**РОССИЙСКИЙ УНИВЕРСИТЕТ ДРУЖБЫ НАРОДОВ**

**Факультет физико-математических и естественных наук**

**Кафедра прикладной информатики и теории вероятностей**

**Реферат по презентации «Комплексы программ по агрегации ограниченной диффузией»**

*дисциплина: Математическое моделирование*

Студенты: Аль-Дорихим Рамзи,

Ильинский Арсений Александрович,

Худицкий Василий Олегович,

Топонен Никита Андреевич

Группа: НКНбд-01-19

**МОСКВА**

2022 г.

Слайд 1

Добрый день, сегодня мы представим программную реализацию неравновесной агрегации на языке Java17. Данный доклад подготовили: Рамзи Аль-Дорихим, Василий Худицкий, Никита Топонен, Арсений Ильинский.

Слайд 2

Начнем с агрегации, ограниченной диффузией.

Слайд 3

1. Первым этапом идет создание окна размерности N\*N, а в нем создаем сетку с шагом 1, в которой будет осуществляться агрегация. Создаем двумерный массив частиц N на N, в котором будем хранить занятые ячейки.

2. Во втором пункте помещаем затравочную частицу на поле, установив флаг точки в двумерном массиве на true.

Слайд 4

Перед нами представлен цикл, который будет выполняться пока агрегация не достигнет границ.

Переменной m присваиваем случайное число от 0 до 1, от которого будет зависеть зона старта частицы. С равной вероятностью частица может вылететь с левой, правой, нижней или верхней частей поля. Для этого присвоим координаты частицы на одной из границ.

Слайд 5

Далее идет цикл, пока частица не вылетит за пределы окна, либо пока не приклеится к другой частице. В нем сначала частица делает случайный шаг на расстояние равное одному равновероятно в одну из четырех сторон: влево, вправо, вниз или вверх.

Слайд 6

Затем проверяем, занят ли соседний участок, то есть достигла ли наша частица агрегата. Если одно из соседних полей занято, то частица приклеивается, то есть занимает данное поле. Закрашиваем данный пиксель, и перерисовываем поле.

Далее проверяем достиг ли аггрегат границы. Если достиг – заканчиваем выполнение программы. Если нет – выпускаем новую частицу.

Слайд 7

На рисунке один мы видим фрактал, полученный с помощью агрегации, ограниченной диффузией.

Слайд 8

При диффузионно-ограниченной агрегации (DLA) частица всегда прилипает к кластеру с вероятностью 1, но мы можем уменьшить вероятность прилипания. И такой процесс роста будет называться химически-ограниченной агрегацией.

Слайд 9

1. Как видно, теперь мы задаем некоторую переменную n, и присваиваем ей значение от 0 до 1.  
2. После чего, мы проверяем, меньше ли наша переменная n, чем определенное заданное число (в нашем случае 0.3) – т.е. таким образом, наша переменная n, попадает в нужный диапазон с вероятностью в 0.3.  
3. Далее если наше условие выполняется, то приклеиваем частицу к кластеру.

Слайд 10

Как мы можем видеть на рисунке, данная модификация приведет к появлению более плотных агрегатов (увеличению размерности), потому что у частицы увеличится шанс проникать во внутренние области и заполнять пустоты.  
Но стоит отметить, что размерность остается меньше размерности пространства, таким образом, кластер остается фракталом.

Слайд 11

Баллистическая модель похожа на модель DLA. Отличие состоит в том, что направление движения частицы выбирается один раз, поэтому частица становится частью агрегата или выходит за границы радиуса уничтожения намного быстрее.

Слайд 12

Как видно, теперь частица выбирает направление движения один раз, и затем двигается прямолинейно до тех пор, пока не достигнет другой частицы, либо не вылетит за пределы поля.

Слайд 13

Как мы видим на рисунке 3, итоговая структура, сгенерированная при помощи данного метода, получается более плотной, чем при использовании DLA, поскольку в алгоритме частица движется прямолинейно, а не моделирует броуновское движение, которое вносит большую разветвленность в структуру.

Слайд 14

Далее рассмотрим кластер-кластерную агрегацию, особенность которой заключается в том, что кластеры частиц могут двигаться по полю и объединяться друг с другом.

Слайд 15

Будем хранить информацию о кластерах так: используем словарь, в котором ключом будет номер кластера, а значением динамически расширяемая коллекция частиц.

Слайд 16

Сначала необходимо заполнить поле частицами. Для этого для каждый частицы будем задавать случайную координату на поле и после проверки на то, что она уже не занята, создаем новый кластер и добавляем в него частицу.

Слайд 17

Само движение кластеров осуществим так: будем проходить по всему словарю от начала и до конца и двигать кластер вправо, влево, вверх или вниз. Затем для каждой частицы кластера проверяем все клетки вокруг нее на наличие нового соседа, не из ее кластера. Если таковой существует, происходит объединение кластеров. Такой процесс происходит пока все частицы не объединяться в один кластер.

Слайд 18

Выше использовались некоторые вспомогательные методы. Первый из них – это moveCluster. Он перемещает кластер следующим образом: для каждой частицы из кластера изменяется координата на одно и то же значение, а также эти изменения вносятся в массив занятых полей и перерисовывается изображение.

Слайд 19

И второй вспомогательный метод – это mergeClusters. Он отвечает за слияние кластеров, содержащих частицы. Сначала заполняются коллекции, содержащие первую и вторую частицы соответственно. Затем они сливаются в один, а второй остается пустым. Так как по словарю мы итерируемся по итератору, в Java нельзя изменять объекты данной коллекции. Поэтому будем сливать все в один кластер, а второй оставлять пустым, а не удалять.

Слайд 20

На данном изображении видно, что имеется множество кластеров, расположенных по всему полю.