

Отчет по групповому проекту. Этап 3

Неравновесная агрегация, фракталы

Шалыгин Г. Э. Низамова А. А. Голощапова И. Б.
Серегин Д. А. Пиняева А. А.

11 марта 2023

Содержание

1	Цель работы	5
2	Теоретическое введение	6
2.1	Постановка задачи	6
2.2	Случайные блуждания	6
2.2.1	Одномерные случайные блуждания	6
2.2.2	Многомерные случайные блуждания	7
2.3	Фрактальная размерность	7
3	Математические модели	9
3.1	Агрегация, ограниченная диффузией	9
3.1.1	Сеточная модель.	9
3.1.2	Бессеточная модель.	9
3.1.3	Химически-ограниченная агрегация	10
3.1.4	Баллистическая агрегация	10
4	Програмная реализация	11
4.1	Используемые возможности языка Julia	11
4.2	Листинги используемых функций для модели DLA	11
4.2.1	Генерация псевдослучайных чисел	11
4.2.2	Генерация координат следующей частицы	12
4.2.3	Дополнительные функции	12
4.2.4	Блуждание частицы	12
4.2.5	Псевдокод модели DLA	13
4.3	Другие модели	14
4.3.1	Бессеточная	14
4.3.2	Химически-ограниченная	14
5	Промежуточные результаты	15
5.1	Сеточная модель	15
5.2	Бессеточная модель	16
5.3	Итоги	16
	Список литературы	17

Список иллюстраций

5.1	Результат сеточной модели	15
5.2	Результат сеточной модели	16

Список таблиц

1 Цель работы

Цель:

- Реализовать алгоритмы моделирования неравновесной агрегации.

Задачи:

- Рассмотреть возможности языков для программной реализации алгоритмов
- Реализовать алгоритмы, описанные втором этапе

2 Теоретическое введение

2.1 Постановка задачи

Существуют разнообразные физические процессы, основная черта которых — неравновесная агрегация. Примеры: образование частиц сажи, рост осадков при электрическом осаждении и т. п. Во всех случаях происходит необратимое прилипание частиц к растущему кластеру из-за сильного смещения равновесия в сторону твердой фазы, и вырастают разветвленные агрегаты (рост правильных ограниченных кристаллов происходит в условиях, близких к равновесным, когда возможно как прилипание частиц, так и их обратный переход в раствор)

2.2 Случайные блуждания

2.2.1 Одномерные случайные блуждания

Рассмотрим простую модель — пусть частицы могут двигаться только вдоль прямой, делая шаги случайной длины. Примем также, что величина и направление каждого шага определяются независимо от положения частицы и от предыдущих шагов (модель «пьяного моряка»). Будем наблюдать за частицей через равные промежутки времени. Координата частицы вычисляется по рекуррентной формуле $x_k = x_{k-1} + \delta_k$, где δ_k — очередной шаг блуждания. При отсутствии силового поля смещение влево и вправо равновероятны. В общем случае вероятность того, что длина шага лежит в промежутке от δ до $\delta + d\delta$, равна $dp = w(\delta)d\delta$.

Функция w называется плотностью вероятности для величины шага Δ , который называют случайной величиной.

Подробнее в [1].

2.2.2 Многомерные случайные блуждания

Предоставим теперь нашим частицам возможность двигаться также по координате y — рассмотрим двумерные случайные блуждания (случай трех измерений получается аналогично). Можно независимо задавать смещение по вертикали δ_y равномерно распределенным таким же образом, как и смещение по горизонтали δ_x .

Функция распределения в двумерном случае представляется в виде произведения двух функций распределения по координатам x и y , так как x и y являются независимыми случайными величинами. Подробнее в [1].

2.3 Фрактальная размерность

Фигура на плоскости или тело в пространстве имеют размерность. Определить ее можно разными способами. В случае, когда у фигуры есть выделенная центральная точка, можно построить много сфер различного радиуса с центром в этой точке. Для каждой сферы можно вычислить массу части фигуры, которая оказалась внутри этой сферы. В случае, когда масса пропорциональна радиусу сферы в некоторой степени ($m \sim R^D$), показатель степени D называется размерностью тела. Это так называемый метод сфер или ящиков. Для линий $D = 1$, у плоских фигур $D = 2$, у «обычных» тел $D = 3$. Однако, многие объекты в природе имеют размерность, выражающуюся дробным числом.

Такие тела, следуя Б. Мандельброту [9], называют фракталами (от латинского слова fractus — дробный). Фракталами являются также дендриты, вырастающие при электроосаждении металлов, кластеры, полученные при агрегации коллоидов; фрактальную структуру имеют ветви деревьев, кровеносная система и т.

п.

Еще один способ может применяться при наблюдении за процессом роста агрегата от центра. В этом случае число частиц в кластере $N \sim R_{ch}^D$. В качестве характерного радиуса R_{ch} можно выбирать, например, максимальный радиус кластера R_{max} ,

Желающим более подробно познакомиться с фракталами рекомендуем книгу [2].

3 Математические модели

3.1 Агрегация, ограниченная диффузией

3.1.1 Сеточная модель.

Возьмем регулярную сетку на плоскости, например, квадратную. В центр поместим затравочную частицу. Затем с достаточно большого расстояния будем выпускать по одной новые частицы. Выпущенная частица совершает случайные блуждания по сетке, делая шаги в одном из четырех доступных направлений с равной вероятностью. Если частица оказывается по соседству с затравкой, она прилипает и остается в этом узле. Затем выпускаем следующую частицу, которая может прилипнуть к одному из занятых узлов. Шаг решетки в этой модели соответствует длине связи между частицами (расстоянию устойчивого равновесия для взаимодействия двух частиц).

Некоторые указания. Для ускорения работы программы разумно выпускать частицы с круга радиусом немного больше R_{\max} — текущего максимального радиуса агрегата.

Если частица уходит далеко, уничтожаем ее и выпускаем новую.

3.1.2 Бессеточная модель.

Структура полученных DLA-кластеров отражает структуру сетки (имеются выделенные направления). Чтобы получить более симметричные кластеры, можно отказаться от сетки. В этом случае рост происходит следующим образом: вначале

помещаем в центр поля затравочную частицу, затем с круга некоторого радиуса выпускаем следующую, которая случайно блуждает. Если частицы сближаются на расстояние взаимодействия (например, их удвоенный радиус), они слипаются. После этого выпускаем новую частицу и т. д.

Детальная информация в [3]

3.1.3 Химически-ограниченная агрегация

При диффузионно-ограниченной агрегации частица всегда прилипает к кластеру с вероятностью 1. Можно уменьшить вероятность прилипания. Такой процесс роста называется химически-ограниченной агрегацией. Он моделирует ситуацию, когда вероятность зависит от того, каким концом молекула повернута к другой. Это приведет к появлению более плотных агрегатов (увеличению размерности), потому что у частицы увеличится шанс проникать во внутренние области и заполнять пустоты. Размерность, однако, остается меньше размерности пространства, т. е. кластер остается фракталом.

3.1.4 Баллистическая агрегация

До сих пор мы рассматривали рост кластеров с точечной затравки. Однако, довольно часто встречаются ситуации, когда агрегаты растут на поверхности, например, при выпадении осадка на дне или стенках сосуда. Если новые частицы доставляются к растущему кластеру за счет диффузии, имеем просто модель DLA с измененными начальными условиями.

Другой случай — *баллистическая агрегация*, при которой частицы свободно падают по прямолинейным траекториям. Частица прилипает, когда оказывается рядом с занятым узлом. В этом процессе получается более плотный агрегат (но не сплошной), однако его граница сильно изрезана и является фракталом.

4 Програмная реализация

4.1 Используемые возможности языка Julia

Округление: - `round(x)` округление к ближайшему целому числу `typeof(x)` - `round(T, x)` округление к ближайшему целому числу `T` - `floor(x)` округление `x` к `-Inf` `typeof(x)` - `floor(T, x)` округление `x` к `-Inf` `T` - `ceil(x)` округление `x` к `+Inf` `typeof(x)` - `ceil(T, x)` округление `x` к `+Inf` `T` - `trunc(x)` округление `x` по направлению к нулю `typeof(x)` - `trunc(T, x)` округление `x` по направлению к нулю `T`

Математические операции: - `abs(x)` модуль - `sqrt(x)`, \sqrt{x} квадратный корень `x` - `sin(x)`, `cos(x)` - `maximum(X)`, `minimum(X)` - максимум и минимум массивов

Макрос `@time` позволяет измерить время работы функции. Подробное описание в [4]

4.2 Листинги используемых функций для модели DLA

4.2.1 Генерация псевдослучайных чисел

```
next = 0
function rand()
    global next = (next * 1664525 + 1013904223) % 2^32
    return next / 2 ^ 32
end
```

4.2.2 Генерация координат следующей частицы

```
function GetNextParticular(x_center, y_center, r)
    r = r
    angle = 2 * pi * rand()
    x = r * cos(angle) + x_center
    y = r * sin(angle) + y_center
    return round(x), round(y)
end
```

4.2.3 Дополнительные функции

Расстояние между точками $(x_1, y_1), (x_2, y_2)$

```
function dist(x1, y1, x2, y2)
    return sqrt((x2-x1)*(x2-x1) + (y2-y1)*(y2-y1))
end
```

Проверка того, что частица столкнулась с кластером

```
function check(x, y)
    for i in 1:n
        if abs(X[i] - x) + abs(Y[i] - y) == 1
            return true
        end
    end
    return false
end
```

4.2.4 Блуждание частицы

```
function RandomWalk(x, y, i, r, xl, xr, yu, yd)
    step = 1; dx = [1, -1, 0, 0]; dy = [0, 0, 1, -1]
```

```

while step < 500 && dist(x, y, (xl + xr) / 2, (yu+ yd) / 2) < 4 * r
    if check(x, y)
        X[i] = x; Y[i] = y
        return true
    end
    j = floor(Int, 100 * rand()) % 4 + 1
    x = x + dx[j]
    y = y + dy[j]
    step += 1
end
return false
end

```

4.2.5 Псевдокод модели DLA

```

function DLA(t)
    i = 1
    while i < t
        xl = minimum(X)
        xr = maximum(X)
        yu = minimum(Y)
        yd = maximum(Y)
        r = dist(xl, yd, xr, yu) / 2 + 3
        x, y = GetNextParticular((xr+xl)/2, (yu+yd)/2, r)
        ok = RandomWalk(x, y, i, r, xl, xr, yu, yd)
        if ok
            i += 1
        end
    end
end
end

```

4.3 Другие модели

- **Бессеточная:** добавляется выбор случайной длины шага.
- **Химически-ограниченная:** вводится условие прилипания.

4.3.1 Бессеточная

```
function RandomWalk(x, y, i, r, xl, xr, yu, yd)
    <...>
        x = x + 2 * rand() - 1
        y = y + 2 * rand() - 1
    <...>
end
```

4.3.2 Химически-ограниченная

```
function RandomWalk(x, y, i, r, xl, xr, yu, yd)
    <...>
        if check(x, y) and random() > 0.2
            X[i] = x
            Y[i] = y
        <...>
    end
```

5 Промежуточные результаты

5.1 Сеточная модель

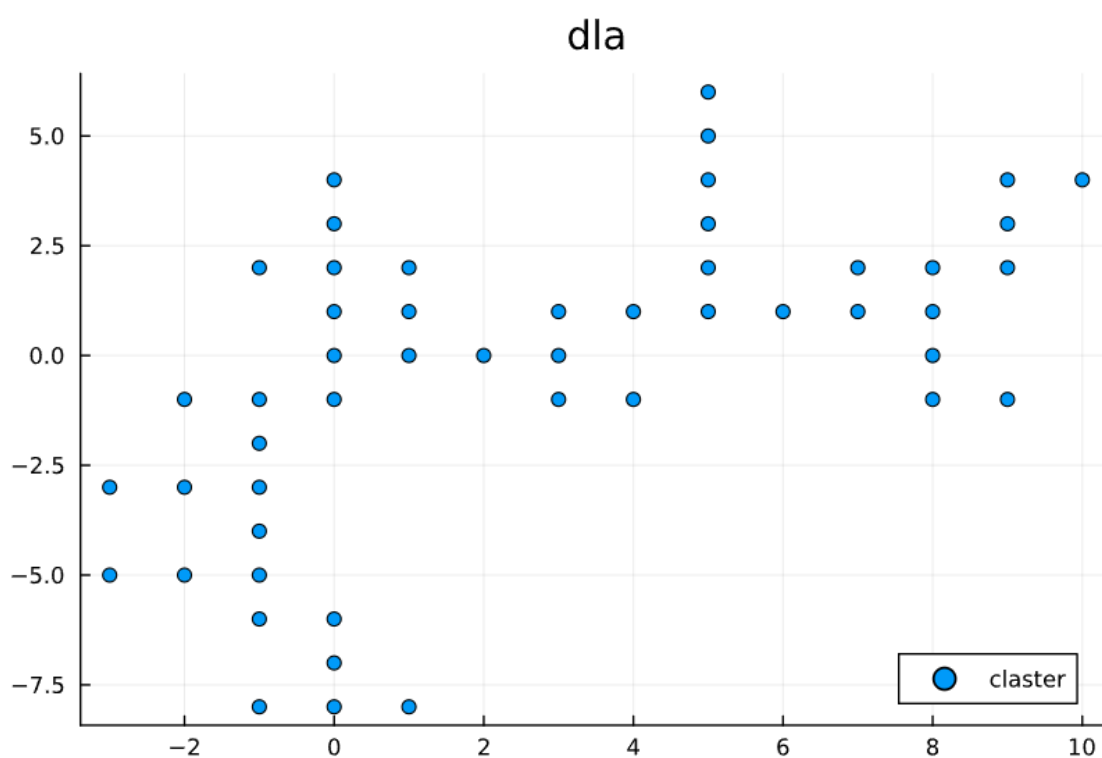


Рис. 5.1: Результат сеточной модели

5.2 Бессеточная модель

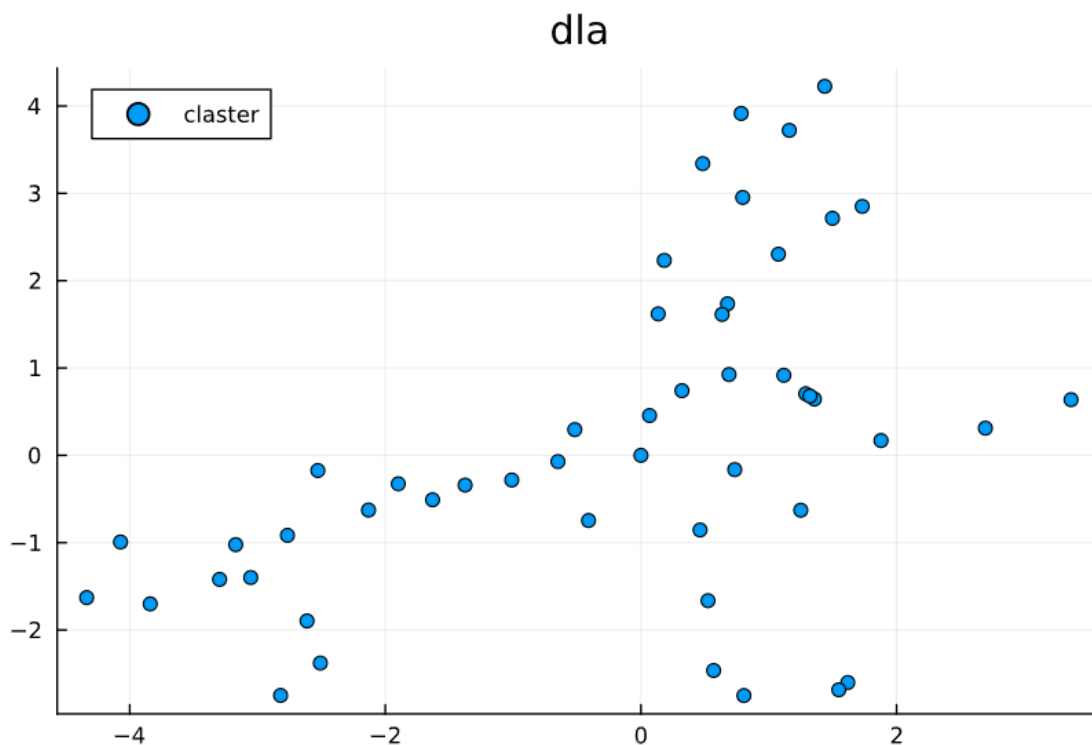


Рис. 5.2: Результат сеточной модели

5.3 Итоги

Программные реализации для моделей неравновесной агрегации написанные на Julia позволяют проводить вычислительные эксперименты для изучения явления. # Выводы

- Реализованные алгоритмы позволяют провести вычислительные эксперименты.
- Возможно изучение разных моделей при минимальных изменениях кода.

Список литературы

1. Н.Ширяев А. СЛУЧАЙНЫЕ БЛУЖДЕНИЯ и БРОУНОВСКОЕ ДВИЖЕНИЕ. Математический институт им. В. А. Стеклова РАН, МГУ им. М. В. Ломоносова, 2003. 103 с.
2. К. Б.В. Основы фрактальной геометрии и фрактального исчисления. Улан-Удэ: ИЗДАТЕЛЬСТВО БГУ, 2013. 224 с.
3. Д. А. Медведев Э.Р.П. А. Л. Куперштох. Моделирование физических процессов и явлений на ПК: Учеб. пособие. СПб.: Новосибирск: Новосиб. гос. ун-т., 2010. 101 с.
4. Белов Г.В. Краткое описание языка программирования Julia. Мир, 2020. 115 с.