

计算物理第四次作业第 11 题

姓名：姚星宇 学号：PB21000188 班级：2021 级少年班学院四班

2022 年 11 月 8 日

1 题目重述

模拟 2 维 DLA 以及介电击穿 (DBM) 图案并讨论

2 题目分析

2.1 二维 DLA 模型

对于二维 DLA 模型，根据讲义上讲述的方法可得到编写程序的思路：在最大距离外的圆上按角度随机选择一点释放粒子，随后进行二维网格上的随机行走。若粒子走出另外一圆，则该粒子确认为无效，回到前一步；若该粒子运动到周围已经有粒子的位置，则粘连上去，回到前一步产生新的粒子。

在程序实现上，关键的是 4 个不同类型的数据：记录已有粒子位置的二维数组 `res`，记录已有粒子的边界的二维数组 `boundary`（边界点上没有粒子，但是每个边界点都至少与一个粒子相邻），记录最大距离的浮点数 r_0 ，记录正在随机行走的粒子的位置的数组 `r`。

每添加一个新的粒子时，生成一个随机数 $x \in [0, 1]$ ，随后将初始坐标 $[(1.5r_0 + 5) \cos(2\pi x), (1.5r_0 + 5) \sin(2\pi x)]$ 赋值给 `r`，作为在最大距离外一段距离随机抽取的初始坐标。然后进行随机行走，每走一步，先检测是否越过模拟边界（ $r[0]^2 + r[1]^2$ 与 $(2r_0 + 10)^2$ 的大小比较），若越过，则重新抽取坐标并重新行走；若没有越过，则遍历边界二维数组 `boundary` 进行比较：若边界数组中不含有 `r`，则继续行走；若边界数组中含有 `r`，即粒子已经运动到边界上，则视为成功添加一个新的粒子，并展开后续流程。

当成功添加一个新粒子后，先将新粒子的位置值加入粒子二维数组 `res`，以表示该粒子已经粘连（需要注意的是，python 中的 list 的 `append` 方法为传入地址而非传入值，故此处的加入不能是直接将变量加入列表 `res.append(r)` 而是将变量值加入 `res.append([r[0], r[1]])`；然后在边界二维数组中去除该粒子，表示该边界点已被占据；最后，检测新粒子周围的四个点，若这点：1. 没有被粒子占据（不在 `res` 中）2. 不为边界点（不在 `boundary`）中 3. 该点周围至少有一个点已经被粒子占据，则将这一点标记为边界点（即加入 `boundary`，方法同上）。在同时，也需检测新粒子距离原点的距离，并与 r_0 进行比较（ $r[0]^2 + r[1]^2$ 与 r_0 比较），并将 r_0 更新为较大值。完成上述处理后，开始添加下一个粒子。

模拟开始时，需进行如下初始化： $r_0 = 1$ ，`res = [[0, 0]]`，`boundary = [[0, 1], [0, -1], [1, 0], [-1, 0]]`（当然，自定义初始化较为容易实现，但这里没有做，而是固定一种情况进行初始化）。随后即可开始模拟。

2.2 介电击穿 (DBM) 图案

介电击穿模型，其模拟流程大致为：对于可能生长的点解出该点电势，然后对于电势进行处理，并以处理后的数据为权重进行抽样，最后得到新生长出的粒子。

流程相较 DLA 简单些，但解出电势的方法多样。这里笔者认为，由于边界点的随意性与狄里克雷边界条件的特殊性，以及只要求得到少部分点电势的题目特殊性，这里应该使用随机行走解出拉普拉斯方程。

在程序实现上，关键的是个 6 不同类型的数据：记录已有粒子位置的二维数组 `res`，记录内边界点位置的二维数组 `inboundary`，记录外边界点位置的二维数组 `outboundary`（内外边界点的概念会在后文说明），记录外边界点各位置抽样权重的数组 `array`，记录粒子最大距离的浮点数 r_0 ，记录正在随机行走的粒子的位置的数组 `r`。

内边界点为满足以下条件的点：该点已经被粒子占据，且相邻的点中至少有一个点未被粒子占据；外边界点为满足以下条件的点：该点未被粒子占据，且相邻的点中至少有一个点被粒子占据。这两类边界的意义为：新粒子只会在外边界上生长；解未被粒子占据的点的电势所使用的随机行走解拉普拉斯方程的方法中，随机行走的路径只会在内边界上结束。

在每一轮模拟中，首先，解出 `outboundary` 中各点的电势：遍历 `outboundary`，对每一点进行随机行走。对 `outboundary` 中第 i 个点进行的随机行走中，对其每一步，进行如下检测：若粒子逃离一定范围 ($r[0]^2 + r[1]^2 > (2r_0 + 10)^2$)，则认为该粒子逃逸，`array[i]` 不变，并在 `outboundary[i]` 重新释放一个粒子；反之，则检测 `r` 是否在 `inboundary` 中：若是，则 `array[i] += 1`，并结束这次行走；反之则继续行走。结束 `outboundary` 的遍历后，可获得各点的电势。按题给要求对 `array` 进行处理 ($|1 - array/N|^\gamma$ ， N 为行走次数)。以 `array` 为权重进行抽样，可得到新的粒子所在的位置。

得到新增加的粒子所在的位置后，首先将该粒子加入 `res` 并从 `outboundary` 中移除，表示这点已经被粒子占据，然后对该点周围的点进行检测：若该点没有被粒子占据且不再外边界中，则将其加入外边界；若该点在内边界中，且周围各点均被粒子占据，则将该点从内边界移除。最后同 DLA 中的方法更新 r_0 。完成上述处理后，开始下一轮模拟。

模拟开始时，需进行如下初始化： $r_0 = 1$ ，`res` = `[[0, 0]]`，`outboundary` = `[[0, 1], [0, -1], [1, 0], [-1, 0]]`，`inboundary` = `[[0, 0]]`（当然，自定义初始化较为容易实现，但这里没有做，而是固定一种情况进行初始化）。随后即可开始模拟。

3 结果

DLA 模拟 1000 个点与 DBM 模拟 300 个点的图像结果如图 1。

参考文献

[1] 丁泽军. 计算物理讲义 [M]

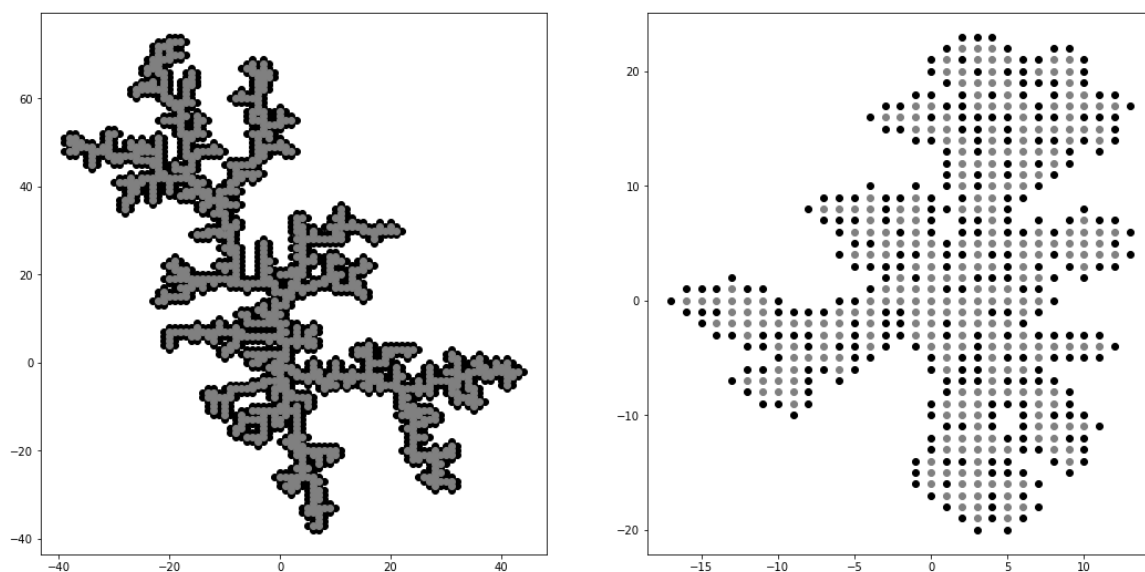


图 1: 程序运行结果