

计算物理 作业报告3

PB14203209 张静宁 2017.10.4

第三题

进行单中心扩散限制凝聚 (Diffusion-limited Aggregation, DLA)模型的模拟，并用两种方法计算模拟得到的DLA图形的分形维数，求分形维数时需要作出双对数图。

算法思路

二维DLA模拟

取一个正方形点阵，点阵中心放置一个种子粒子作为凝聚中心，在远离种子的正方形边界上随机产生一个粒子，使其做随机行走。当粒子走到与团簇相接触时，就被粘住不动，成为团簇的一部分；当粒子越过远离中心的逃离边界时，粒子就逃离了。无论粒子是粘住了，还是逃离了，都将重新随机产生一个粒子，重复上述过程，这样就可以得到足够大的 DLA 团簇。

画布和团簇

用一个二维数组(*particle_set*)来存放团簇粒子的坐标，我们采用一个和画布一样大的二维数组，数组的标号和画布上的坐标对应。如`particle_set[5][5] = 1`，表示画布上坐标(5,5)的点有一个团簇粒子，如果`particle_set[5][5] = 0`，则表示该点没有团簇粒子。因此我们的画布是`[0, RANGE]*[0, RANGE]`，`RANGE`定义了画布的大小。

边界

模拟程序涉及如下四个边界（从小到大）：

1. 当前团簇粒子的边界
2. 随机生成粒子的边界
3. 判断逃离的边界
4. 画布边界

在主程序中，我们让随机生成粒子的边界和判断逃离的边界重合，将覆盖住团簇的最小正方形作为团簇的边界。为了避免团簇碰到生成粒子的边界，而造成类似晶体长到烧杯上的情况。我们将随机生成粒子的边界设置成动态的，始终比团簇边界大一圈，如距离为50个单位长度。除此之外，我们还必须把团簇边界设定在画布边界以内，如两边距离100个单位长度。

随机行走函数

我们采取二维四个方向的随机行走，随机数的值 $\{0, 1, 2, 3\}$ 对应上下左右四个方向。每次随机行走的步长为画布上一个单位长度。输入粒子的(x, y)坐标，调用一次函数走一步，将改变粒子坐标。

随机产生粒子函数

用来在正方形边界上随机产生粒子。

判断粘住或逃离函数

判断当前位置的粒子是否被粘住成为团簇的一员，或者越界逃离，这两种情况都需要生成新的粒子，故返回0。如果并没有被粘住或者越界，将继续随机行走，返回1。

程序使用说明

编程环境：Ubuntu，要预装 C 和 Python3

- dla.c C 语言编写的计算程序，负责计算并输出数据点
- dla 编译后在Linux系统可执行的文件
- plot.py Python 编写的绘制DLA模型图片的脚本
- dynamic_image.py Python 编写的绘制DLA模型动图的脚本
- sandbox.py 用sandbox方法计算分形维数，并绘制双对数图
- boxcounting.py 用盒计数法计算分形维度，并绘制双对数图

分别执行以下三条命令：

```
$ gcc dla.c -o dla          # 编译 dla.c 程序
$ ./dla > data              # 在当前目录下执行 chaos 程序，并将结果输出到 data 文件
$ python plot.py            # 执行绘图程序，将生成的图片保存为 pdf 和 png 格式，同时复制数据
$ python3 dynamic_image.py  # 执行绘图程序，生成动图保存为 gif 格式，同时复制数据
```

在执行第二条命令后会输出如下信息，提醒用户输入相关参数：

```
Please input how many particle you want(Positive integer): 11000 # 输入想得到的团簇粒子数量
Totally 11000 particles !
```

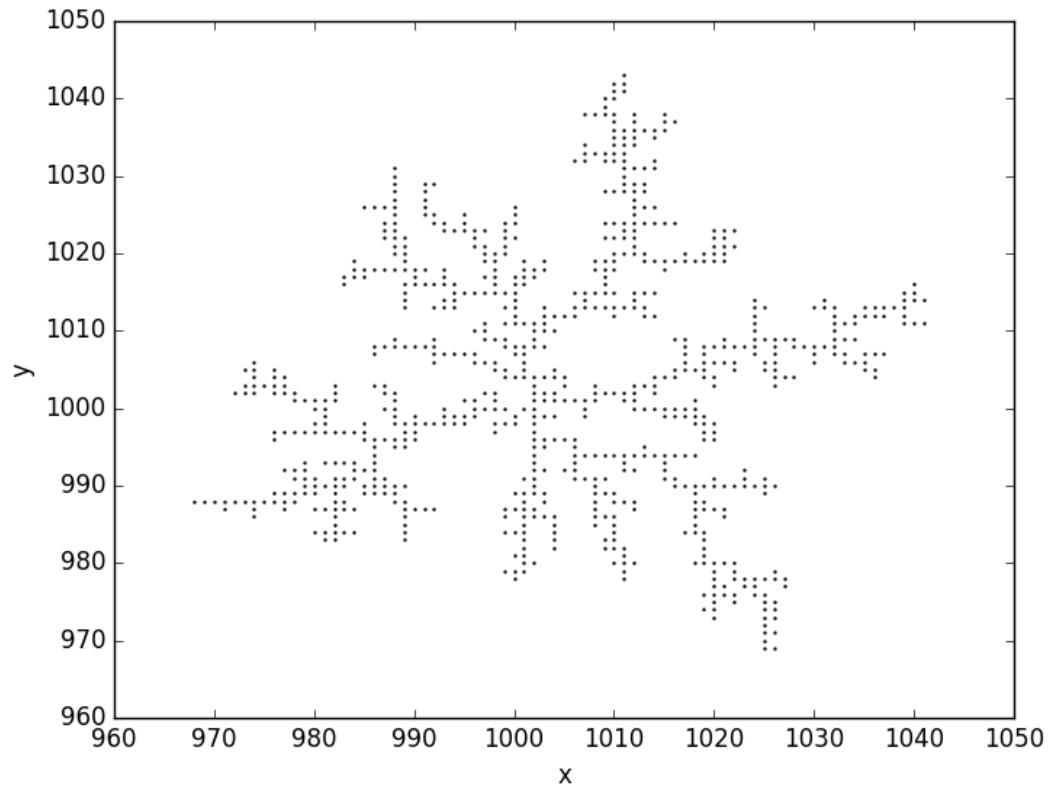
在执行第三条命令后，可以在results文件夹中得到三个文件：

- *.txt 当前条件下得到的数据文件
- *.pdf 当前条件下得到的图片
- *.png 当前条件下得到的图片

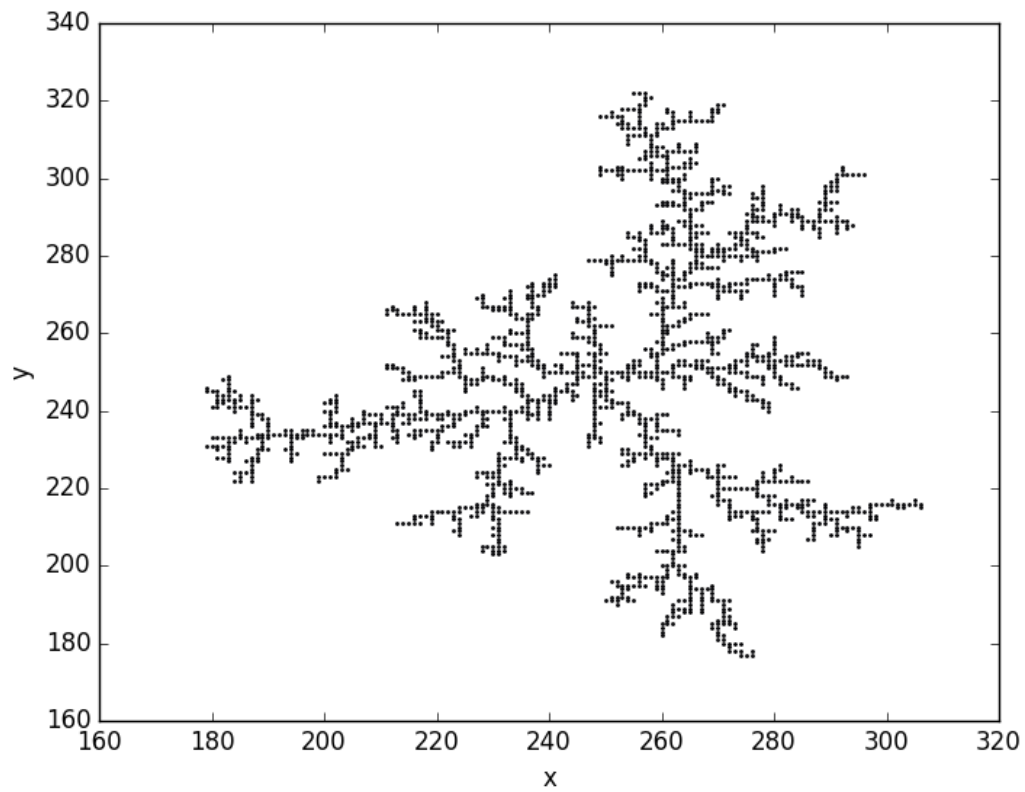
单中心DLA模型模拟

设置团簇粒子数量，从少到多的结果。

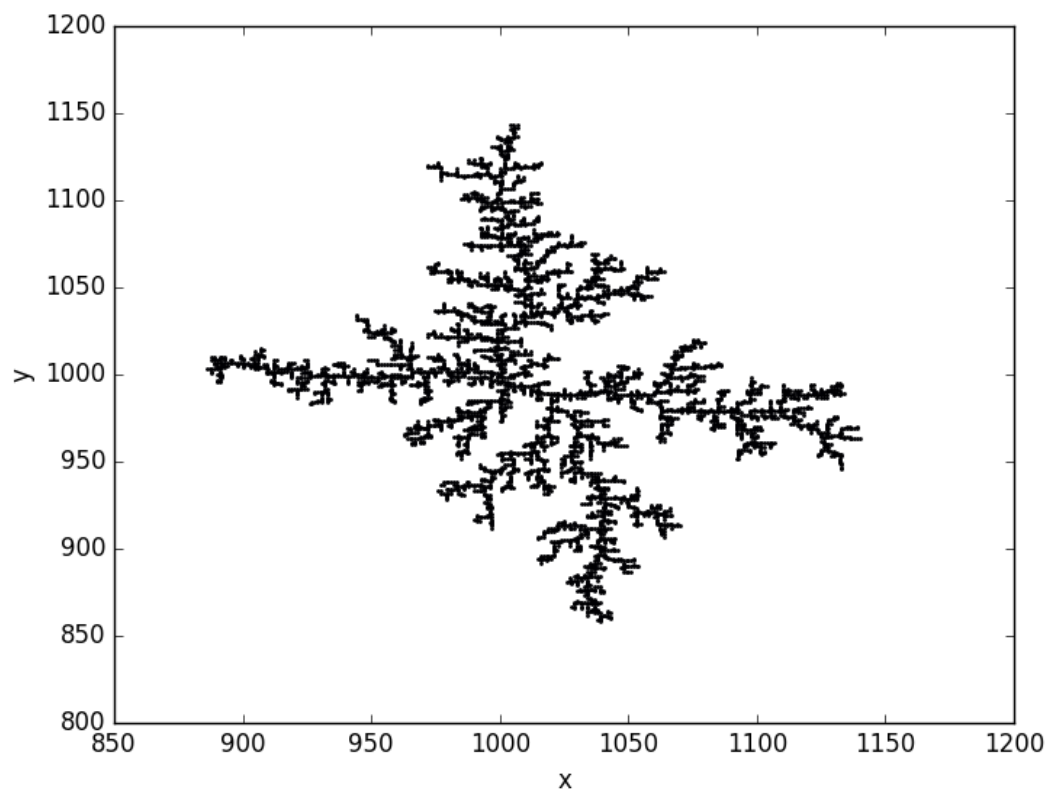
800 particles!



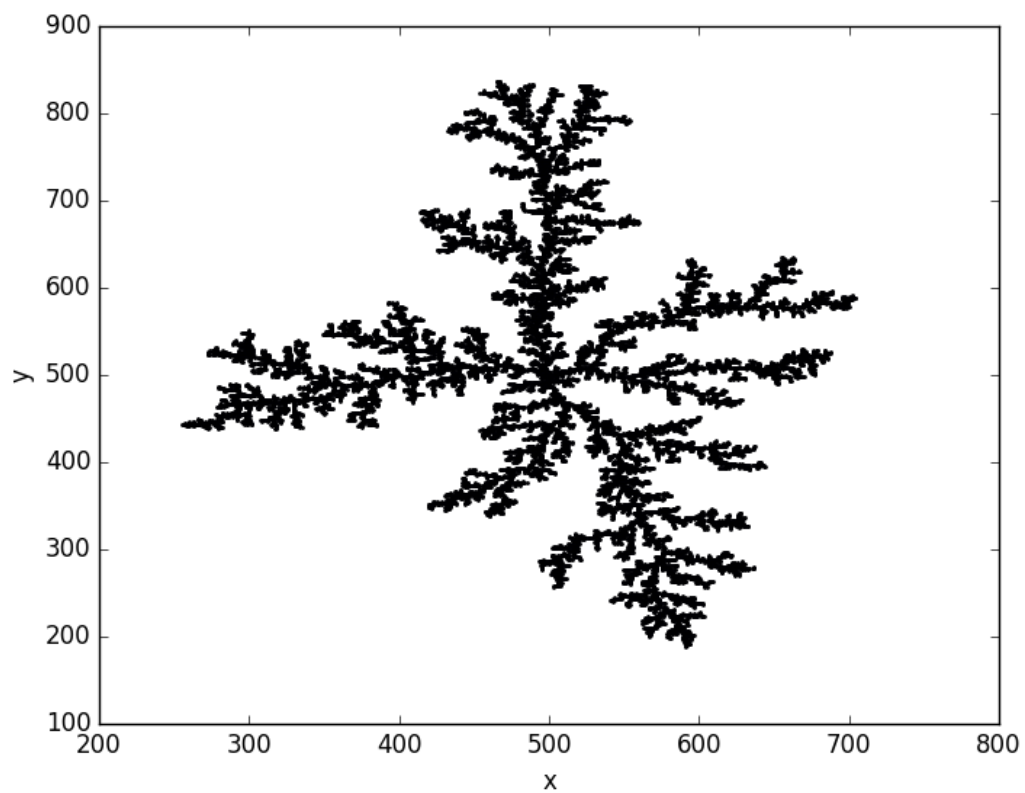
2000 particles!



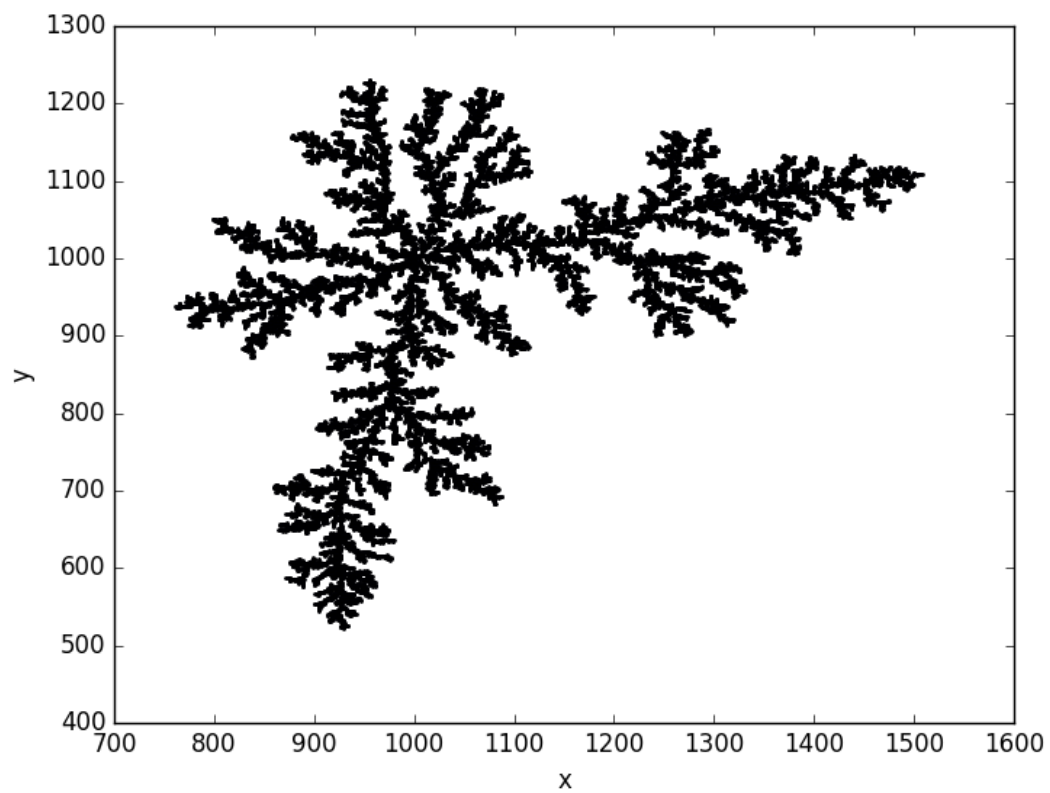
5000 particles!



18000 particles!

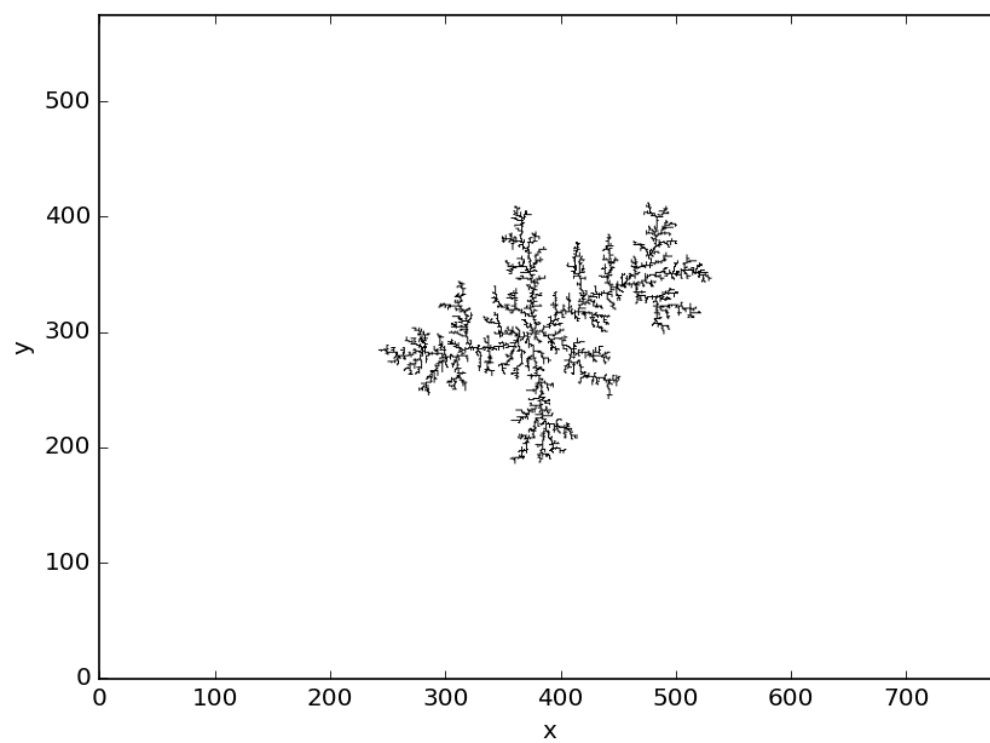


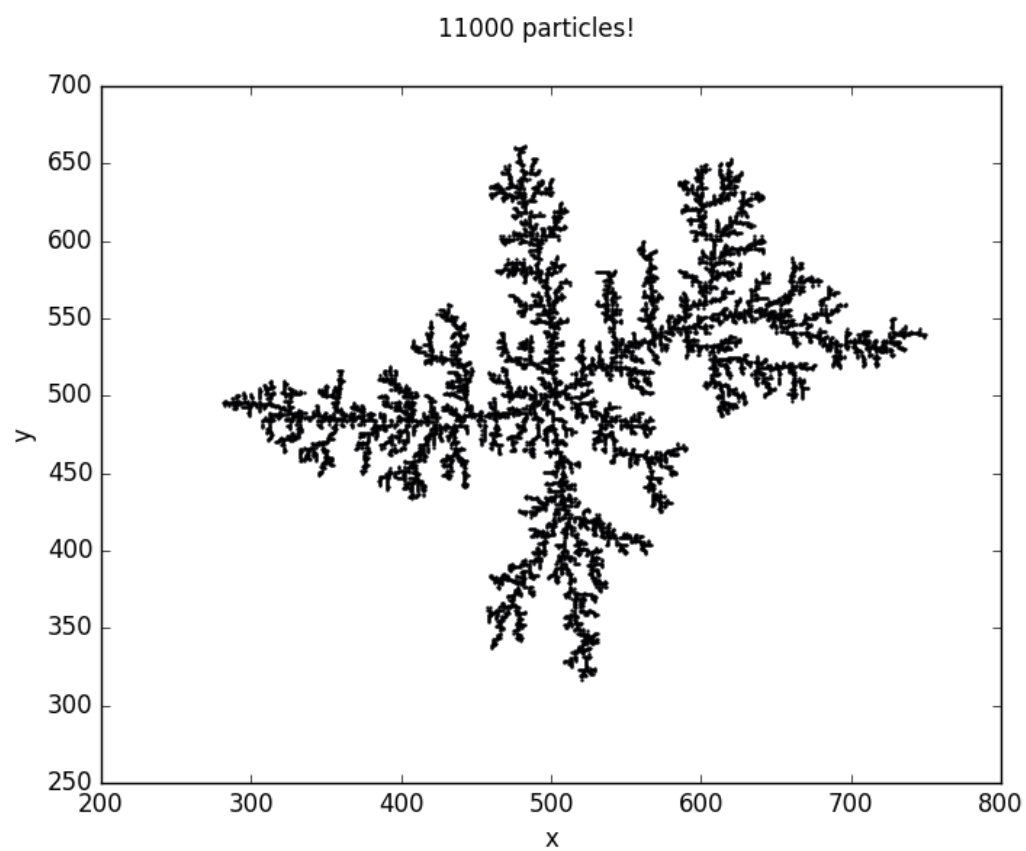
30000 particles!



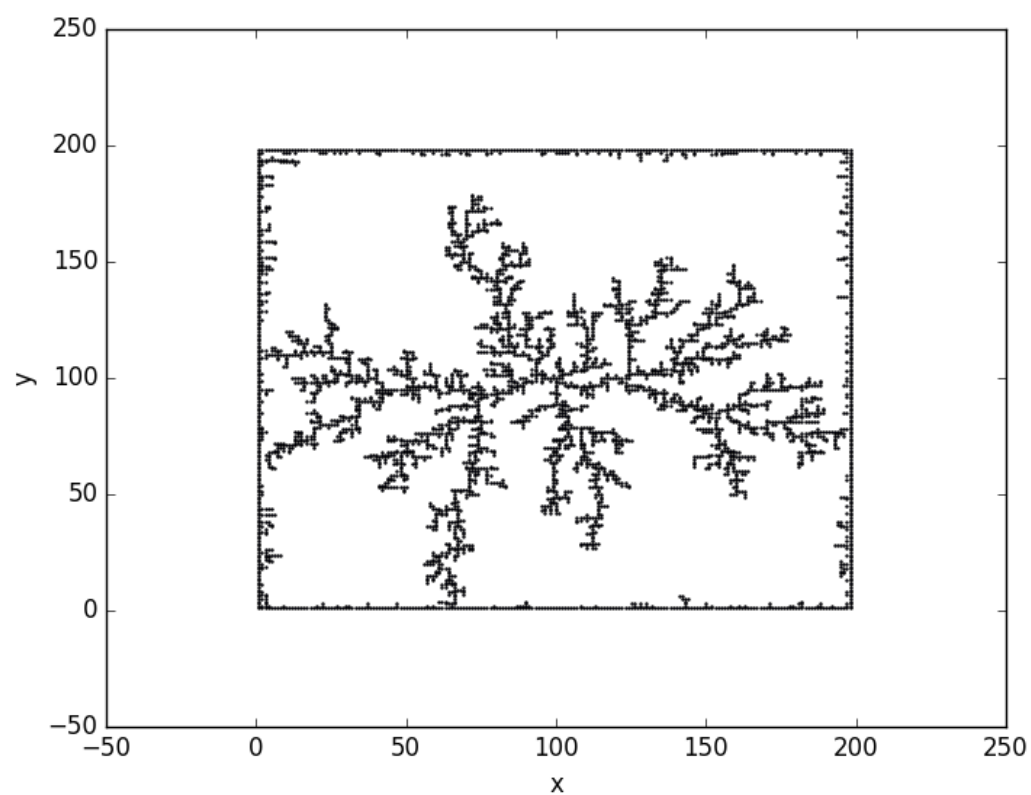
看DLA模型系统演化的动图：

Totally 11000 particles!



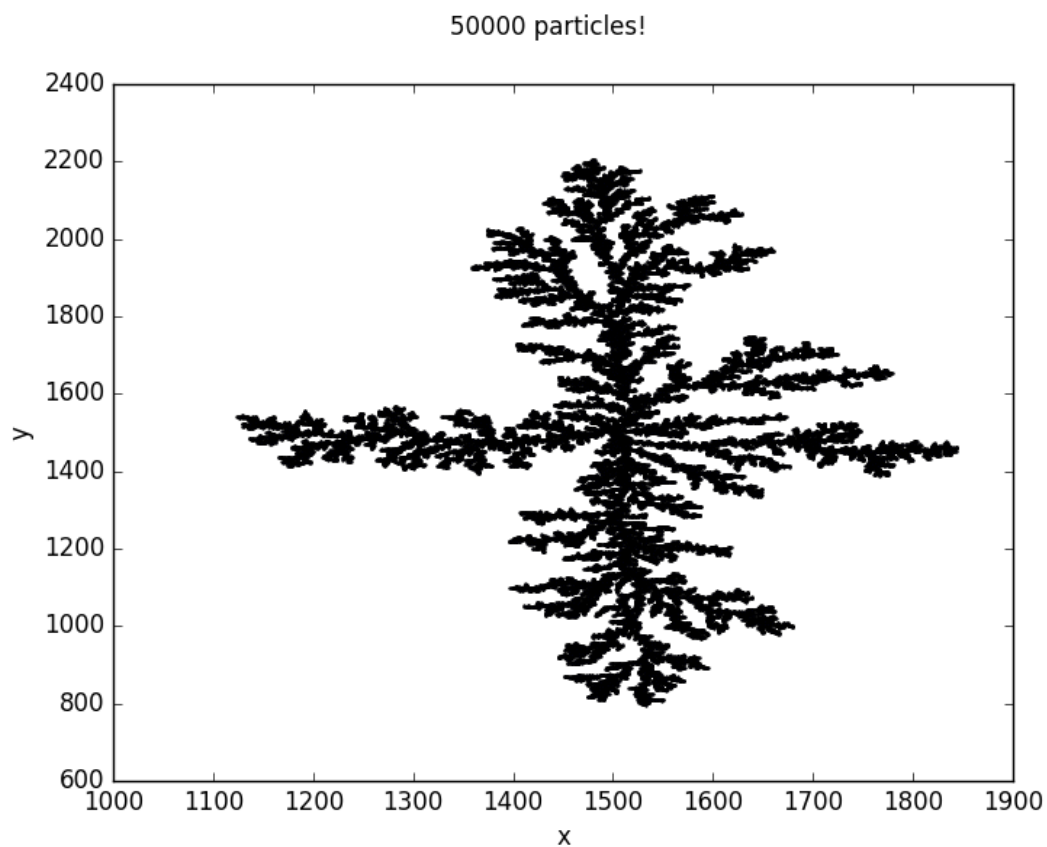


一开始没处理好边界关系（将随机生长边界设置为定值），出现团簇生长碰到生成粒子的边界，导致沿着边界生长的情况，如下：

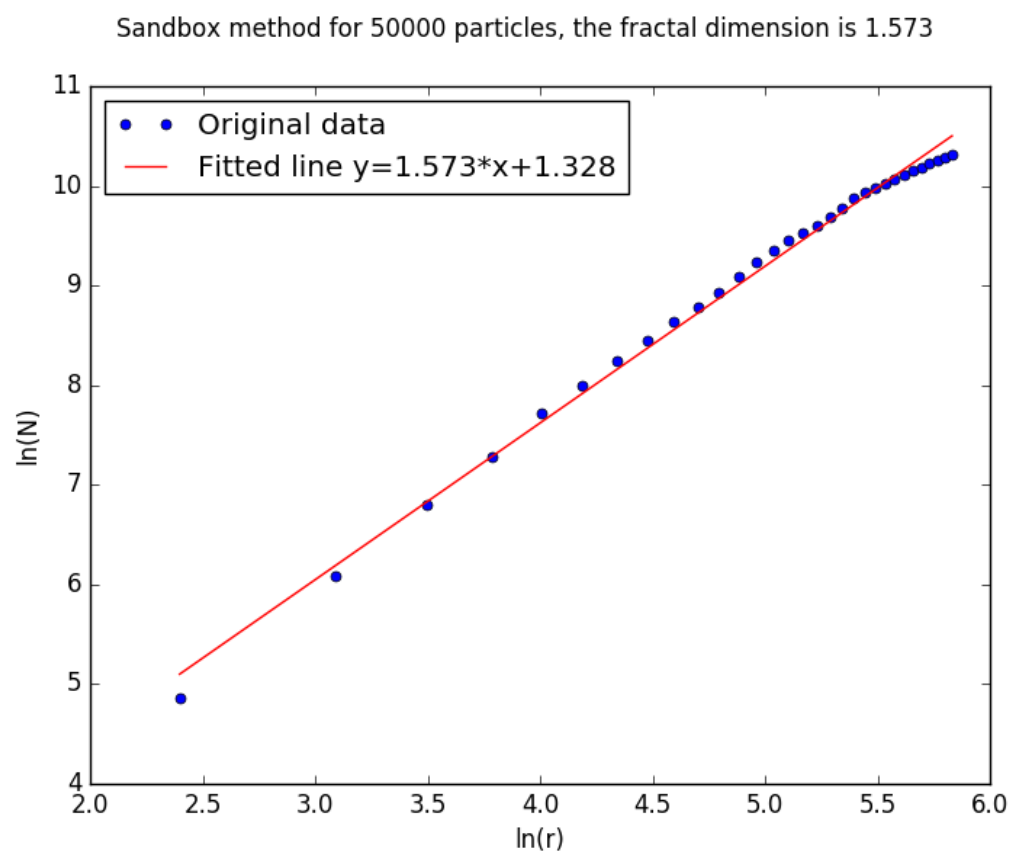


计算分形维数

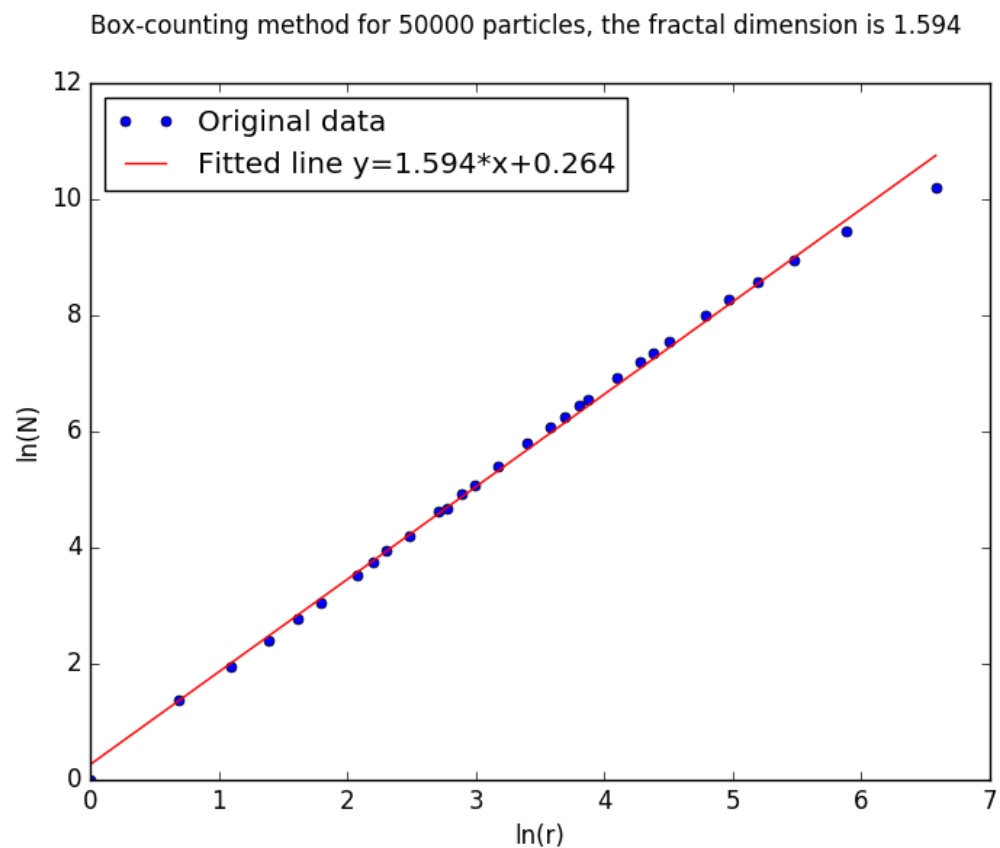
我采用两种方法：**Sandbox**法和盒计数法，分别计算如下50000个团簇粒子生成的DLA图形的分形维数。



Sandbox方法



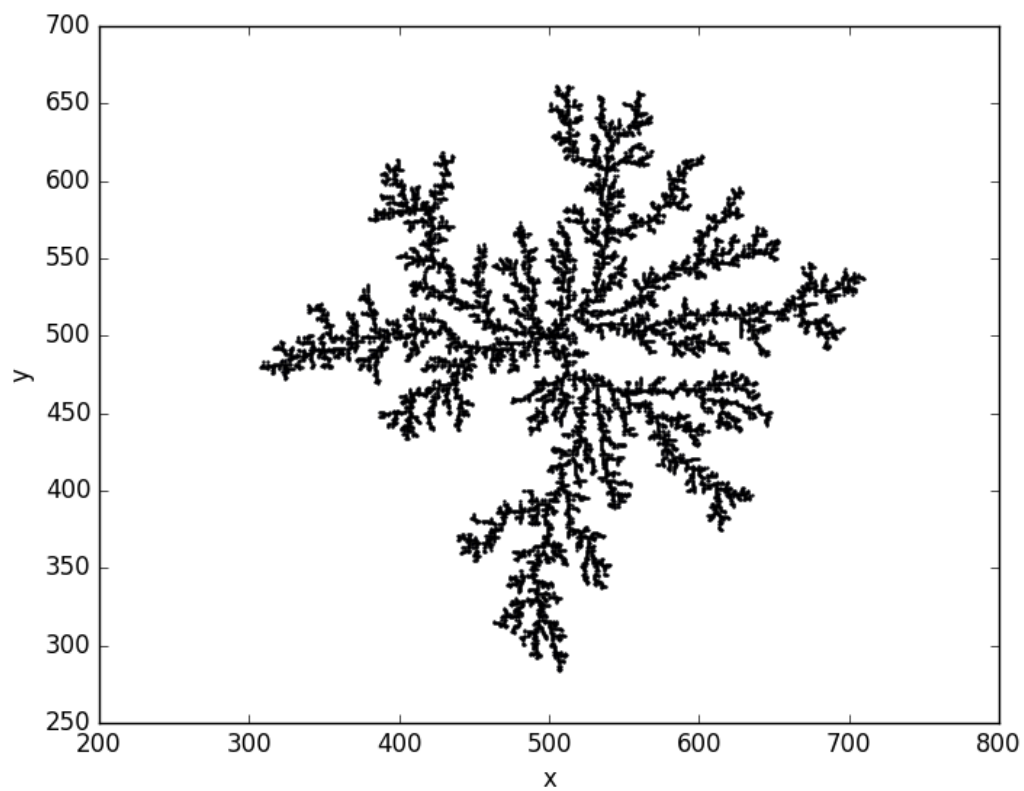
盒计数法



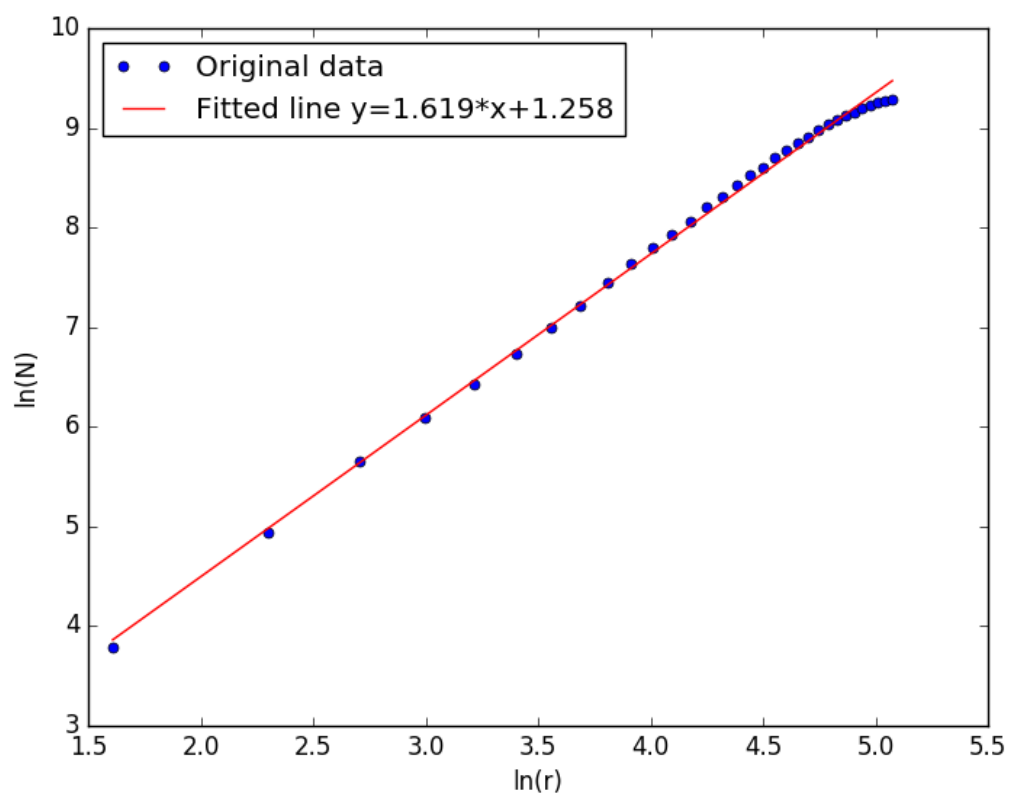
Sandbox方法计算得到的图形分形维数为1.573，盒计数法计算得的分形维数为1.594，相差0.02，较为相近。

再分别计算如下12000个团簇粒子生成的DLA图形的分形维数，两种方法得到的分形维数相差0.007。

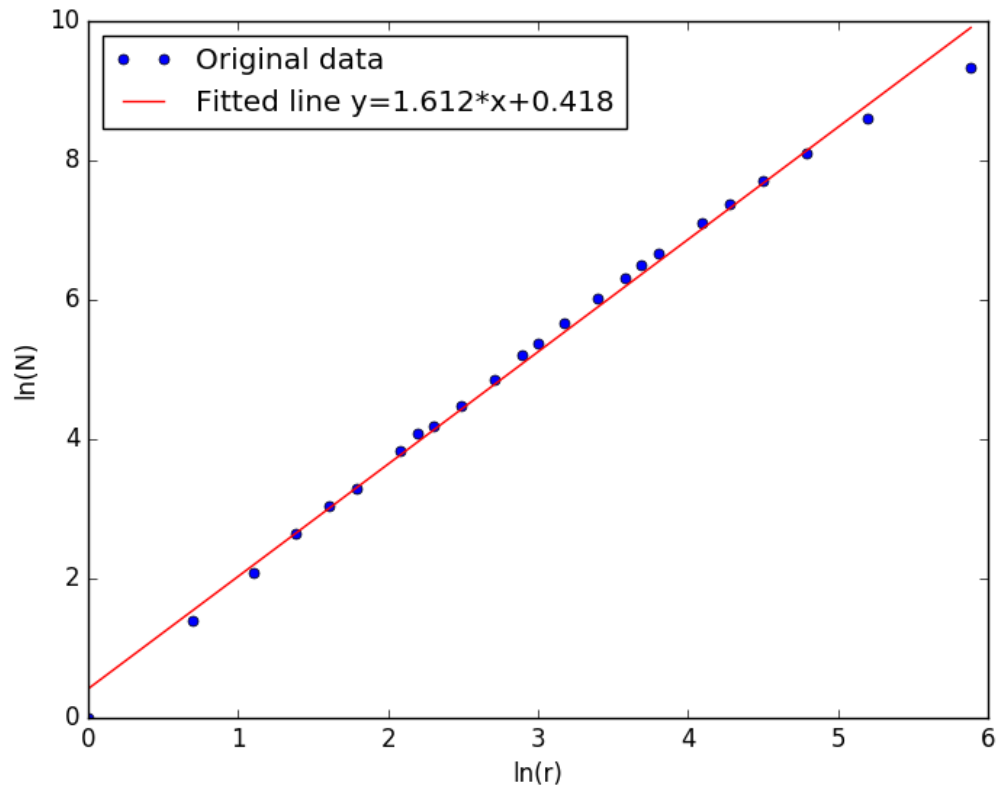
12000 particles!



Sandbox method for 12000 particles, the fractal dimension is 1.619



Box-counting method for 12000 particles, the fractal dimension is 1.612



总结

- 写模拟的DLA模型主程序时，难点在于几个边界关系的处理，比较琐碎、需要注意很多的细节。
- 生成伪随机数的时候，注意要用到不同的种子，否则每次的结果都一样。虽然每次得到的图形都不同，但外观上都是分叉的形状，很相似。
- 要做模型演化的动图时，还需要额外学习画动图的方法，也会花去不少时间。
- 另外在学习用两种不同的方法计算所得到的DLA图形的分形维数时，也需要对数据进行较多的处理，不同的方法处理方法不同，都会花去较多时间写程序。