

第 18 题作业报告

PB18000341 范玥瑶

A. 作业题目

进行单中心 DLA 模型的模拟（可以用圆形边界，也可以用正方形边界），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。

B. 算法及主要公式

B.1. 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，在远离种子的任意位置（可设置一个圆形边界或正方形边界，边界距中心的距离远大于想要模拟得到的图形尺寸）随机产生一个粒子，使其做随机行走，当粒子走到与中心种子或团簇相接触时，就被粘住不动，成为团簇的一部分，如果粒子走出了边界则认为丢失，不进行计数；然后再随机产生一个粒子，重复上述过程，这样就可以得到足够大的 DLA 团簇。

B.2. Hausdorff 维数

Hausdorff 维数可以用以下两种方式进行计算。

B.2.1. 盒计数法

将尺寸 ε 的网格覆盖在分形图形上，求出计数网格中有图形象素（不管有许多像素还是很少像素）的方格数目 $N(\varepsilon)$ ，将一系列 $N(\varepsilon)$ 、 ε 数据作 $\ln N(\varepsilon) \sim \ln(1/\varepsilon)$ 图，如能得到一条直线，它说明 $N(\varepsilon)$ 和 ε 满足 $N(\varepsilon) \sim (1/\varepsilon)^D$ ，即直线的斜率 D 是图形的分维。如果图上只有一部分是直线时，则此图形的自相似性（标度不变性）只存在于直线部分的测度范围内。实际的分形和理想的规则分形不同，它只存在于有限的范围之内。

B.2.2. Sandbox 法

将一系列尺寸 r (>1) 不断增大的方框（也可以是圆）覆盖到分形图形（如 DLA 图形）上，计数不同方框（或圆）中像素数 N （即以像素为测量单元），在 $\ln N \sim \ln r$ 图上如有直线部分，则在此范围内存在 $N \sim r^D$ ，直线部分的斜率即分形维数 D 。有文献报道，如果将方框（或圆）内图形的质心和方框中心（或圆心）重合，得到的 $\ln N$ 和 $\ln r$ 相关性更好，因此在本次计算中将选择和 DLA 模拟的生长中心选为方框（或者圆，如果生长的边界是圆）选为方框中心。上述方框（或圆）的最大尺寸应限于回转半径之内。

回转半径 R_g 的定义是：

$$R_g = \sqrt{\frac{\sum_{i=1}^N r_i^2}{N}}$$

r_i 是粒子（占据的像素）到生长中心的距离。 N 像素总数，求和代表遍历所有像素点。

B.3. 算法

计算中采用正方形边界。粒子产生的边界设为点阵边界。

定义一个边长 $N=2048+2=2050$ （+2 的作用在 dla 部分解释）的二维整型数组 $\text{field}[N][N]=\{\}$ （元素取 0 代表没有粒子占据），其中中间的 2048×2048 个格子是生长用的点阵，将种子放置在中心，即将 $\text{field}[N/2][N/2]$ 置为 1。因为盒计数法要求格子可以用整数个网格进行覆盖，所以选取了 2^{11} 个格子。 $\text{field}[N/2][N/2]$ 不是严格的中心，会偏右上，不过影响不大。输入随机数种子 seed ，用 Scharge 方法连续生成后续随机数，但不进行归一化。输入粒子数 n ，要求 $0 < n < N^2$ ，程序中限制粒子数最大值是 $(2^{11})^2/10 \approx 419430$ ，粒子在运行过程中大多数不会凝聚，根据预实验凝聚率是 10% 左右，所以粒子数可以稍微设多一些。

用整型变量 count 对成功凝聚的粒子进行计数，第一个种子粒子也被计入其中。用整型变量 inf 标识 DLA 图案中距离中心的最小距离，作为新粒子初位置距离中心距离下界。在主函数中利用循环语句调用子函数 dla 进行计算，如果成功凝聚，dla 将返回 1，在屏幕上输出粒子状态信息，更新 DLA 图像的边界即粒子生成起点范围的边界，输出粒子坐标到文件"location.txt"，反之 dla 返回 0，输出粒子状态信息后进入下一循环。循环结束后，计算回转半径 Rg，调用子函数 boxcount 和 sandbox 分别计算分维。

dla 的算法如下：利用 Schrage 方法生成的随机数 l 在正方形边界内距离中心大于 inf 的范围内生成粒子的初始坐标(x, y)，如果这个位置被占据则重新生成直到没有被占据为止。生成的方法是：l 对 (N-1-2*inf) 取余数，如果等于 0 或者 N-2*inf-2 就舍去重新生成新的随机数直到粒子不出现在点阵边界外(即数组的边界上)；对于不在数组 field 边界上的余数，如果小于 center-inf 则 x 等于余数，否则 x 等于余数加 1+2*inf。生成新的 l，类似地得到 y。成功生成粒子后进行随机行走，如果粒子四个方向上存在相邻的被占据点则粒子凝聚，将 field 上对应的值赋值 1，返回 1 到主函数；当粒子上下左右四个方向都不被粒子占据时，生成新的随机数 l，对 4 取模，分别对应四种移动，如果移动后粒子越过边界(即出现 x, y<1 或>N-2) 则返回 0。边长对 2048 加上 2 的作用是保证粒子出现在边界上的时候不会因为边界外的数据是 1 而被误判凝聚。

boxcount 的算法如下：由于 $N=2^{11}$ ，利用循环语句取 $\varepsilon=2^0\sim 2^{-11}$ ，相应地，检查的网格对应大小 $2^{11}\times 2^{11}\sim 1\times 1$ 的二维数组。第 i 个循环中 $\varepsilon = 2^{-i}$ ，检查矩阵 field 的中间部分所用网格的边长包含 $step = \frac{\varepsilon}{2^{-11}} = 2^{11-i}$ 个元素，覆盖 field 的一行需要 $\frac{1}{\varepsilon} = 2^i$ 个网格。

利用循环语句，先行后列地遍历 field，调用子函数 check 检测网格中是否包含占据的像素，计算出含有像素的网格数 $N(\varepsilon)$ ，计算 $\ln N(\varepsilon)$ 和 $\ln(1/\varepsilon) = i * \ln 2$ ，输出到文件"boxcount.txt"。直接计算 $i * \ln 2$ 是为了减小计算机的舍入误差。

check 的算法是逐行逐列地检查第 j,k 个网格中每一个元素，即 j*step+1 到(j+1)*step 行，k*step+1 到(k+1)*step 列。

sandbox 的算法如下：将回转半径 Rg 传入 sandbox，得到不大于 Rg 的最大整数 max，作为最大检测半径。利用循环语句依次检测半径 $r=1\sim \max$ 的方框中占据格点数 $N(r)$ 。为节省算力 $N(r)$ 的计算会继承 $N(r-1)$ 的结果。将 N 预置为 1， $r=1\sim \max$ 时，如图 1，N 等于 N 加上新增的一圈面积中的占据格点数。写成公式如下：

$$N = \sum_{i=center-r}^{center+r} (field[center-r][i] + field[center+r][i]) + \sum_{i=center-r+1}^{center+r-1} (field[i][center-r] + field[i][center+r])$$

输出 $\ln N$ ， $\ln r$ 到文件"sandbox_count.txt"保存。

C. 计算结果及具体分析、讨论

C.1. 绘制 DLA 图形

取 seed=1, n=1E5 进行计算得到 $R_g = 148.627600$ ，count=20147 粒子凝聚率：

$$\eta = \frac{count}{n} \times 100\% = \frac{20147}{10^5} \times 100\% = 20.147\%$$

用 Origin 作 DLA 生长图形如图 1。

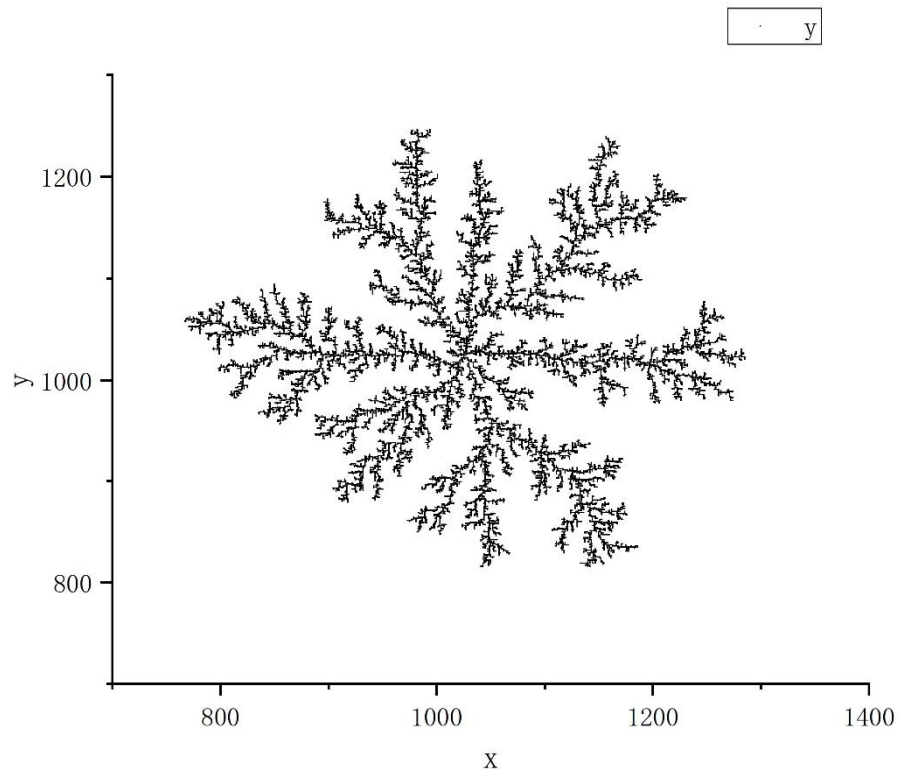


图1 DLA生长图形

由图 1 横纵坐标，注意到 DLA 生长图形主要分布在 $x,y \in (800,1400)$ 的区域，大概占据生长区域的 $\frac{1400-800}{2048} \approx 30\%$ ，满足生长区域边界距中心的距离远大于想要模拟得到的图形尺寸的要求。

C.2. 计算分维

用 Origin 作盒计数法双对数图如图 2，Sandbox 法双对数图如图 3.

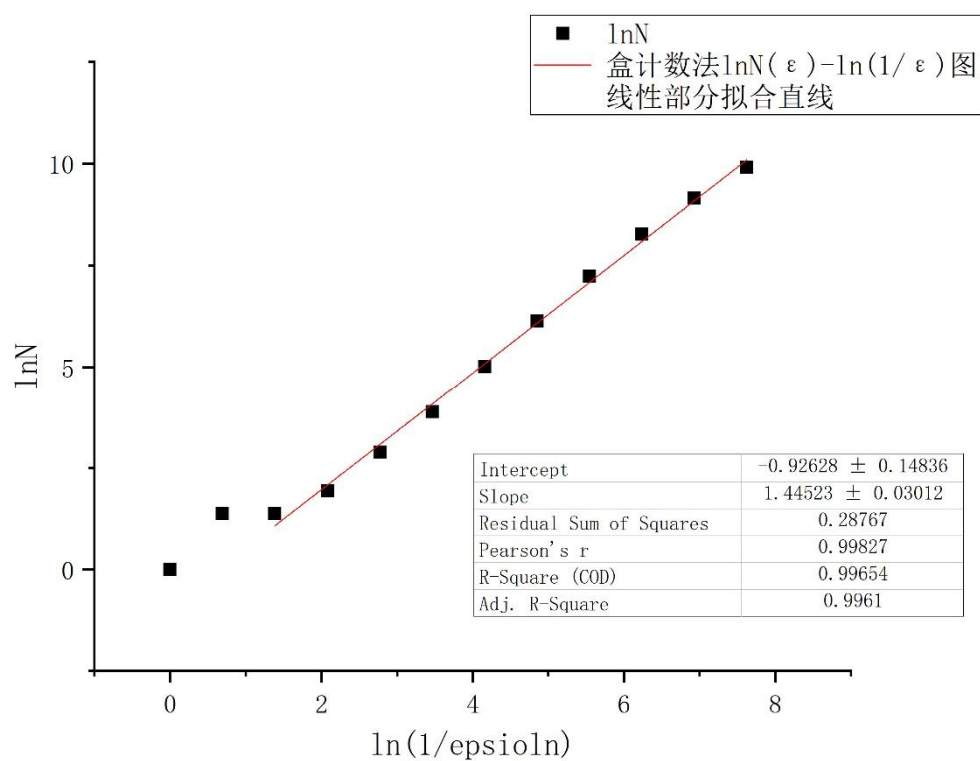


图2 盒计数法 $\ln N - \ln(1/\epsilon)$ 图

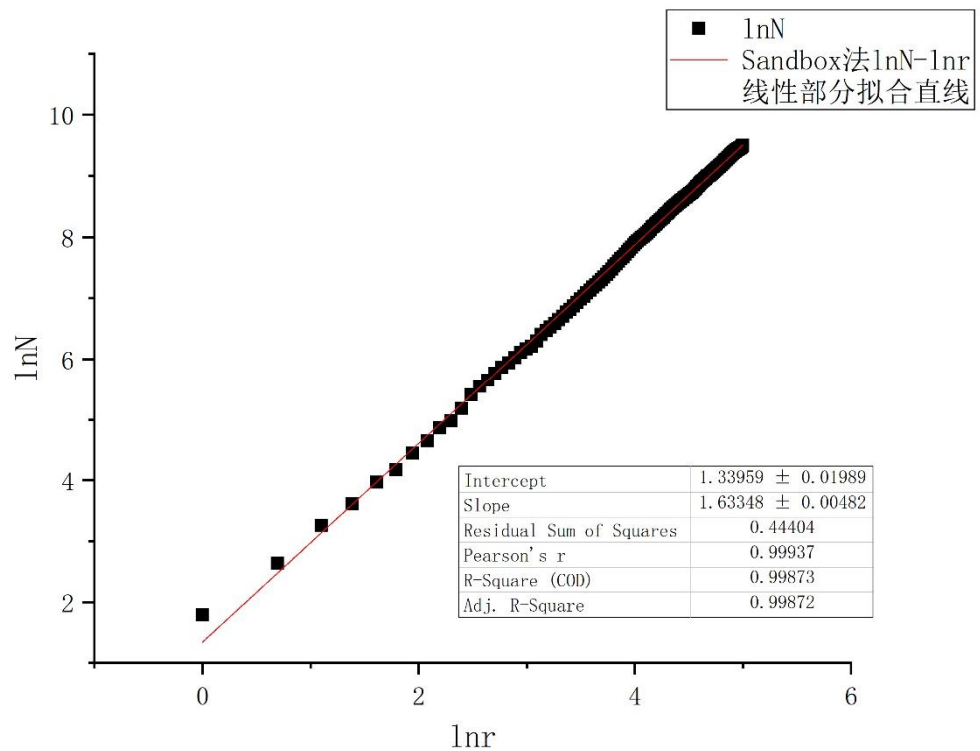


图3 Sandbox法 $\ln N - \ln r$ 图

对盒计数法 $\ln N(\epsilon) \sim \ln(1/\epsilon)$ 图的线性部分进行拟合，得到斜率

$$D_1 \approx 1.445$$

盒计数法得到 DLA 图形的分维是 1.445.

对 Sandbox 法 $\ln N \sim \ln r$ 图的线性部分进行拟合, 得到斜率

$$D_2 \approx 1.633$$

Sandbox 法得到 DLA 图形的分维是 1.633.

两者相差约 0.2.

C.3. 探究粒子凝聚率和粒子数的关系

在 $\text{seed}=1$, $n=1\text{E}4$ 的预实验中粒子的凝聚率约为 10%, 这影响了程序中对粒子数目上限的设置。尽管可以通过改动程序使得程序在 n 个粒子成功凝聚时停止运行以方便地控制图上的粒子数目, 进而防止出现粒子过多不能分辨, 或者粒子过少图像稀疏的情况, 但是, 设置模拟的粒子数可以作为从溶液中结晶的实际情境中, 溶液浓度的一个简化模拟。因此, 粒子凝聚率和粒子数初值的关系可以被视为从溶液中结晶时, 远小于溶液总体积的一定体积 (面积) 中, 形成结晶大小和该区域粒子数, 亦即溶液浓度的关系。

补充了 $n=1\text{E}3$ 和 $n=1\text{E}4$ 的模拟, 得到 DLA 图形如图 4 和图 5. 因为点数太少, 粒子坐标是分立的整数值, 所以图 4 看起来不连续。

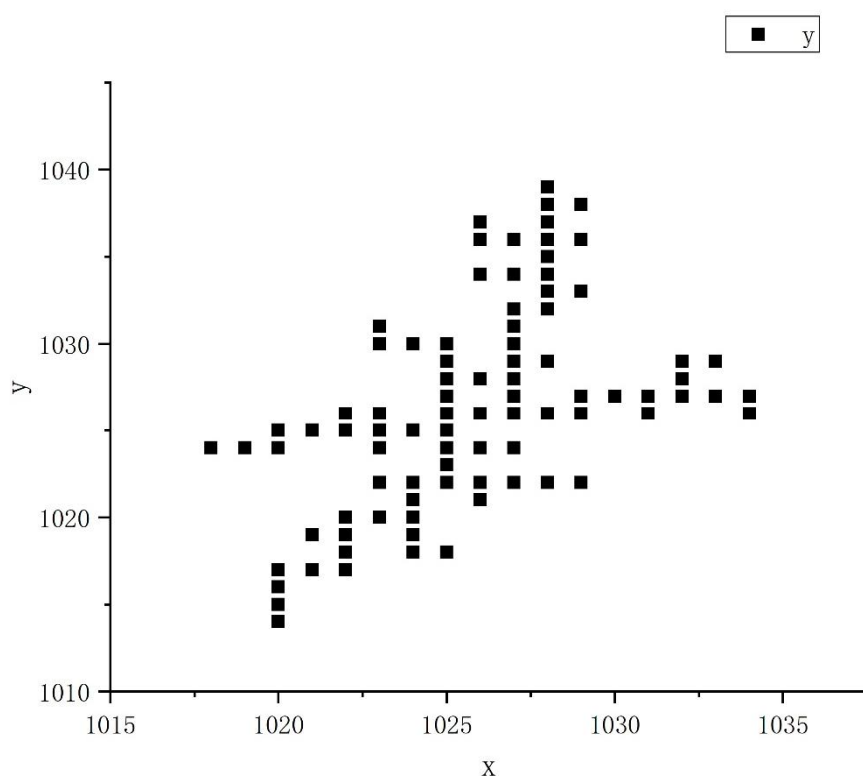


图4 $n=1\text{E}3$ 的DLA生长图

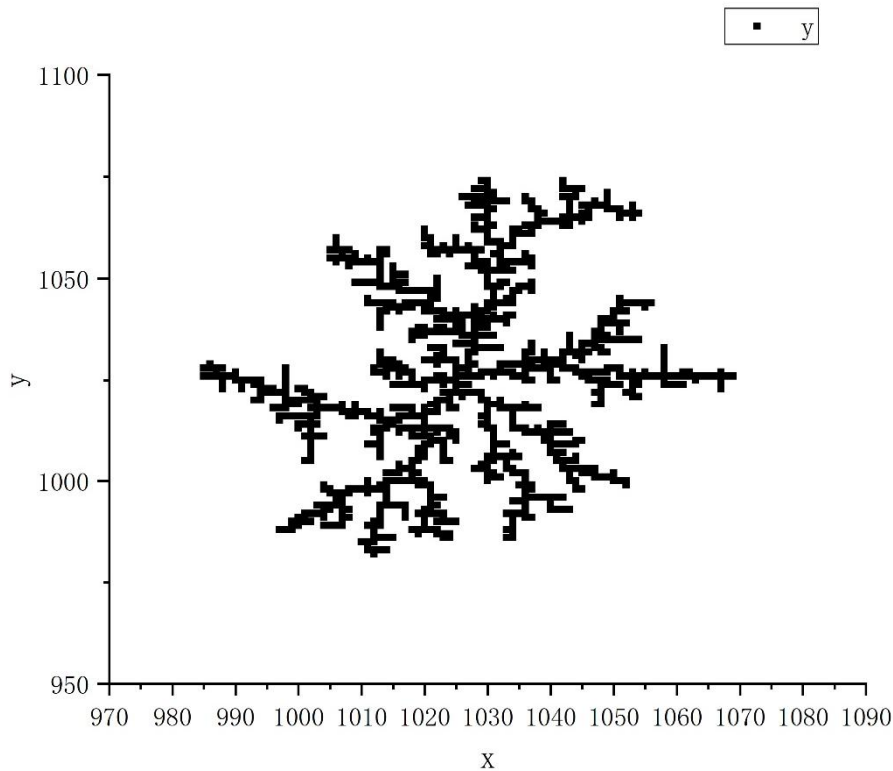


图5 n=1E4的DLA生长图形

两次模拟中分别得到 count=87 和 count=1138，粒子凝聚率分别是

$$\eta[n = 1E3] = \frac{87}{1E3} * 100\% = 8.7\%$$

$$\eta[n = 1E4] = \frac{1138}{1E4} * 100\% = 11.38\%$$

凝聚率随粒子数增多略有提升。因为 n 的大小限制，不适合进行跨数量级的更多试验。改编程序进行模拟，计算 n=100~419430 粒子的 DLA 模拟，求出不同 n 对应的凝聚率。具体的算法是进行 n=419430 的 DLA 模拟，从 n=100 起每隔 10 个点输出一粒凝聚率，这样就可以避免重复进行此前的运算从而节省时间和计算资源。取 seed=1 进行计算，将结果导入 Origin 得到 η -n 折线图如图 6.可以发现粒子凝聚率和模拟粒子数成正相关，最大凝聚率约 35%，增速随粒子数上升而递减。和实验中，浓度越高的稀溶液越容易结晶相符合。

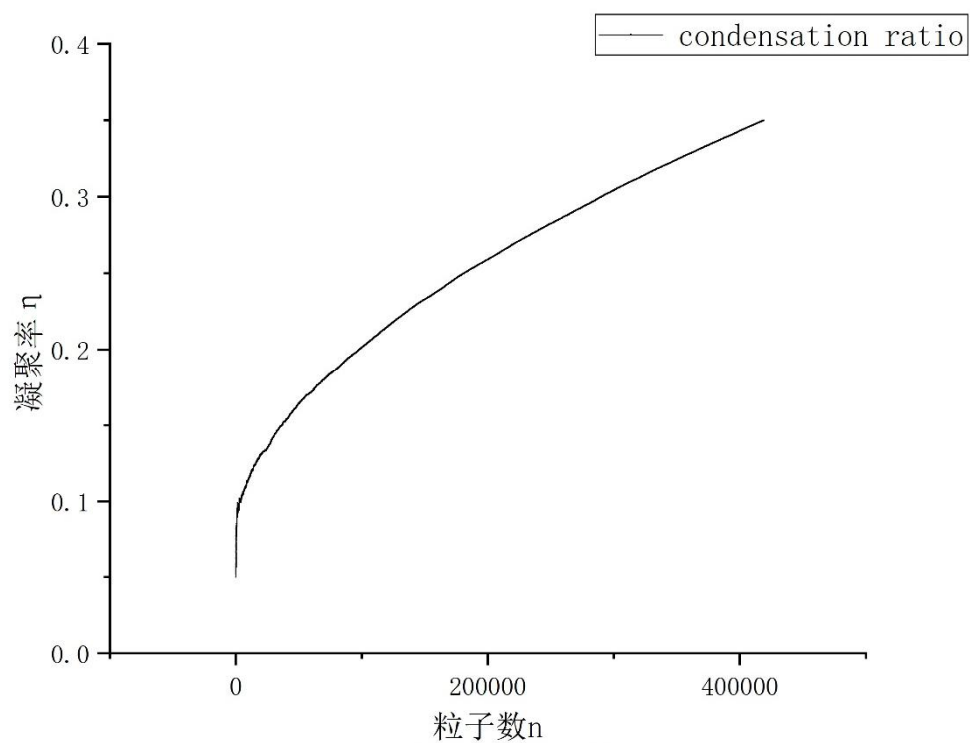


图6 凝聚率-粒子数图

D. 总结

本次作业中学生编写程序进行了二维单中心 DLA 模拟，作出了大小 $2^{11} \times 2^{11}$ 的区域中的 DLA 生长图此后用盒计数法和 Sandbox 方法计算了 DLA 生长图形分维的计算，分别得到 1.445 和 1.633；此后学生改编程序，补充研究了凝聚率和粒子数的关系，得到粒子凝聚率和模拟粒子数成正相关，最大凝聚率约 35%，增速随粒子数上升而递减的结论，和浓度越高的稀溶液越容易结晶这一实验规律相符合。