Рост дендритов

Этап №4

Миронов Д. А. Павлова П. А. Матюшкин Д. В.

20 марта 2024

Российский университет дружбы народов, Москва, Россия

Цели и задачи

Целью проекта является математическое моделирование дендритного роста.

Задачи проекта:

1. Изучить теоретическую информацию о дендритах и о моделях их роста.
2. Разработать алгоритм, который включает в себя:

- моделирование теплопроводности;
- исследование влияние начального переохлаждения S и величины капиллярного радиуса λ на форму образующихся дендритов;
- исследование зависимость от времени числа частиц в агрегате и его среднеквадратичного радиуса в разных режимах;
- определение фрактальной размерности полученных образцов;
- исследвание влияния величины теплового шума δ на вид образующихся агрегатов;

3. Написать комплексы программ по разработанному алгоритму;

Описание явления роста дендритов

Дендриты - это маленькие ветвистые образования, похожие на деревья или ветви, которые могут появляться в разных системах, от нервных клеток до кристаллов в металлах.

Самые распространённые структуры морозных узоров — дендриты.

Модель роста дентритов

Пусть у нас есть квадратная область размера N × N узлов, в центре которой задана некоторая затравка (небольшая затвердевшая область, на границе которой происходит дальнейшая кристаллизация).
Расстояние между узлами по горизонтали и вертикали обозначим h, а шаг по времени Δt.

- $h = 1$
- $\Delta t = 1$

Уравнение теплопроводности

Изменение температуры со временем описывается уравнением теплопроводности:

$$\rho c_p \frac{\partial T}{\partial t} = \kappa \nabla^2 T \equiv \kappa \left(\frac{\partial^2 T}{\partial x^2} + \frac{\partial^2 T}{\partial y^2} \right)$$

Свойства вещества:

- ρ - плотность
- c_p - теплоемкость при постоянном давлении
- κ - коэффициент теплопроводности
- T - температура плавления

Среднее значение

Величина $\nabla^2 T$ в узле (i, j) может быть записана как разница среднего значения температуры в соседних узлах $\langle T_{(i,j)} \rangle$ и температуры в самом узле, $\nabla^2 T \approx (\langle T_{(i,j)} \rangle - T_{(i,j)})/h^2$.

Общая формула среднего значения:

$$\langle T_{(i,j)} \rangle = (T_{(i+1,j)} + T_{(i-1,j)} + T_{(i,j+1)} + T_{(i,j-1)} +$$

$$+ w(T_{(i+1,j+1)} + T_{(i+1,j-1)} + T_{(i-1,j+1)} + T_{(i-1,j-1)}))/(4 + 4w)$$

Коэффициент $0 \leq w < 1$ учитывает влияние диагональных соседей.

Среднее значение

Строго говоря, $\nabla^2 T \approx \frac{\langle T_{(i,j)} \rangle - T_{(i,j)}}{(4+4w)(1+2w)h^2}$.

Новое значение температуры после каждого такого шага вычисляется как $\widehat{T_{(i,j)}} = T_{(i,j)} + \chi \Delta t \nabla^2 T/m$, такая схема устойчива при $\chi \Delta t/(mh^2) < 1/4$.

Рост дендрита

Состояние каждого узла n:

- $n = 0$ соответствует жидкой фазе
- $n = 1$ - твердой

Промежуточные состояния учитывать не будем.

Всего может быть четыре ближайших соседа и четыре диагональных. Разумно считать, что граница плоская, когда $n = 1$ у пяти соседей

$$1/R \approx s_{i,j} = \sum_1 n_{i,j} + w_n \sum_2 n_{i,j} - (\frac{5}{2} + \frac{5}{2} w_n)$$

Здесь первая сумма – по ближайшим соседям, вторая – по диагональным. Коэффициент $0 \leq w_n \leq 1$ учитывает ослабление влияния соседей с ростом расстояния.

Тепловой шум

Необходимо еще учитывать тепловой шум. В простейшем случае можно прибавлять к температуре в узле некоторую случайную добавку $\eta_{i,j}\delta$, где $\eta_{i,j}\delta$ - случайное число, равномерно распределенное в интервале [-1,1], а δ — величина флуктуаций температуры.

Узел, расположенный на границе, меняет свое состояние с жидкого на твердое, если

$$T \leq \widetilde{T}_m(1 + \eta_{i,j}\delta) + \lambda s_{i,j}$$

- T_m - температура плавления
- $\widetilde{T}_m$ - безразмерное начальное переохлаждение
- λ - капиллярный радиус

Алгоритм роста дендритов

Алгоритм, используемый в проекте для моделирования роста дендритов, включает в себя несколько шагов:

1. Инициализация: Задаются начальные условия, такие как размеры области, начальные значения температуры и состояния узлов.
2. Уравнение теплопроводности: Используется уравнение теплопроводности для моделирования распространения тепла в системе. Оно описывает, как температура изменяется со временем и пространством внутри области.
3. Вычисление среднего значения температуры в соседних узлах: Для каждого узла вычисляется среднее значение температуры с учетом температур соседних узлов и их коэффициентов.

Алгоритм роста дендритов

4. Обновление температуры в каждом узле: Новое значение температуры вычисляется на основе предыдущего значения температуры, шага по времени и градиента температуры.

5. Моделирование роста дендритов: Учитываются состояния каждого узла, а также влияние соседей на процесс роста дендритов.

6. Учет теплового шума: К температуре каждого узла добавляется случайная добавка, моделирующая тепловой шум.

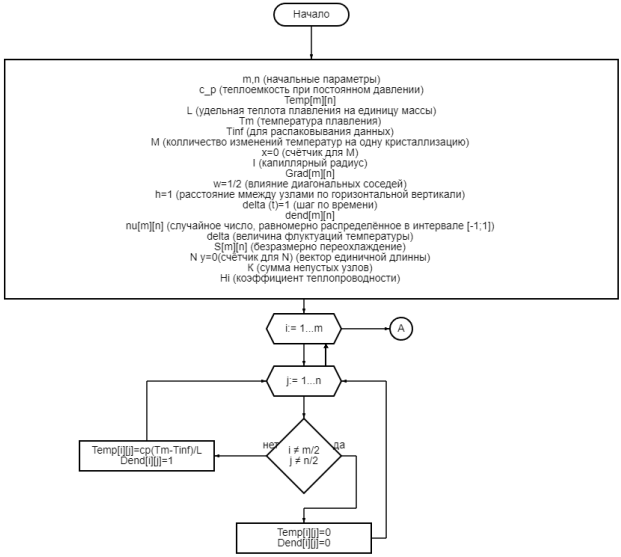
7. Изменение состояния узлов на границе: Узлы на границе могут изменять свое состояние с жидкого на твердое в зависимости от условий роста дендритов.

Преимущества алгоритма

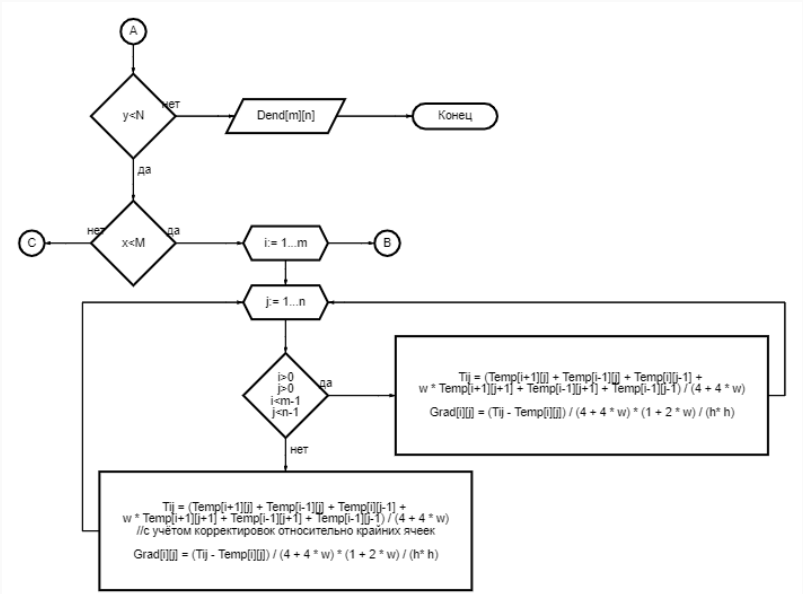
Этот алгоритм моделирования роста дендритов имеет несколько преимуществ:

1. Учет различных факторов.
2. Адаптивность.
3. Вычислительная эффективность.

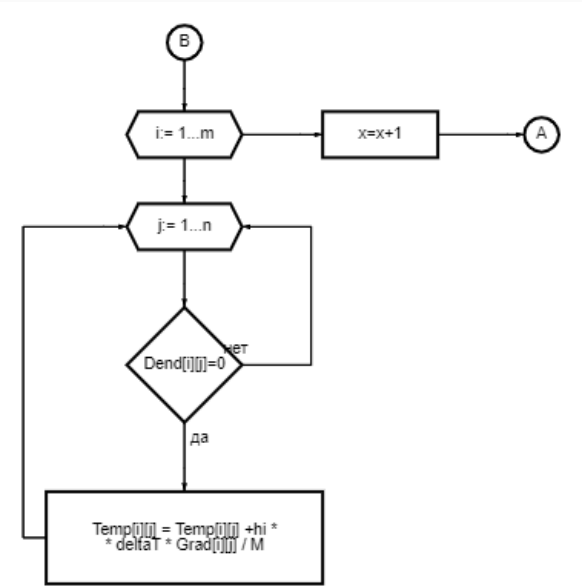
Алгоритм роста дендритов



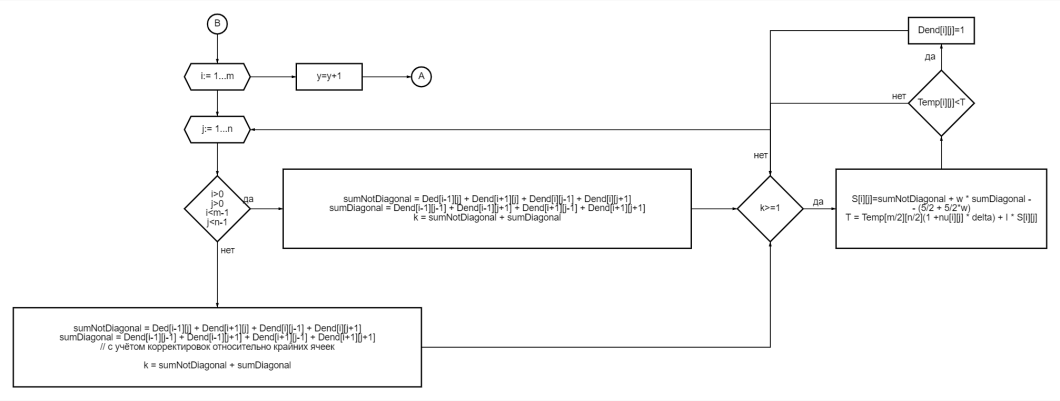
Алгоритм роста дендритов



Алгоритм роста дендритов



Алгоритм роста дендритов



Основные блоки кода

1. Инициализация необходимых переменных;
2. Обработка диффузии температуры через пространство моделирования;
3. Обновление значения температуры вдоль определенного пути;
4. Определение, где происходит рост дендритов на основе определенных условий;
5. Сохранение результата в виде фотографии.

Программная реализация

```
const M = 12
const N = 120
const w = 2.5
const h = 1.0
const deltaT = 1.0

const cp = 3.8
const L = 4.42

const Tm = 5
const Tinf = 6

const l = 4.31323
const delta = 139.547327
const hi = 100

const rows, cols = 131, 131
Dend = zeros(Float64, rows, cols)
Temp = zeros(Float64, rows, cols)
Dend[Int((rows - 1) / 2), Int((cols - 1) / 2)] = 1
Temp[Int((rows - 1) / 2), Int((cols - 1) / 2)] = cp * (Tm - Tinf) / L
Grad = zeros(Float64, rows, cols)

global x = 0
global y = 0
const dendrite_size = 10
```

Программная реализация

```
function part_A()
    global y
    if y < N
        global x
        if x < M
            for i in 1:rows
                for j in 1:cols
                    Tij_sum = 0
                    Tij_sum_w = 0

                    if i > 1
                        Tij_sum += Temp[i - 1, j]
                        if j > 1
                            Tij_sum_w += Temp[i - 1, j - 1]
                        end
                    end

                    if j > 1
                        Tij_sum += Temp[i, j - 1]
                        if i < rows
                            Tij_sum_w += Temp[i + 1, j - 1]
                        end
                    end

                    if i < rows
                        Tij_sum += Temp[i + 1, j]
                        if j > 1
                            Tij_sum_w += Temp[i + 1, j - 1]
                        end
                    end

                    if j < cols
                        Tij_sum += Temp[i, j + 1]
                        if i < rows
                            Tij_sum += Temp[i + 1, j + 1]
                        end
                    end

                    Tij = (Tij_sum + w * Tij_sum_w) / (4 + 4 * w)
                    Grad[i, j] = (Tij - Temp[i, j]) / ((4 + 4 * w) * (1 + 2 * w) * (h * h))
                    if Dend[i, j] == 0 && rand() < 0.05 * dendrite_size
                        Dend[i, j] = 1
                        Temp[i, j] = Temp[Int((rows - 1) / 2), Int((cols - 1) / 2)]
                    end
                end
            end
        end
    end
end
```

Программная реализация

```
function part_B()
    global x
    for i in Int((rows - 1) / 2):-1:1
        for j in Int((cols - 1) / 2):-1:1
            if Dend[i, j] == 0
                Temp[i, j] = Temp[i, j] + hi * deltaT * Grad[i, j] / M
            end
        end
    end

    for i in Int((rows - 1) / 2):rows
        for j in Int((cols - 1) / 2):cols
            if Dend[i, j] == 0
                Temp[i, j] = Temp[i, j] + hi * deltaT * Grad[i, j] / M
            end
        end
    end

    global x += 1
    part_A()
end
```

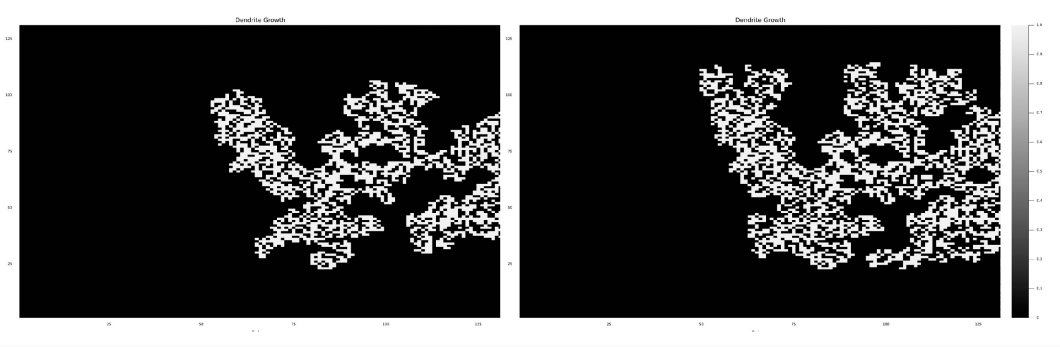
Программная реализация

```
function part_C()
    global y
    for i in 1:rows
        for j in 1:cols
            sum_not_diagonal = 0
            sum_diagonal = 0

            if i > 1
                sum_not_diagonal += Dend[i - 1, j]
                if j > 1
                    sum_diagonal += Dend[i - 1, j - 1]
                end
            end

            if j > 1
                sum_not_diagonal += Dend[i, j - 1]
                if i < rows
                    sum_diagonal += Dend[i + 1, j - 1]
                end
            end

            if i < rows
                sum_not_diagonal += Dend[i + 1, j]
                if j > 1
                    sum_diagonal += Dend[i + 1, j - 1]
                end
            end

            if j < cols
                sum_not_diagonal += Dend[i, j + 1]
                if i < rows
                    sum_diagonal += Dend[i + 1, j + 1]
                end
            end

            k = sum_diagonal + sum_not_diagonal
            if k >= 1
                S = sum_not_diagonal + w * sum_diagonal - (2.5 + 2.5 * w)
                T = Temp[Int((rows - 1) / 2), Int((cols - 1) / 2)] * (1 + nu[i, j] * delta) + 1 * S

                if Temp[i, j] < T
                    Dend[i, j] = 1
                    Temp[i, j] = Temp[Int((rows - 1) / 2), Int((cols - 1) / 2)]
                end
            end
        end
    end
end
```

Программная реализация

```
heatmap(Dend, color=:grays, c=:grays, xlims=(1, cols), ylims=(1, rows), xlabel="Column", ylabel="Row", title="Dendrite Growth", size=(1920, 1080))

savefig("dendrite_growth.png")
```

Результат



Заключение

- В заключении проекта по моделированию дендритного роста мы успешно создали алгоритм, учитывающий различные факторы, такие как теплопроводность, начальное переохлаждение, капиллярный радиус и тепловой шум. Написали комплексы программ по разработанному алгоритму.