**Python语言与安全程序设计**

**作业报告**

学生姓名 王承淳

学 号 08183004

专 业 信息安全

任课教师 朱长征

目录

[1 题目 3](#_Toc71750237)

[2 设计思路 3](#_Toc71750238)

[2.1 分形的相关概念 3](#_Toc71750239)

[2.2 图像绘制 3](#_Toc71750240)

[2.3 绘制过程控制 3](#_Toc71750241)

[2.4 多线程 3](#_Toc71750242)

[2.5 慢动作视觉效果 4](#_Toc71750243)

[3 开发配置环境 4](#_Toc71750244)

[4 运行使用方法 4](#_Toc71750245)

[5 运行效果截图 5](#_Toc71750246)

[5.1 koch动态 5](#_Toc71750247)

[5.2 koch多线程 5](#_Toc71750248)

[5.3 koch优化（最终版） 6](#_Toc71750249)

[5.4 IFS（树叶形状） 6](#_Toc71750250)

[5.5 IFS植物形态曲线 7](#_Toc71750251)

[5.6 IFS脑分形曲线 7](#_Toc71750252)

[6 运行效果分析总结 7](#_Toc71750253)

# 题目

多线程绘制经典两种以上的分形曲线，至少绘制koch雪花曲线，IFS植物形态曲线/脑分形曲线，加慢动作效果。分形资料参考<https://www.douban.com/note/230496472>

# 设计思路

题目概括为“多线程绘图加慢动作”。基本问题是分形是什么以及如何画出，之后考虑如何体现“绘制过程”，在此基础上添加多线程。如果需要绘制的是线段，再将线段分成多段以达到视觉上的慢动作。

## 分形的相关概念

分形，具有以非整数维形式充填空间的形态特征。通常被定义为“一个粗糙或零碎的几何形状，可以分成数个部分，且每一部分都（至少近似地）是整体缩小后的形状”，即具有自相似的性质。分形（Fractal）一词，是芒德勃罗创造出来的，其原意具有不规则、支离破碎等意义。1973年，芒德勃罗（B.B.Mandelbrot）在法兰西学院讲课时，首次提出了分维和分形的设想。

阅读所给资料相关代码，题目要求绘制的分形，即通过递归、迭代、随机等方法，来模拟出图像点线，从而达到相应的视觉效果。

## 图像绘制

图像的绘制过程可以使用turtle和matplotlib，为了实现图像的精确控制，采用matplotlib来进行绘图。

## 绘制过程控制

通过查询资料，采用animation模块进行动画的模拟。

## 多线程

使用python自带的threading库来进行线程控制。

## 慢动作视觉效果

对于点图来讲，快慢可以用sleep控制。但线图在绘制时会有明显的断层，为了曲线的平滑过渡，将一条线段拆成多份可以在视觉上解决这个问题。

# 开发配置环境

操作系统：Windows10 家庭中文版（64位）

处理器：Intel(R) Core(TM) i5-6200U CPU @ 2.30GHz 2.40GHz

RAM：16.0GB

显示：NVIDIA GeForce 940M独立显示芯片，2GB独立显存

Python版本：Python 3.7.6

所需库：matplotlib==3.3.4 numpy==1.19.2

# 运行使用方法

在对应软件目录使用pip install -r requirements.txt安装所需库。

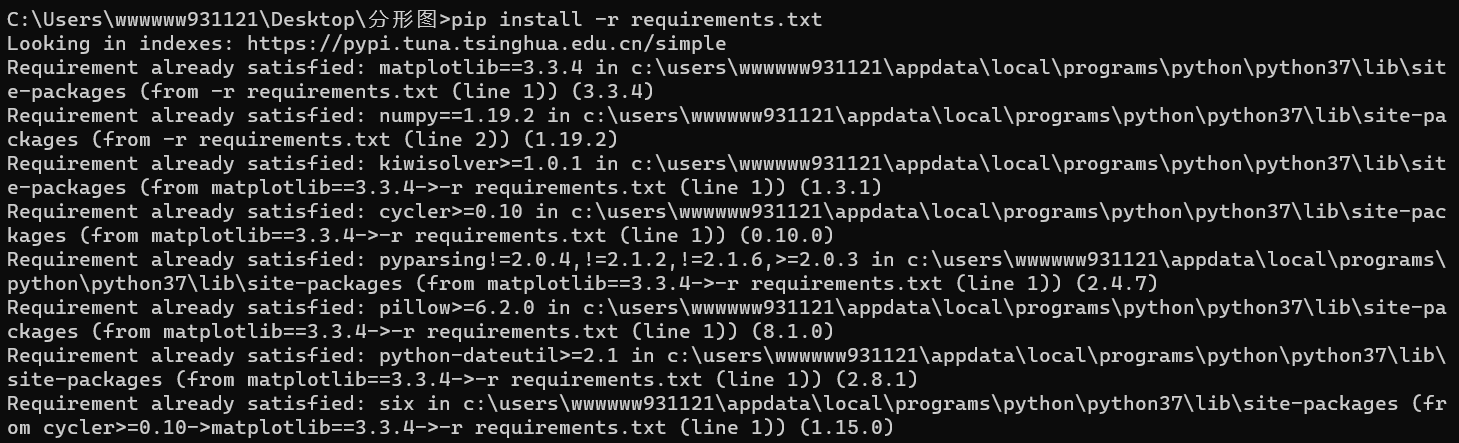


图 4.1 依赖安装

在命令行使用python [文件名]即可运行程序。

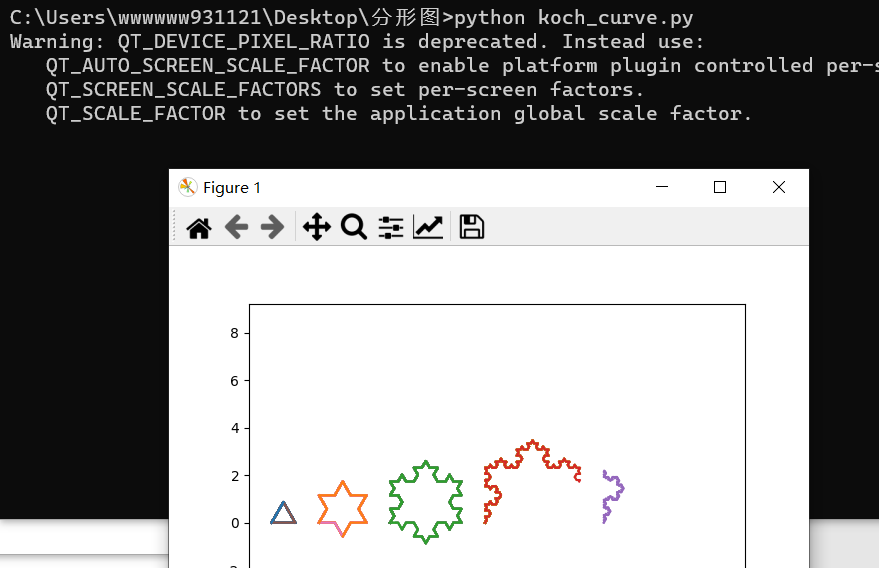


图 4.2 程序运行

# 运行效果截图

## koch动态

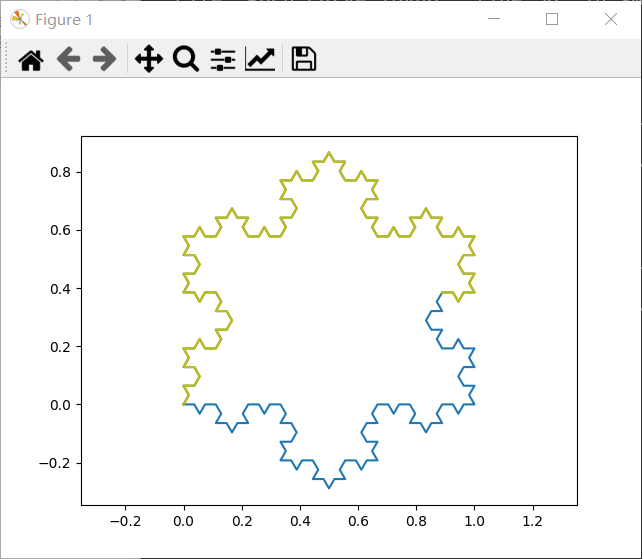


图 5.1 koch动态闪烁绘图

## koch多线程

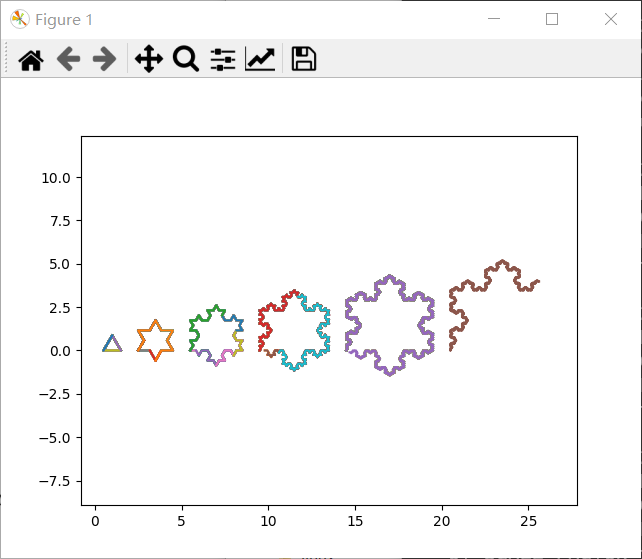


图 5.2 koch动态闪烁多线程绘图

## koch优化（最终版）

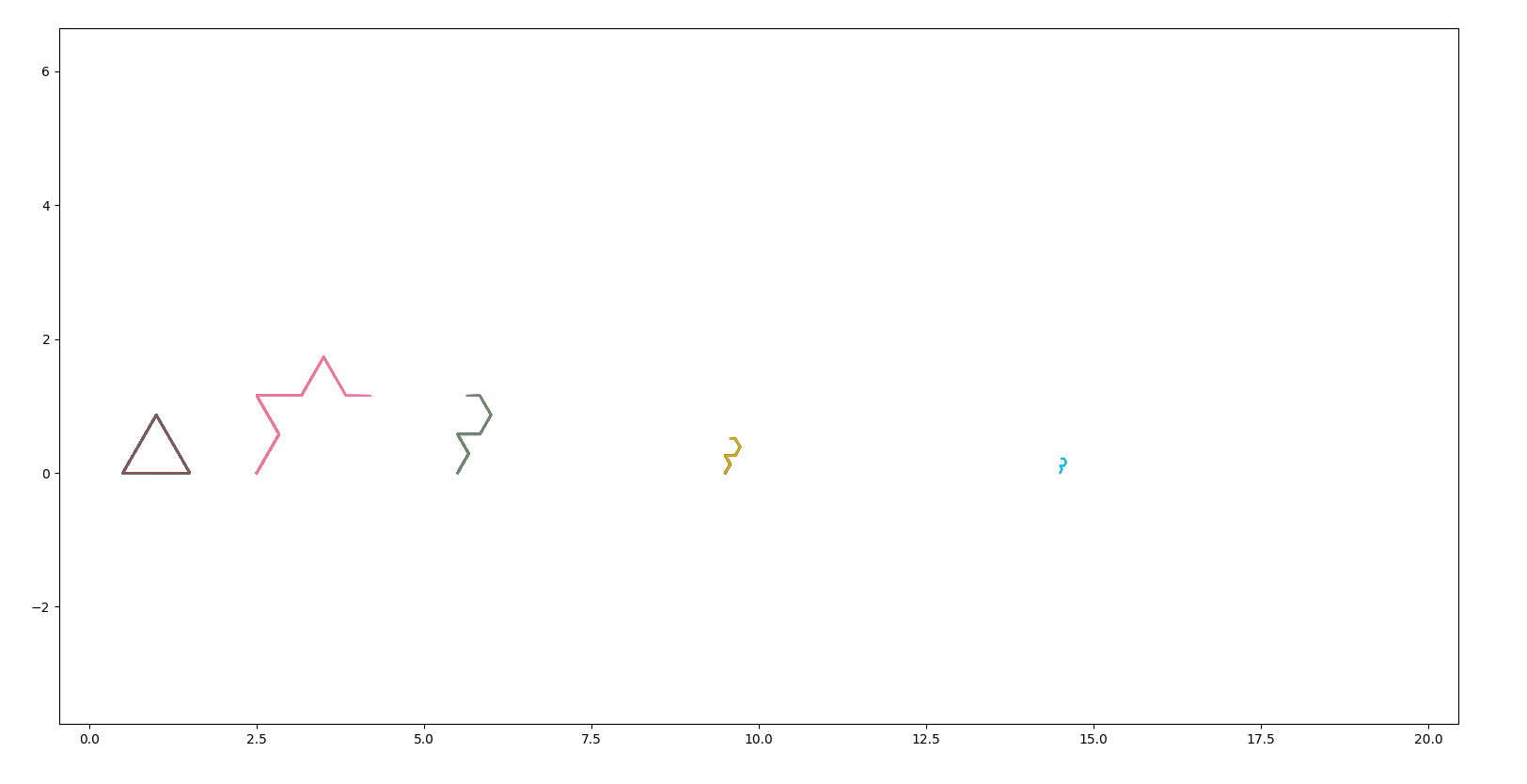


图 5.3 koch动态闪烁多线程绘图加平滑过渡

## IFS（树叶形状）



图 5.4 IFS（树叶形状）曲线

## IFS植物形态曲线

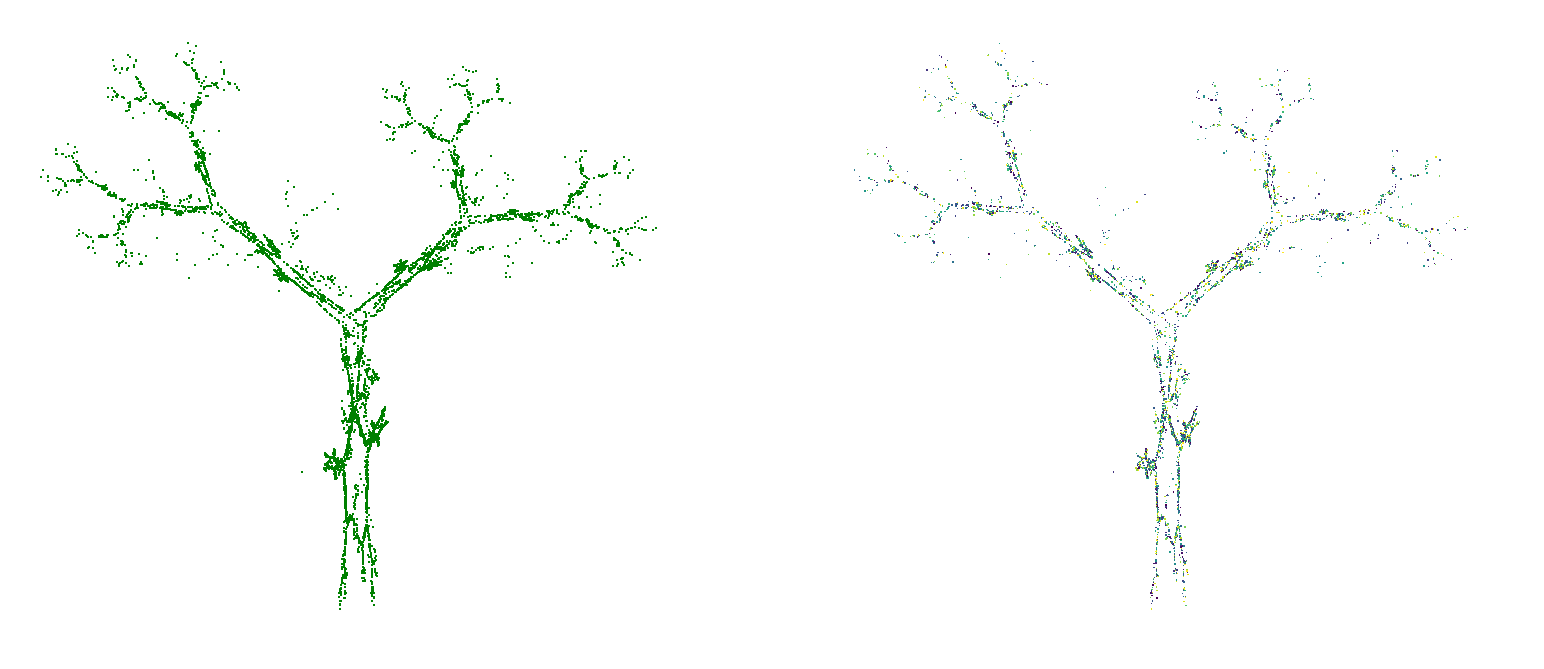


图 5.5 IFS植物形态曲线

## IFS脑分形曲线

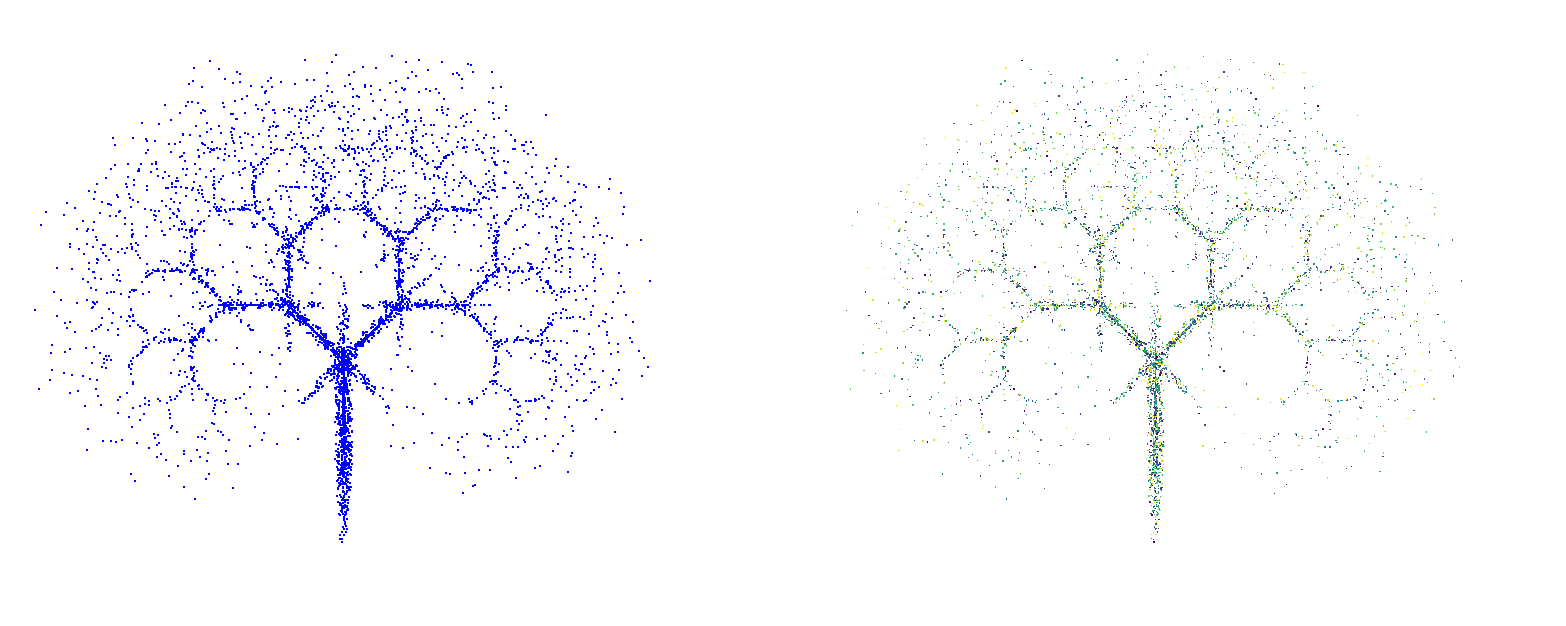


图 5.6 IFS脑分形曲线

# 运行效果分析总结

程序中中，公共数据表示当前绘制点位，多个线程控制点位变换，animation线程每次更新数据时重新按照0到当前下标渲染整个图。

在最后，尝试了另一种方法，修改brain.py得到结果图 6.1，左边是每次渲染一个点，结果发现此时绘制过程慢很多。

经过研究，由于animation的绘制是以一定速度更新的，所以其他线程自由运行，修改控制数据，animaton则以设定间隔进行绘制。

实际上有一种更好的方法，即不采用animaton，直接生成线程池，线程内用pause和clear来进行动画控制。由于写程序时一心只想用animaton，所以没有想到。

由于之前做手术请了一个月假，又在DDL时写Python安全作业，网络攻防、攻防实践结课报告赶不完了TAT。

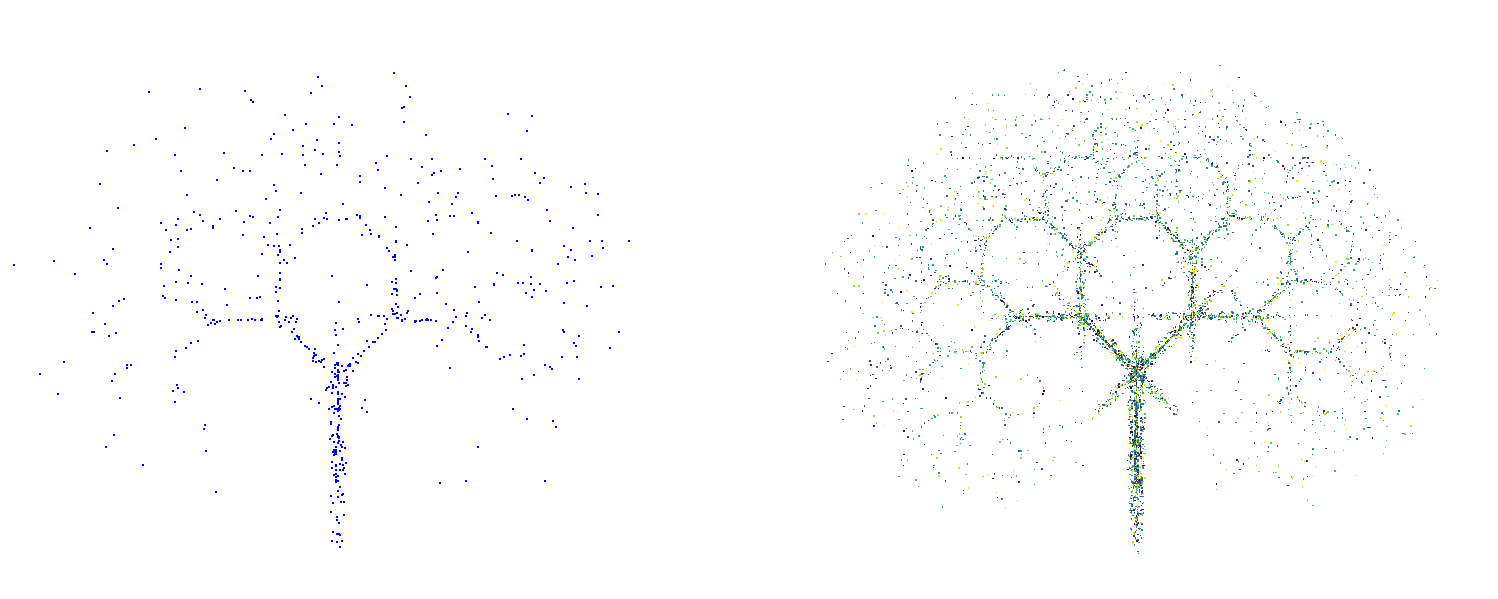


图 6.1 IFS脑分形曲线渲染对比

注：Python采用GIL全局解释器锁，单位时间多个核只能跑一个线程，然后时间片轮转。