

Неравновесная агрегация, фракталы

Этап 2

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

04 марта 2023

Российский университет дружбы народов, Москва, Россия

Информация

- Серегин Денис Алексеевич
- НФИ-02-20
- Российский университет дружбы народов

Вводная часть

- У неравновесной агрегации есть модели.
- Для изучения строятся алгоритмы.
- Модели сложные, ресурсы ограничены, значит необходимы эффективные алгоритмы.

- Математические модели неравновесной агрегации.
- Построение алгоритмов решения задачи моделирования процесса неравновесной агрегации.

Цель:

- Изучить алгоритмы моделирования неравновесной агрегации.

Задачи:

- Рассмотреть методы используемые в построении алгоритмов неравновесной агрегации.
- Построить алгоритмы модели неравновесной агрегации.
- Построить алгоритм вычисления фрактальной размерности.

Используемые методы

- Алгоритм, порождающий последовательность чисел, элементы которой почти независимы друг от друга и подчиняются заданному распределению (обычно дискретному равномерному).
- Линейный конгруэнтный метод:
 1. Начальные условия: m – достаточно большое натуральное число, a – множитель, c – приращение, X_0 – начальное значение.
 2. Возвращаемые значения: последовательность X_1, X_2, \dots
 3. Последовательность описывается формулой:

$$X_{n+1} = (a \cdot X_n + c) \mod m$$

Алгоритмы для DLA

Глобальные переменные: a , m , c , x_{cur} .

Algorithm 1: Get random

Input :

Output: x - равномерно распределенное случайное число из $(0, 1)$

- 1 $x = (x_{cur} * a + c) \bmod m$
 - 2 $x_{cur} = x;$
 - 3 return $x / m;$
-

Algorithm 2: Get next particular

Input : x' , y' , r - координаты центра и радиус текущего агрегата

Output: x , y - координаты новой частицы

```
1  $r = r + \max(10, r);$   
2  $\text{angle} = 2 * \text{PI} * \text{random}();$   
3  $x = r * \cos(\text{angle}) + x';$   
4  $y = r * \sin(\text{angle}) + y';$   
5 return  $x, y;$ 
```

Algorithm 3: Random walk

Input : x, y , ref X , ref Y , x', y' , r - координаты текущей частицы, координаты кластера, координаты центра, радиус

Output: Добавление координат в X, Y

```
1 step = 1; dx = [1, -1, 0, 0]; dy = [0, 0, 1, -1];
2 while step < 500 and (xl - x > r or x - xr > r or y - yu > r or yd - y > r) do
3     if частица оказалась по соседству с кластером then
4         X.append(x); Y.append(y);
5         return;
6     end
7     i = ceil(100*random()) mod 4;
8     x += dx[i]; y += dy[i]; step++;
9 end
```

Algorithm 4: DLA

Input : x_0, y_0, t - координаты первой частицы, количество итераций

Output: X, Y - координаты частиц через время t

```
1  $x_l = x_0; x_r = x_0; y_u = y_0; y_d = y_0;$   
2 for  $i=1..t$  do  
3    $r = \sqrt{\text{pow}((x_r - x_l), 2) + \text{pow}((y_u - y_d), 2))/2};$   
4    $x, y = \text{Get next particular}((x_r - x_l)/2, (y_u - y_d)/2, r);$   
5    $\text{Random walk}(x, y, X, Y, x_0, y_0, r);$   
6 end  
7 return  $X, Y;$ 
```

Algorithm 5: Fractal uniformity

Input : $X, Y, cnt, maxR$ - координаты частиц агрегата, количество сфер, максимальный радиус

Output: d - фрактальная размерность

```
1 d = -1
2 for i=1..cnt do
3     r = random()*maxR; m = количество частиц кластера внутри i-той сферы;
4     newD = ln(m) / ln(r);
5     if d == -1 or abs(d - newD) < 1e-10 then
6         | d = newD;
7     else
8         | return -1;
9     end
10    return d;
```

Другие модели

- Бессеточная: добавляется выбор случайной длины шага.
- Химически-ограниченная: вводится условие прилипания.
- Баллистическая: метод случайного блуждания изменяется на прямолинейную траекторию.

Итоги

- Рассмотренные алгоритмы позволяют построить комплексы программ для вычислительных экспериментов.