

Групповой проект. Этап 4

Рост дендритов

Доборщук Владимир, НФИбд-01-18

Голова Варвара, НФИбд-03-18

Дяченко Злата, НФИбд-03-18

Карташова Алиса, НФИбд-03-18

Родина Дарья, НФИбд-03-18

Цели и задачи группового проекта

Цель проекта: освоение принципов математического моделирования посредством изучения и построения модели роста дендритов.

Задачи проекта:

1. изучить теоретическую информацию о дендритах, в том числе и о модели роста дендритов;
2. разработать алгоритмы, позволяющие:
 - моделировать теплопроводность;
 - исследовать влияние величины капиллярного радиуса λ на форму образующихся дендритов;
3. написать программу, взяв в основу разработанные ранее алгоритмы;
4. проанализировать полученные результаты.

Что есть дендриты?

Что есть дендриты?

Дендрит (от гр. *το δένδρον* - дерево) - ветвящееся и расходящееся в стороны образование, возникающее при ускоренной или стеснённой кристаллизации в неравновесных условиях, когда кристалл расщепляется по определённым законам.

Дендритный рост - процесс образования дендрита.

Основные понятия

- ρ - плотность
- L - удельная теплота плавления на единицу массы
- c_p - теплоемкость при постоянном давлении (также на единицу массы)
- k - коэффициент теплопроводности
- T_m - температура плавления
- $T_\infty < T_m$
- $d_0 = \gamma T_m c_p / (\rho L^2)$ - капиллярный радиус
- $\chi = k / \rho c_p$ - коэффициент температуропроводности
- $c_p (T_m - T_\infty) / L$ - безразмерное переохлаждение

Теоретическая информация

Пусть у нас есть квадратная область размера $N * N$ узлов, в центре которой задана некоторая затравка.

$h = 1$ - расстояние между узлами по горизонтали и вертикали

$\Delta t = 1$ - шаг по времени

Уравнение теплопроводности

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

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

где $0 \leq w < 1$ - коэффициент, учитывающий влияние диагональных соседей

Общая формула для вычисления среднего значения температуры в соседних узлах: $\langle T_{(i,j)} \rangle = (T_{i+1,j} + T_{i-1,j} + T_{i,j+1} + T_{i,j-1} + w(T_{i+1,j+1} + T_{i+1,j-1} + T_{i-1,j+1} + T_{i-1,j-1}))/ (4 + 4w)$

Новое значение температуры после каждого шага:

$$\hat{T}_{i,j} = T_{i,j} + \chi \Delta t \nabla^2 T / m, \text{ что устойчиво при } \chi \Delta t / (mh^2) < 1/4.$$

Состояния каждого узла n :

- $n = 0$ - жидкая фаза
- $n = 1$ - твердая фаза
- промежуточные состояния не учитываются

Всего может быть четыре ближайших соседа и четыре диагональных.

$n = 1$ у пяти соседей - *граница плоская*:

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

где **первая сумма** - по ближайшим соседям, **вторая** - по диагональным,
 $0 \leq w_n \leq 1$ - коэффициент, учитывает ослабление влияния соседей с ростом расстояния.

Также необходимо учитывать **тепловой шум**.

Простейший случай: к температуре в узле прибавляется некоторая случайная добавка $n_{i,j}\delta$, где $n_{i,j}$ - случайное число, равномерно распределенное в интервале $[-1, 1]$, а δ - величина флуктуаций температуры.

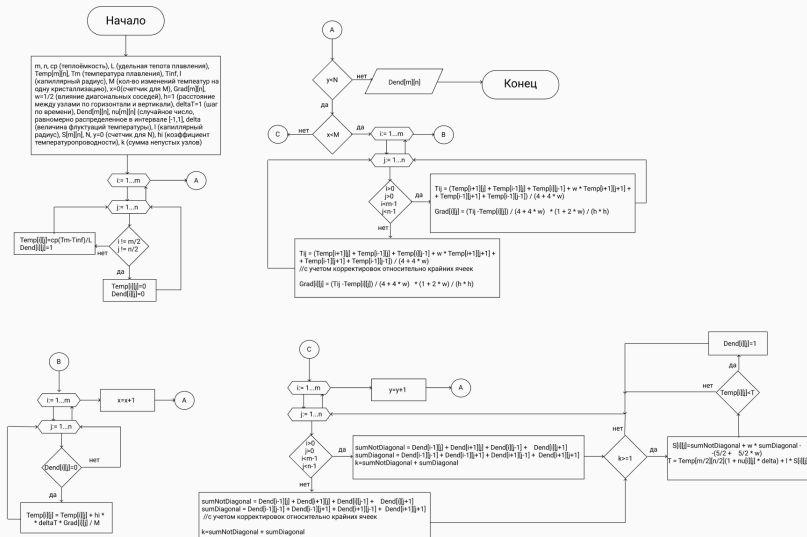
Узел, расположенный на границе, меняет свое состояние с жидкого на твердое, если

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

где T_m - температура плавления, \tilde{T}_m - безразмерное начальное переохлаждение, λ - капиллярный радиус.

Модель роста дендритов

Алгоритм



Инициализация библиотек

```
import numpy as np
import matplotlib.pyplot as plt
```

Начальные значения

```
M = 12
x=0
N = 120
y=0

w = 0.5
h = 1
deltaT = 1

cp = 3.8
L = 4.42

Tm = 5
Tinf = 6

l = 4.31323
delta = 139.547327
k = 0
hi = 414.32

rows, cols = 131, 131 # m,n
Dend = np.zeros((rows,cols))
Temp = np.zeros((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

nu = (1 - (-1)) * np.random.random((rows,cols)) - 1

Grad = np.zeros((rows,cols))
S = np.zeros((rows,cols))
```


Первый блок программы (A)

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

                    if i > 0:
                        Tij_sum += Temp[i-1][j]
                        if j > 0:
                            Tij_sum_w += Temp[i-1][j-1]
                    if j > 0:
                        Tij_sum += Temp[i][j-1]
                        if i < rows-1:
                            Tij_sum_w += Temp[i+1][j-1]
                    if i < rows-1:
                        Tij_sum += Temp[i+1][j]
                        if j > 0:
                            Tij_sum_w += Temp[i+1][j-1]
                    if j < cols-1:
                        Tij_sum += Temp[i][j+1]
                        if i < rows-1:
                            Tij_sum += Temp[i+1][j+1]

                    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))

        part_B()
    else:
        part_C()
```

Второй блок программы (В и С)

```
def part_C():
    global y
    for i in range(rows):
        for j in range(cols):
            sum_not_diagonal = 0
            sum_diagonal = 0
            if i > 0:
                sum_not_diagonal += Dend[i-1][j]
                if j > 0:
                    sum_diagonal += Dend[i-1][j-1]
            if j > 0:
                sum_not_diagonal += Dend[i][j-1]
                if i < rows-1:
                    sum_diagonal += Dend[i+1][j-1]
            if i < rows-1:
                sum_not_diagonal += Dend[i+1][j]
                if j > 0:
                    sum_diagonal += Dend[i+1][j-1]
            if j < cols-1:
                sum_not_diagonal += Dend[i][j+1]
                if i < rows-1:
                    sum_diagonal += Dend[i+1][j+1]

            k = sum_diagonal + sum_not_diagonal
            if k >= 1:
                S[i][j] = 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)+l*S[i][j]

                if Temp[i][j] < T:
                    Dend[i][j] = 1
                    Temp[i][j] = Temp[int((rows-1)/2), int((cols-1)/2)]

        y+=1

    rgba = cmap(norm(Dend))
    plt.pause(1e-12)
    plt.clf()
    plt.imshow(rgba, interpolation='nearest')
    part_A()
```

Вывод результатов

```
cmap = plt.cm.gray
norm = plt.Normalize(Dend.min(), Dend.max())
rgba = cmap(norm(Dend))

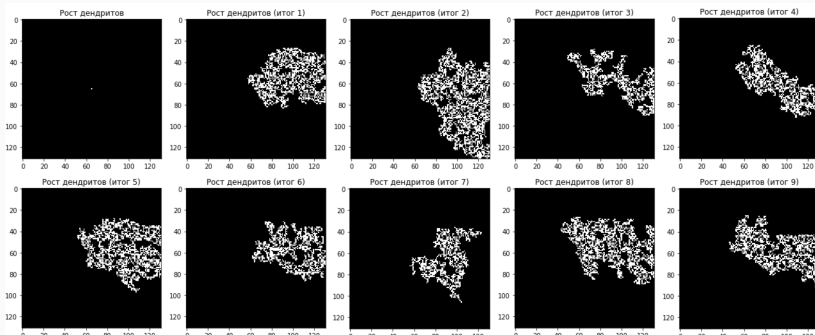
plt.title('Рост дендритов')
plt.imshow(rgba, interpolation='nearest')

part_A()

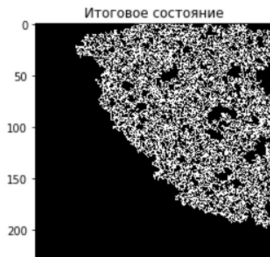
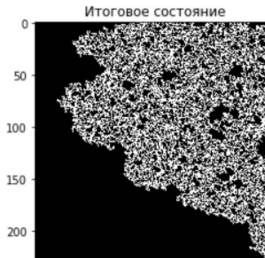
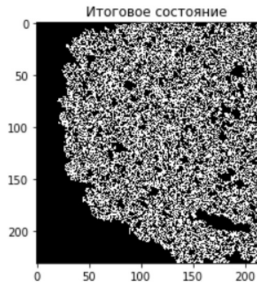
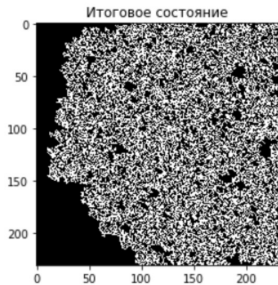
rgba = cmap(norm(Dend))

plt.title('Рост дендритов (итог)')
plt.imshow(rgba, interpolation='nearest')
```

Примеры построенных дендритов



Исследование влияния величины капиллярного радиуса на форму дендрита



Выводы

В работе рассмотрен комплекс вопросов, касающихся механизма образования дендритов. Мы ответили на исследовательский вопрос, как происходит рост дендритов, образованных во время формирования модели и построили численную модель, дающую представление о механизмах и причинах роста кристаллов. Написали программу, моделирующую теплопроводность. Задали начальную температуру равной нулю везде, кроме центральной точки и посмотрели, как меняется распределение температуры. Добавили затвердевание, исследовали влияние величины капиллярного радиуса λ на форму образующихся дендритов.