

Этап 2

Алгоритм решения задачи

Беличева Д. М., Демидова Е. А.,
Смирнов-Мальцев Е. Д., Сунгурова М. М.

Содержание

1	Введение	4
2	Задание	5
3	Выполнение работы	6
3.1	Описание алгоритма	6
3.2	Описание алгоритма	8
3.3	Описание алгоритма выпуска частицы	10
3.4	Случайное блуждание	12
3.5	Описание алгоритма движения частицы	12
4	Выводы	14
	Список литературы	15

Список иллюстраций

3.1	Блок-схема алгоритма модели DLA	7
3.2	Блок-схема алгоритма реализации модели DLA	9
3.3	Блок-схема алгоритма выпуска частицы	11
3.4	Блок-схема алгоритма движения частицы	13

1 Введение

Цель работы

Рассмотреть алгоритм моделирования агрегации, ограниченной диффузией(DLA) на сетке.

Задачи

- Описать алгоритм моделирования DLA
- Описать алгоритм реализации DLA на сетке

2 Задание

Здесь приводится описание задания в соответствии с рекомендациями методического пособия и выданным вариантом.

3 Выполнение работы

3.1 Описание алгоритма

Рассмотрим сеточную модель агрегации, ограниченной диффузией (Diffusion Limited Aggregation, DLA [1]).

Возьмем регулярную квадратную сетку на плоскости. В центр поместим затравочную частицу. Затем с расстояния чуть больше желаемого максимального радиуса итогового агрегата 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 диаметру частицы(мы рассматриваем единичный шаг)(рис. 3.1).

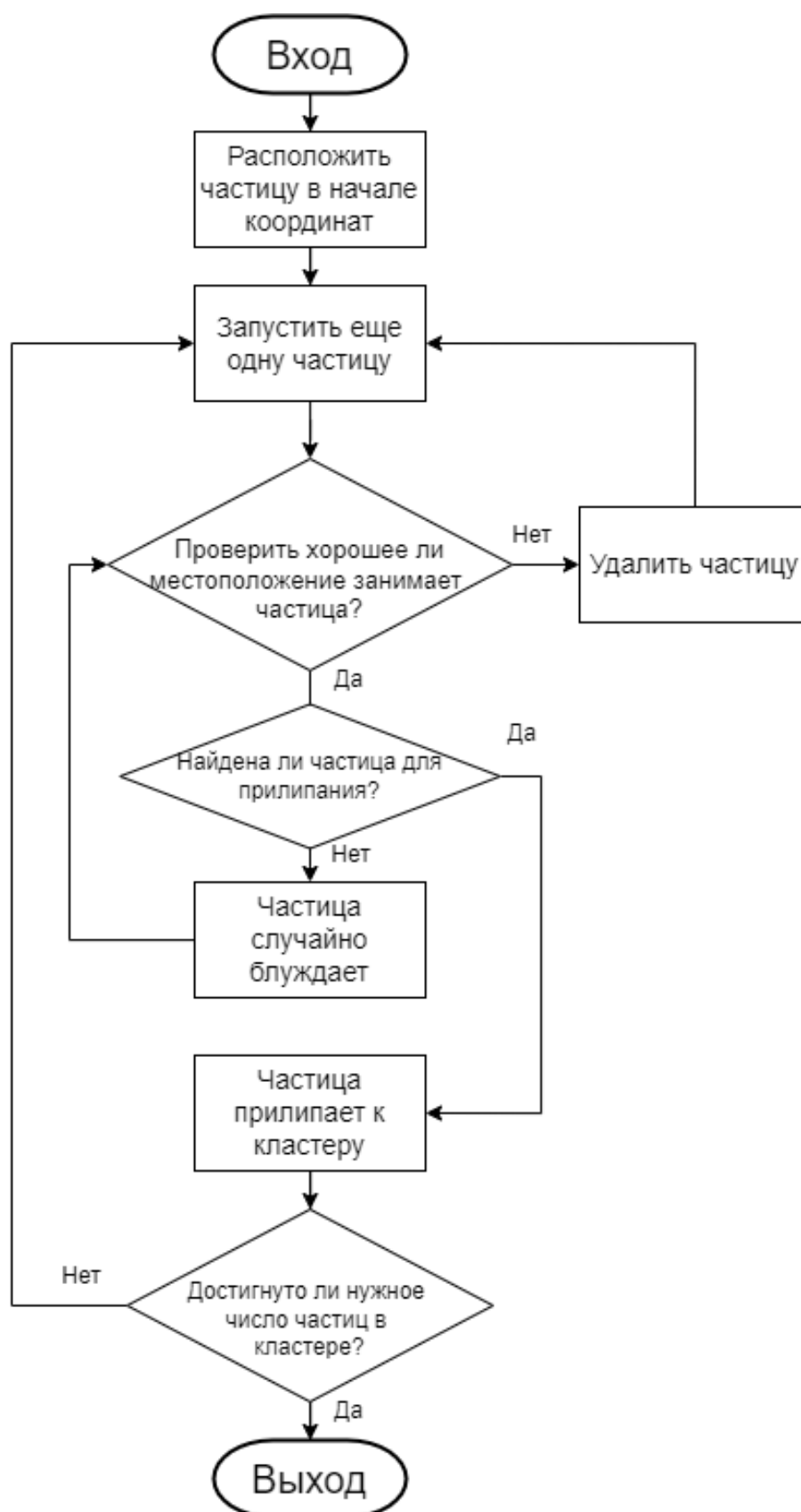


Рис. 3.1: Блок-схема алгоритма модели DLA

3.2 Описание алгоритма

При реализации будем хранить информацию о частицах в массиве, если частица есть – соответствующий элемент матрицы равен один, если нет – ноль. Помещаем частицу в центр поля. Новые частицы будем считать, чтобы можно было остановить программу при их желаемом количестве. Для запуска частицы используем функцию `randomAtRadius`, а для проверки положения частицы и её случайного блуждания `checkAround`. Новая частица запускается, когда текущая оказывается рядом с уже занятой клеткой. Программа прекращает работу либо если уже было запущено достаточное количество частиц, либо если агрегат достиг максимального радиуса(рис. 3.2).

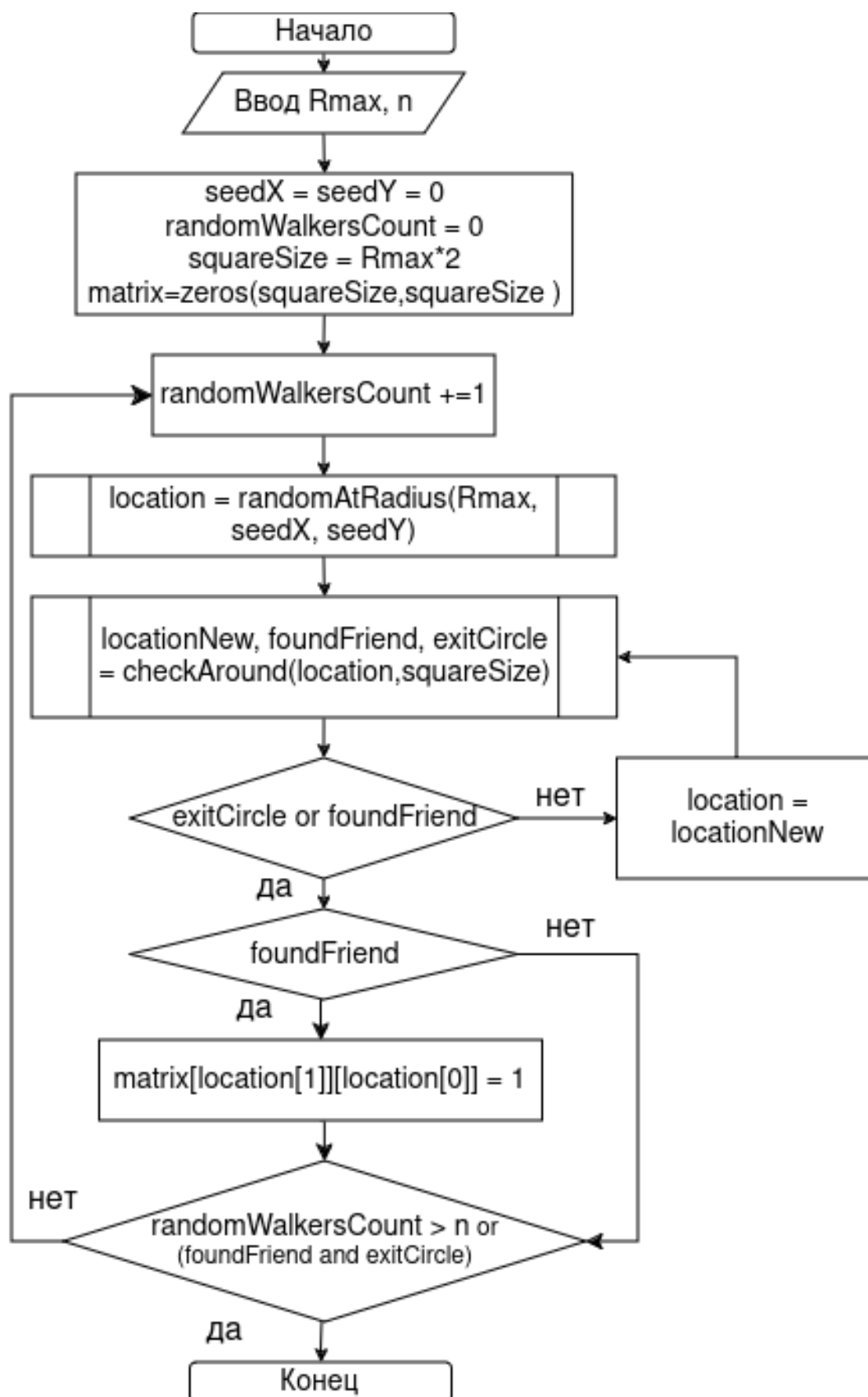


Рис. 3.2: Блок-схема алгоритма реализации модели DLA

3.3 Описание алгоритма выпуска частицы

Для ускорения работы программы разумно выпускать частицы с круга радиусом немного больше R_{\max} текущего максимального радиуса агрегата. Функция генерирует случайное расположение точки на заданном радиусе по формуле[2]:

$$x = r * \cos(\theta) y = r * \sin(\theta),$$

где θ – случайный угол от 0 до 2π , заданный формулой: $2\pi \text{random}$ (рис. 3.3).

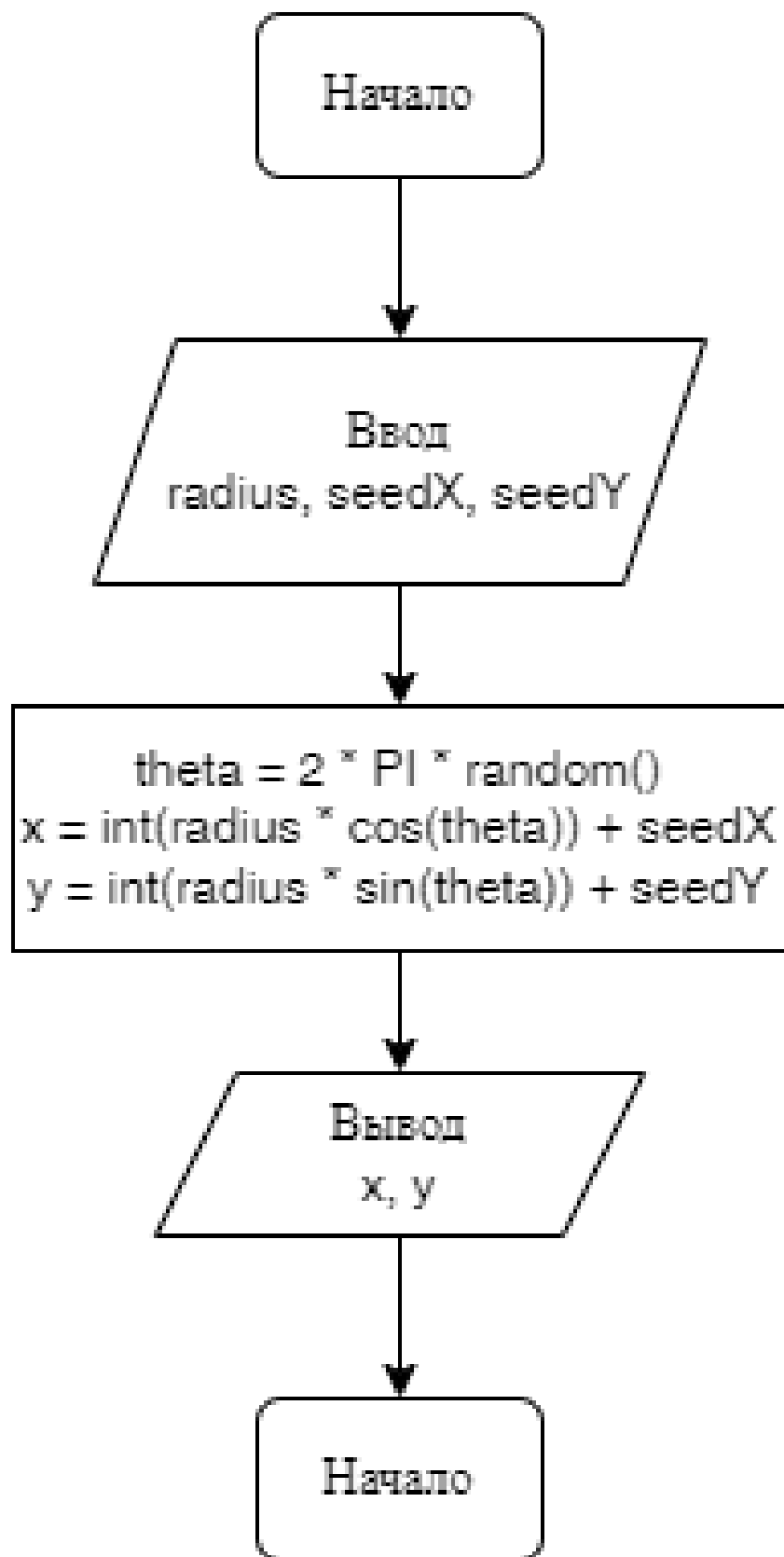


Рис. 3.3: Блок-схема алгоритма выпуска частицы

3.4 Случайное блуждание

Рассмотрим целочисленную решётку Z^2 на плоскости с отмеченной точкой $(0, 0) \in Z^2$ – началом координат. Каждой точке решётки соответствуют четыре точки, в которые можно из неё шагнуть по выходящим из неё ребрам: мы будем обозначать эти точки $v^u = (0, 1)$, $v^d = (0, -1)$, $v^r = (1, 0)$, $v^l = (-1, 0)$ для шагов направо, налево, вверх и вниз соответственно. Случайное блуждание – это недетерминированное передвижение по решетке Z^2 : стартуя из нуля, мы делаем один шаг в секунду, переходя в одну из соседних вершин к той вершине, в которой мы находились в предыдущий момент. При этом решение, в какую вершину шагнуть, принимается случайным образом.

Обозначим $v^u = (0, 1)$, $v^d = (0, -1)$, $v^r = (1, 0)$, $v^l = (-1, 0)$ - шаг на 1 вверх, вниз, влево, вправо соответственно.

$\{S_n\}$ - ряд, описывающий случайное блуждание, $* = u, d, r, l, n$ - количество шагов

$$S_n = \sum_{i=1}^n v_n^*,$$

$$P(v_{i+1} = v_n^*) = \frac{1}{4}$$

3.5 Описание алгоритма движения частицы

Эта функция проверяет местоположение частицы, чтобы убедиться, что она не находится на краю квадрата. Если не на краю, то проверяется периметр (вверх, вниз, влево, вправо). Булевым переменным foundFriend, exitCircle, nearEdge присваивается значение TRUE, если выполняются соответствующие условия: Сначала проверяется, находится ли частица у края(nearEdge). Если частица не находится рядом с краем, проверяется, не находится ли она рядом с соседом(foundfriend) или не достигла ли она требуемого радиуса(exitCircle). После проверки ме-

стоположения, если оно хорошее, начинается случайное диффундирование — определяется новое местоположение. Функция возвращает новое местоположение(X,Y), foundFriend (BOOLEAN), nearEdge (BOOLEAN), exitCircle(BOOLEAN)(рис. 3.4).

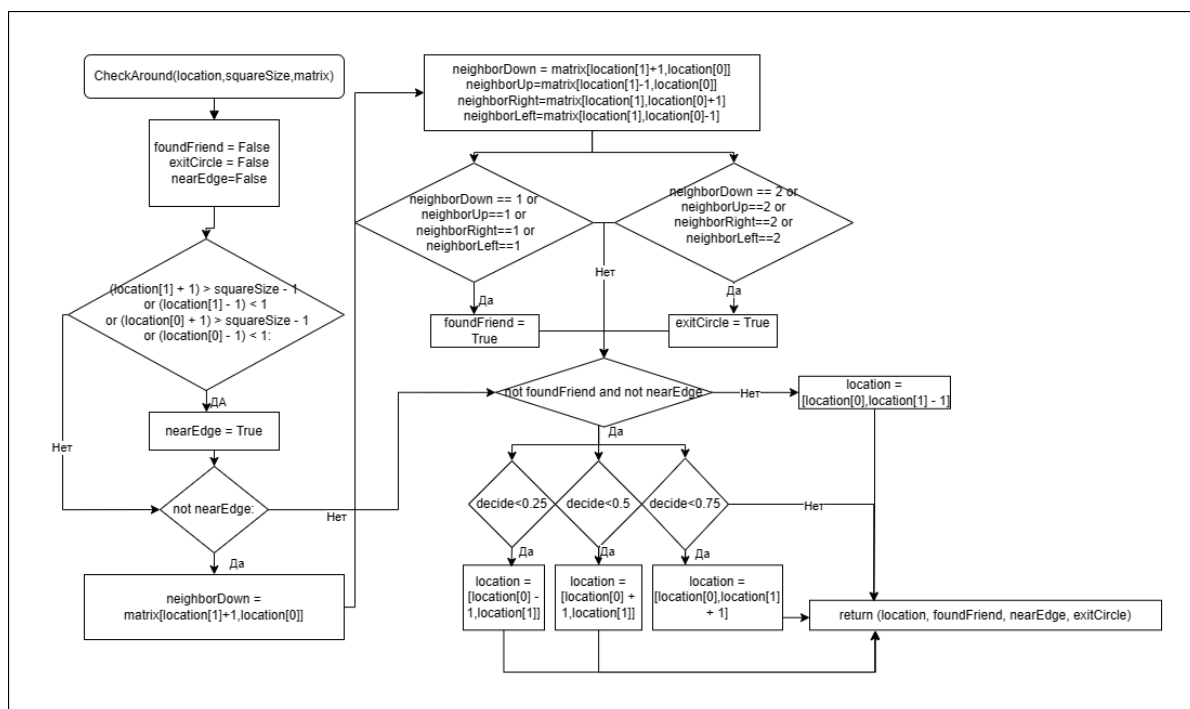


Рис. 3.4: Блок-схема алгоритма движения частицы

4 Выводы

В результате выполнения второго этапа проекта мы описали алгоритм DLA и алгоритм реализации этой модели с помощью блок-схем.

Список литературы

1. Sander L.M. Diffusion-limited aggregation: A kinetic critical phenomenon? Contemporary Physics, 2000.
2. Медведев Д.А. и др. Моделирование физических процессов и явлений на ПК: Учеб. пособие. Новосибирск: Новосиб. гос. ун-т, 2010. 101 с.