

Групповой проект. Этап 3

Описание программной реализации.

Дворкина Е.В. Чемоданова А.А. Серёгина И.А. Волгин И.А. Александрова У.В. Голощапов Я.В.

07 мая 2025

Российский университет дружбы народов, Москва, Россия

Информация

- Серёгина Ирина Андреевна
- Студентка учебной группы НФИбд-01-22
- Российский университет дружбы народов
- <https://github.com/irinaseregina>



Вводная часть

- Написать программу, моделирующую теплопроводность.
- Добавить в модель процесс затвердевания.
- Исследовать, как со временем изменяются количество частиц в агрегате и его среднеквадратичный радиус в различных режимах.
- Определить фрактальную размерность полученных структур
- Проанализировать, как величина теплового шума влияет на морфологию формирующихся агрегатов.

Выполнение задач третьего этапа

Задание базовых параметров моделирования

```
N = 150          # Размер сетки (N x N)
T_initial = -1   # Начальная температура в центральной точке
steps = 200      # Количество временных шагов
dt = 1           # Шаг по времени
h = 1            # Расстояние между узлами
kappa = 0.1      # Коэффициент теплопроводности
w = 0.5          # Коэффициент для диагональных соседей
T_m = 0          # Температура плавления
lambda = 0.01    # Капиллярный радиус
delta = 0.02     # Величина флуктуаций температуры
```

```
T = zeros(N, N)           # Матрица температур
n = zeros(Int, N, N)      # Матрица состояний (0 - жидкое, 1 - твердое)
T[N÷2+1, N÷2+1] = T_initial  # Установка начальной температуры в центральной
n[N÷2+1, N÷2+1] = 1
```


- Метод полиномиальной аппроксимации
- Среднее значение температуры
- Кривизна границы
- Количества затвердевших частиц
- Среднеквадратичный радиус

```
function average_temperature(T, i, j, w)
    horizontal_vertical_neighbors = [
        T[i-1, j], T[i+1, j], T[i, j-1], T[i, j+1]
    ]
    diagonal_neighbors = [
        T[i-1, j-1], T[i-1, j+1], T[i+1, j-1], T[i+1, j+1]
    ]
    avg = sum(horizontal_vertical_neighbors) + w * sum(diagonal_neighbors)
    return avg / (4 + 4*w)
end
```

Рис. 1: Функция average_temperature

- Метод полиномиальной аппроксимации
- Среднее значение температуры
- Кривизна границы
- Количества затвердевших частиц
- Среднеквадратичный радиус

```
function curvature(n, i, j, w)
    horizontal_vertical_neighbors = [
        n[i-1, j], n[i+1, j], n[i, j-1], n[i, j+1]
    ]
    diagonal_neighbors = [
        n[i-1, j-1], n[i-1, j+1], n[i+1, j-1], n[i+1, j+1]
    ]
    sum_hv = sum(horizontal_vertical_neighbors)
```

Рис. 2: Функция curvate

- Метод полиномиальной аппроксимации
- Среднее значение температуры
- Кривизна границы
- Количества затвердевших частиц
- Среднеквадратичный радиус

```
function count_solid_particles(n)
    return sum(n)
end

function mean_squared_radius(n)
    solid_positions = [(i, j) for i in 1:N, j in 1:N if n[i, j] == 1]
    center = (N÷2+1, N÷2+1)
    distances = [norm([i-center[1], j-center[2]]) for (i, j) in solid_positions]
    return sqrt(mean(distances.^2))
end
```

Рис. 3: Функции count_solid_particles и mean_squared_radius

Функция `simulate_heat_conduction` на основе уравнения обновления температуры:

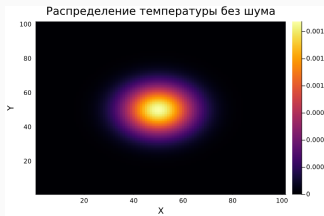


Рис. 4: Распределение температуры без шума

```
function simulate_heat_conduction(N, steps, κ)
    T = zeros(N, N)
    center = div(N, 2)
    T[center, center] = 1.0 # начальная температура в центре

    for step in 1:steps
        T_temp = copy(T)
        for i in 2:N-1
            for j in 2:N-1
                T_temp[i, j] = T[i, j] + κ * (T[i+1, j] + T[i-1, j] + T[i, j+1] + T[i, j-1] - 4 * T[i, j])
            end
        end
        T .= T_temp
    end

    heatmap(T, title="Распределение температуры без шума", xlabel="X", ylabel="Y")
end
```

Рис. 5: Функция `simulate_heat_conduction`

Реализована функция

`simulate_solidification`, которая

выполняет следующие шаги:

1. Обновление температур
2. Проверка условия затвердевания
3. Обновление состояний

```
for i in 2:size(n, 1)-1
    for j in 2:size(n, 2)-1
        if n[i, j] == 0 # Только для жидких узлов
            # Проверяем наличие соседей в твердой фазе
            neighbors = [n[i-1, j], n[i+1, j], n[i, j-1], n[i, j+1],
                        n[i-1, j-1], n[i-1, j+1], n[i+1, j-1], n[i+1, j+1]]
            if any(neighbors .== 1) # Если есть хотя бы один твердый сосед
                # Вычисляем кривизну границы
                s_ij = curvature(n, i, j, w)

                # Вычисляем локальную температуру плавления
                local_T_m = T_m + λ * s_ij

                # Проверяем условие затвердевания
                if T_temp[i, j] <= local_T_m
                    n_temp[i, j] = 1 # Узел затвердевает
                end
            end
        end
    end
end

# Обновляем основные матрицы
T .= T_temp
n .= n_temp
```

Рис. 6: Фрагмент функции `simulate_solidification`

Результаты моделирования. Исследование влияния капиллярного радиуса

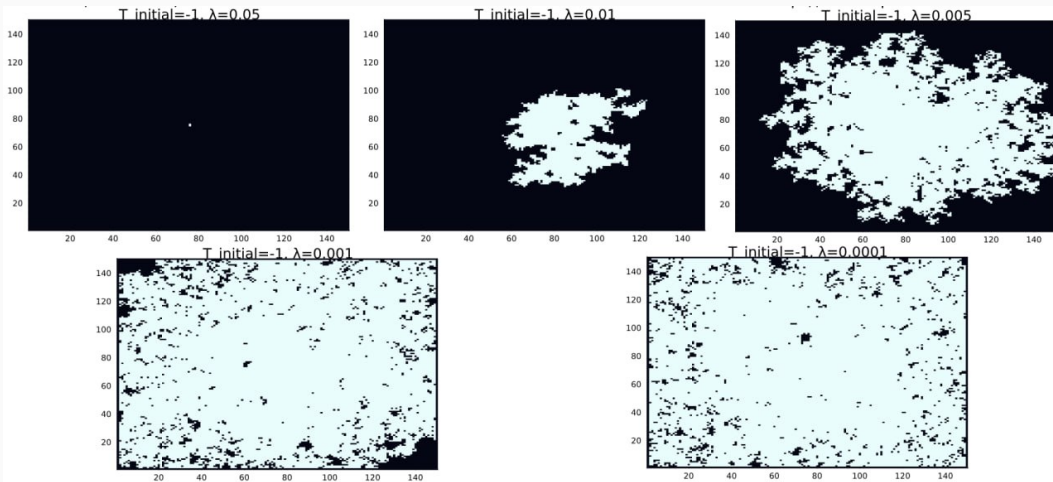


Рис. 7: Исследование влияния начального переохлаждения и величины капиллярного радиуса

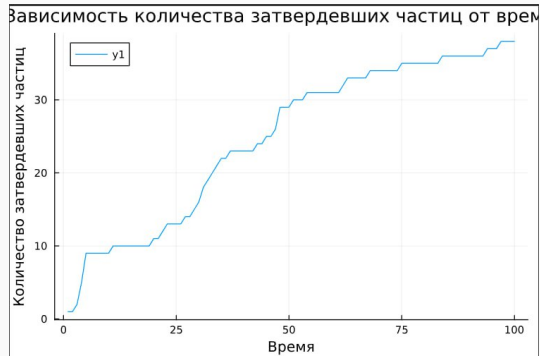


Рис. 8: Зависимость числа затвердевших частиц от времени

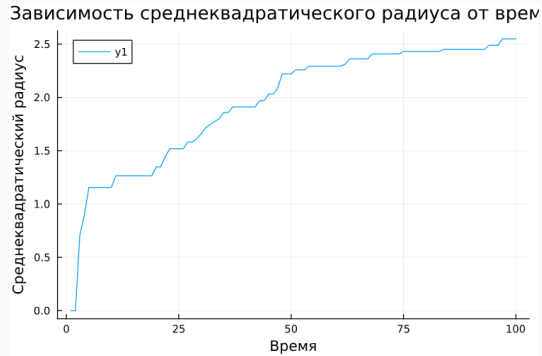


Рис. 9: Зависимость среднеквадратического радиуса от времени

Фрактальная размерность

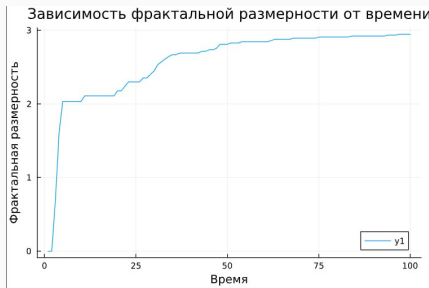


Рис. 10: Зависимость фрактальной размерности от времени

Фрактальную размерность D можно определить через логарифмическую регрессию:

$$D = \frac{\log N(r)}{\log r} \quad ()$$

где:

- $N(r)$ - количество частиц внутри радиуса r
- D - искомая фрактальная размерность

Реализована функция `fractal_dimension`

Влияние теплового шума

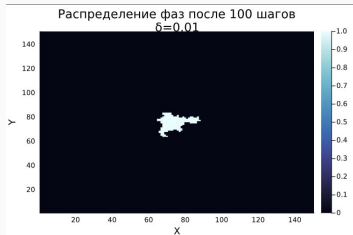


Рис. 11: Значение теплового шума (δ) 0.01

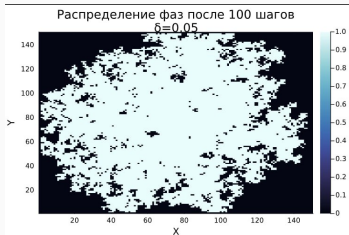


Рис. 12: Значение теплового шума (δ) 0.05

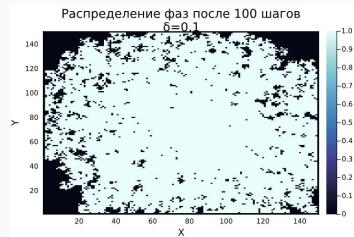


Рис. 13: Значение теплового шума (δ) 0.1

1. Смоделирован процесс теплопроводности.
2. Исследовано влияние начального переохлаждения и капиллярного радиуса на форму дендритов.
3. Проанализирована динамика роста агрегата и его фрактальная размерность.
4. Изучено влияние теплового шума на морфологию агрегатов.

Результаты показывают, что:

- Тепловой шум значительно влияет на структуру дендритов, увеличивая их нерегулярность и скорость роста.