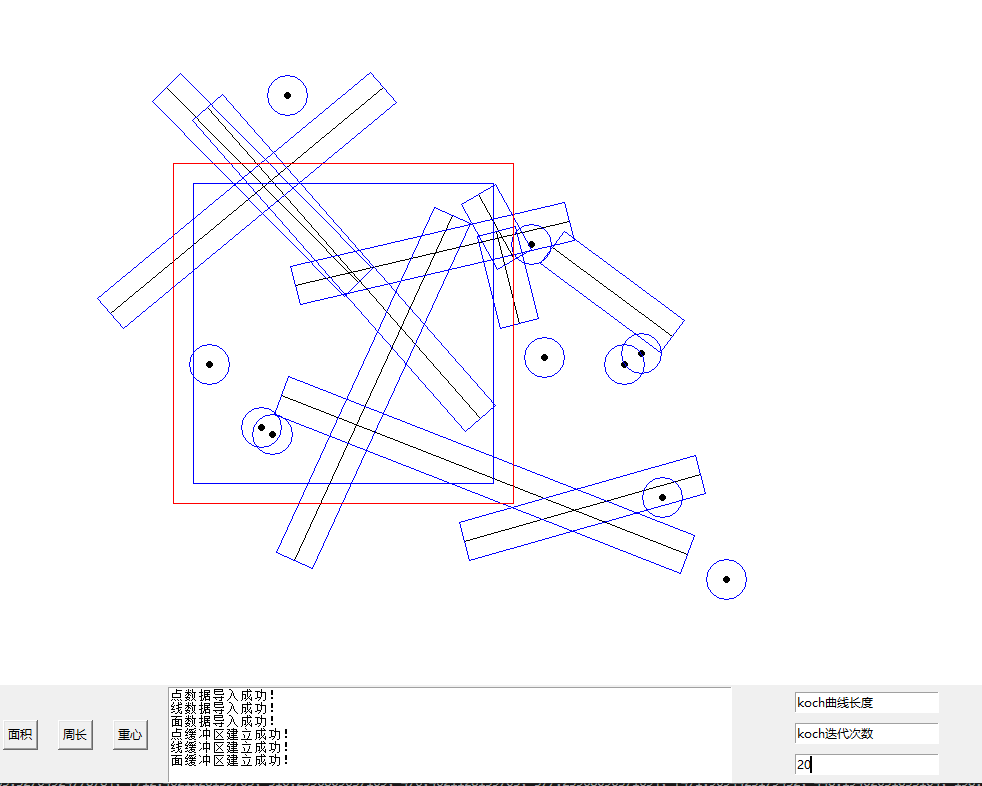
每次向最小代价生成树中T加入一条边的方法构造最终的最小生成树，按照边代价非递增顺序选取边。kruskal算法要求每次加入的没有环，可以用上次二叉树森林中集合表示，首先有n个点的集合（森林），如果选边<i,j>，如果i,j不在同一个集合中，可以合并，如果在同一个，则存在环。且每次选择最小的边，可以用最小堆来实现。

# 软件操作步骤及效果

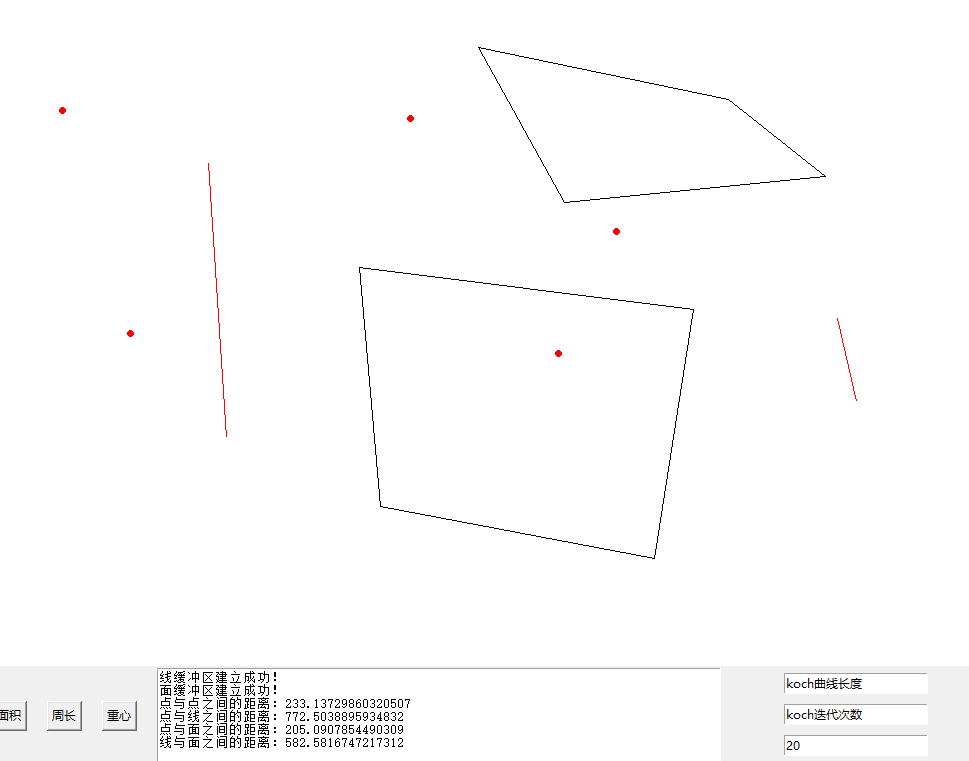
## 主界面



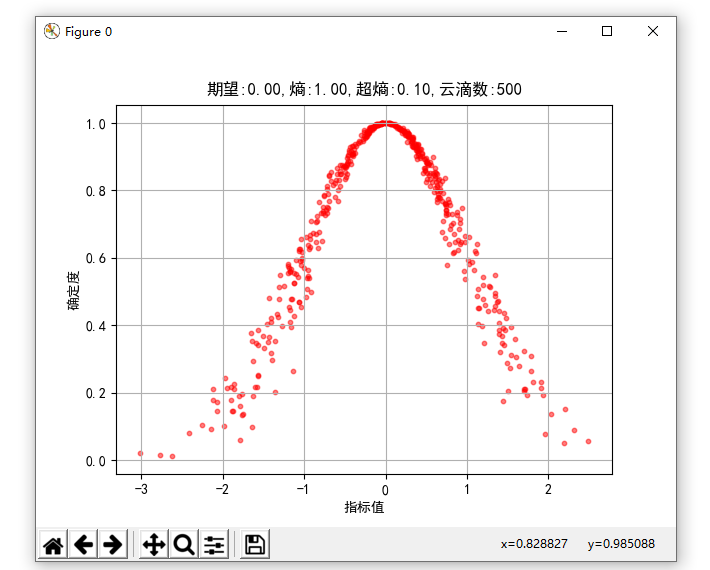
## 加载空间实体数据并绘制缓冲区

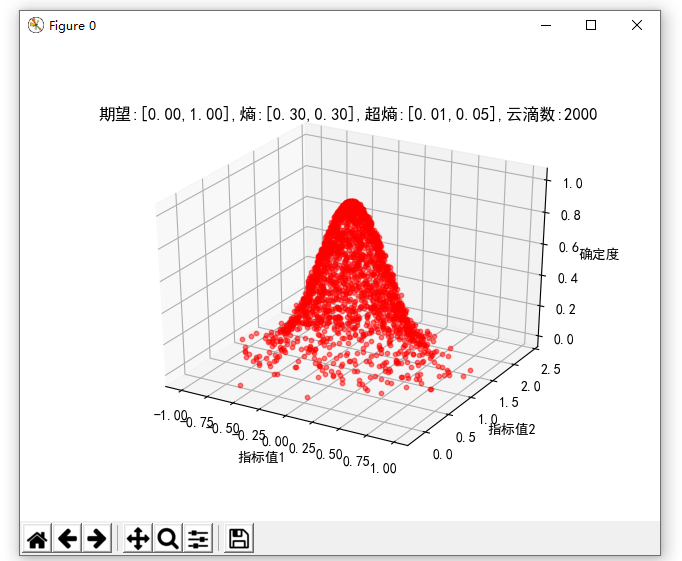


## 空间实体自由绘制

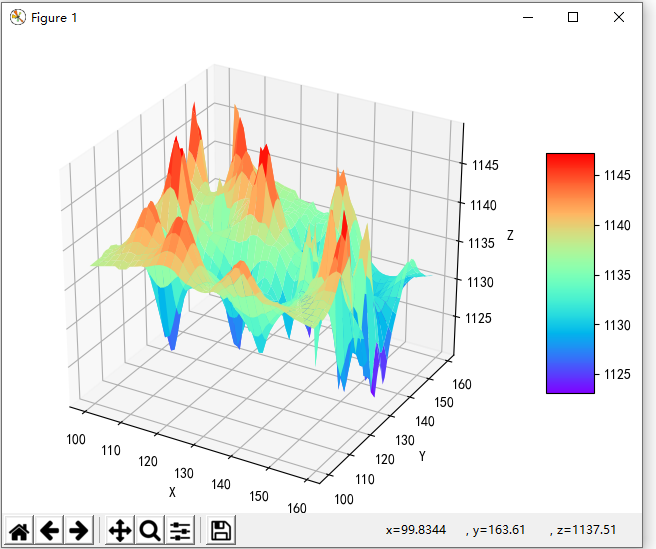


## 建立二维三维云模型

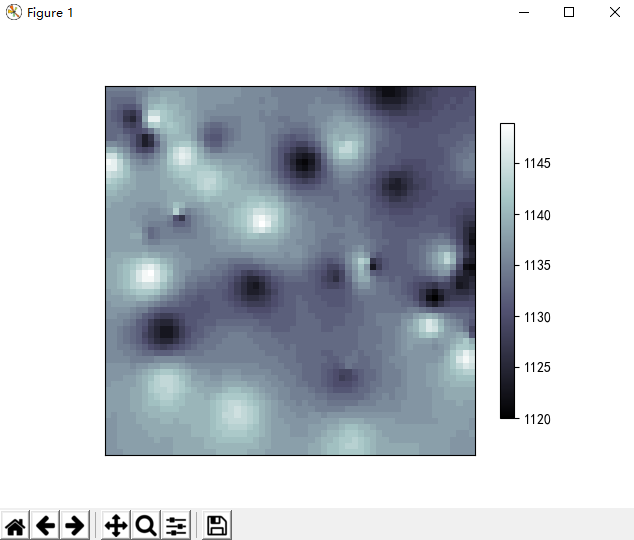


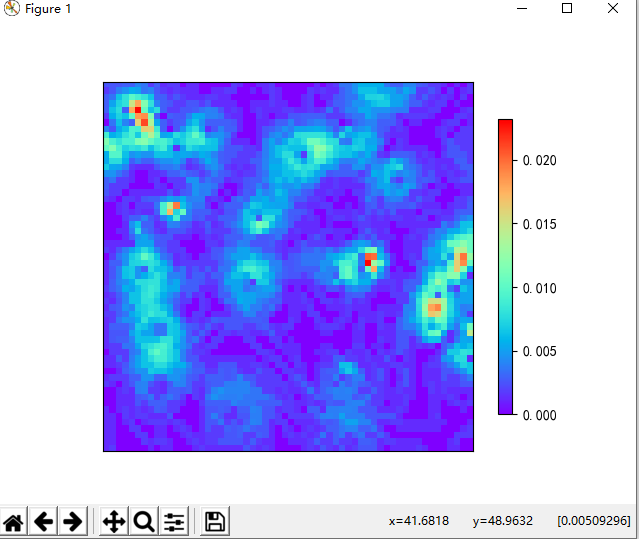


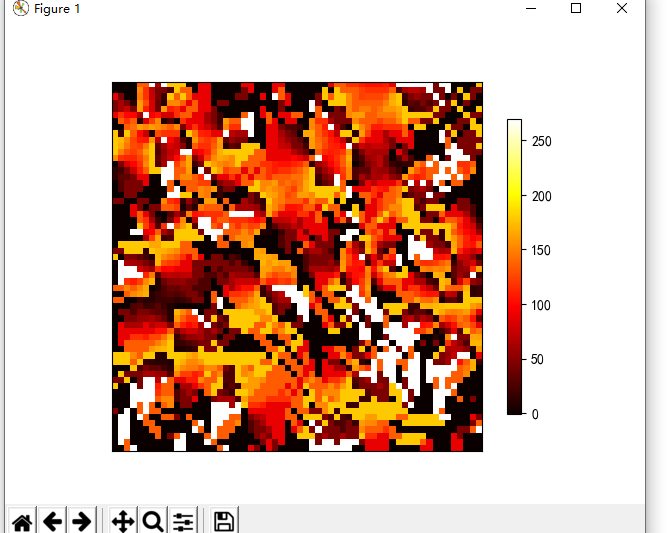
## 绘制DEM曲面图



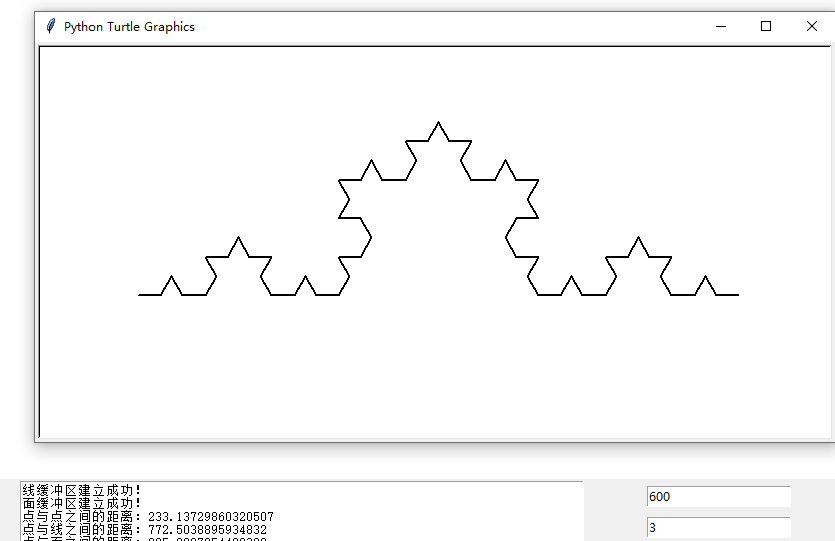
## 地形因子（坡度、坡向和二维DEM）的绘制



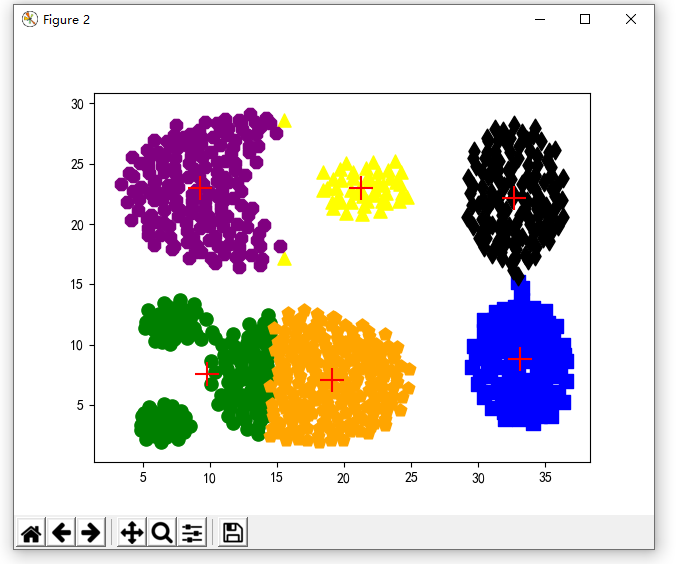




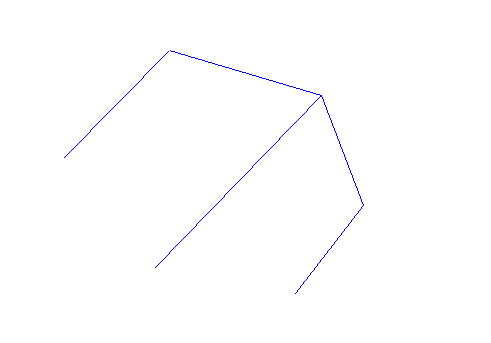
## 绘制koch曲线



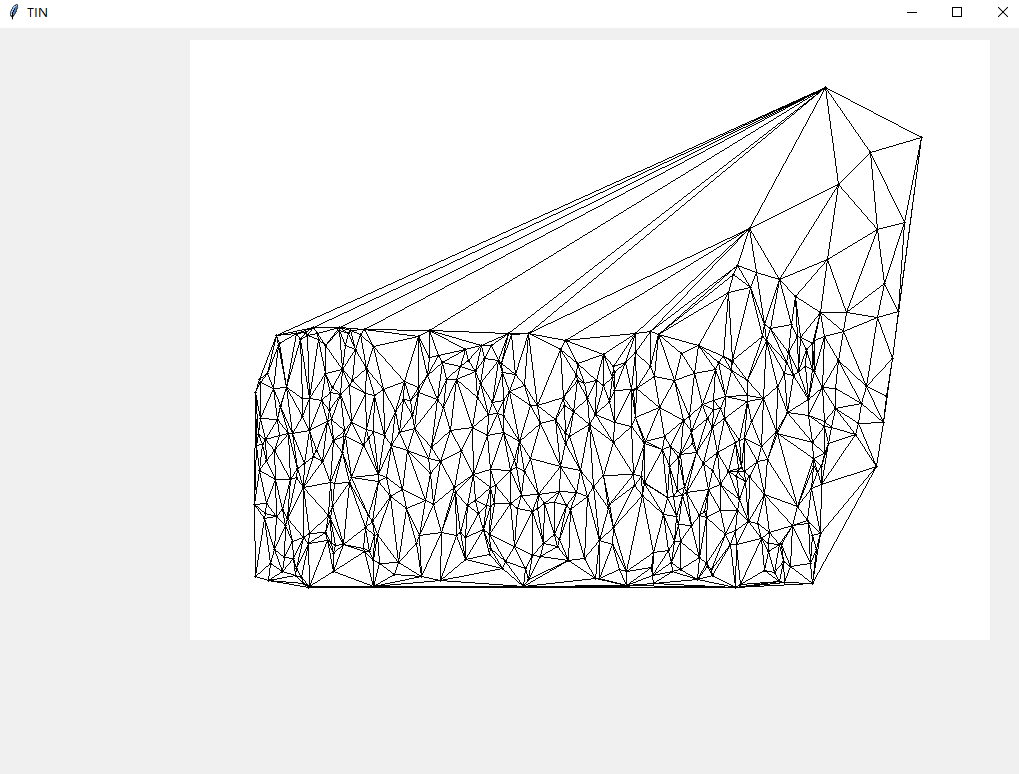
## K均值聚类



## 最小生成素算法



## 三角网生成



# 实习总结

在这短暂而又漫长的几天实习中，我们在相同的兴趣下一起训练，一起学习，在这当中我们学到了很多，也提高了很多。在实习过程中，我认识了许多以前很少听说和接触的算法，虽然还不是很深透的理解，但是在不断地钻研和请教中编程能力的确有子很大的提高。  
在编程的时候，面对每个问题都需要自己想出合适的思路和算法。在这里，我也要感谢各位一起实习的同学和老师给我们仔细地讲解算法原理与解决遇到的问题。