

Групповой проект. Этап 3

Рост дендритов

Дорук Мерич, Мухамедиар Адиль, Яссин Мохамад Аламин, Акуленко Максим

Цели и задачи группового проекта

Цель проекта: На основе построения модели роста дендритов, изучить принципы математического моделирования.

Задачи проекта:

1. изучить теоретическую информацию о дендритах и о моделях их роста;
2. разработать алгоритмы
3. написать программу, взяв в основу разработанные ранее алгоритмы;
4. проанализировать полученные результаты.

Основные понятия и уравнения

- M - Количество итераций(шагов)
- x - удельная теплота плавления на единицу массы
- y - теплоемкость при постоянном давлении (также на единицу массы)
- k - коэффициент теплопроводности
- T_m - температура плавления
- $T_\infty < T_m$
- ω - Коэффициент, учитывающий влияние диагональных соседей
- h - Расстояние между узлами по вертикали и горизонтали
- δ - Шаг по времени
- c_p - Теплоемкость при постоянном давлении на единицу массы
- L - Удельная теплота плавления на единицу массы
- $Grad$ - Массив для хранения температурного градиента
- $Cal - grad - tem$ - Вычисляет температурный градиент в каждой ячейке(интерация по всем, кроме границ + для каждого

Код проекта Алгоритм роста дендритов

```
1 using Random, LinearAlgebra
2 using PyPlot
3
4 M = 12
5 x = 0
6 y = 0
7 N = 120
8
9
10 w = 0.5
11 h = 1
12 deltaT = 1
13
14 cp = 3.8
15 L = 4.42
16
17 Tc = 5
18 Tinf = 6
19
20 l = 4.31323
21 delta = 139.547327
22 k = 0
23 hl = 414.32
24
25 rows, cols = 131, 131 #Cetuxa NxN
26 Dend = zeros(rows, cols)
27 Temp = zeros(rows, cols)
28 Dend[div(rows - 1, 2)+1, div(cols - 1, 2)+1] = 1
29 Temp[div(rows - 1, 2)+1, div(cols - 1, 2)+1] = cp * (Tm - Tinf) / L
30
31 Random.seed!(1234)
32 nu = (1 - (-1)) .* rand(rows, cols) ./- 1
33
34 Grad = zeros(rows, cols)
35 S = zeros(rows, cols)
```

Рис. 1: Codeblock 1

Код проекта Алгоритм роста дендритов

```
37 function cal_grad_tem()
38     global x
39     global Dend, Temp, Grad, S
40
41     if y < N
42         if x < M
43             for i in 1:rows
44                 for j in 1:cols
45                     Tij_sum = 0
46                     Tij_sum_w = 0
47
48                     if i > 1
49                         Tij_sum += Temp[i-1, j]
50                         if j > 1
51                             Tij_sum_w += Temp[i-1, j-1]
52                         end
53                     end
54
55                     if j > 1
56                         Tij_sum += Temp[i, j-1]
57                         if i < rows
58                             Tij_sum_w += Temp[i+1, j-1]
59                         end
60                     end
61
62                     if i < rows
63                         Tij_sum += Temp[i+1, j]
64                         if j > 1
65                             Tij_sum_w += Temp[i+1, j-1]
66                         end
67                     end
68
69                     if j < cols
70                         Tij_sum += Temp[i, j+1]
71                         if i < rows
```

Рис. 2: Codeblock 2

Код проекта Алгоритм роста дендритов

```
71         if i < rows
72             T1j_sum_M += Temp[i+1, j+1]
73         end
74     end
75
76     T1j = (T1j_sum + w * T1j_sum_M) / (4 + 4 * w)
77     Grad[i, j] = (T1j - Temp[i, j]) / ((4 + 4 * w) * (1 + 2 * w) * (h * h))
78 end
79
80     upd_temp()
81 else
82     status()
83 end
84 end
85 end
86 end
87
88 function upd_temp()
89     global x
90     global Dend, Temp, Grad
91
92     for i in div(rows - 1, 2):1:1
93         for j in div(cols - 1, 2):1:1
94             if Dend[i, j] == 0
95                 Temp[i, j] += h1 * deltaT * Grad[i, j] / M
96             end
97         end
98     end
99
100     for i in div(rows - 1, 2)+1:rows
101         for j in div(cols - 1, 2)+1:cols
102             if Dend[i, j] == 0
103                 Temp[i, j] += h1 * deltaT * Grad[i, j] / M
104             end
105         end
106     end
107 end
```

Рис. 3: Codeblock 3


```
106     end
107
108     N += 1
109     cal_grad_tem()
110 end
111
112 function status()
113     global y
114     for i in 1:rows
115         for j in 1:cols
116             sum_not_diagonal = 0
117             sum_diagonal = 0
118             if i > 1
119                 sum_not_diagonal += Dend[i-1, j]
120                 if j > 1
121                     sum_diagonal += Dend[i-1, j-1]
122                 end
123             end
124             if j > 1
125                 sum_not_diagonal += Dend[i, j-1]
126                 if i < rows
127                     sum_diagonal += Dend[i+1, j-1]
128                 end
129             end
130             if i < rows
131                 sum_not_diagonal += Dend[i+1, j]
132                 if j > 1
133                     sum_diagonal += Dend[i+1, j-1]
134                 end
135             end
136             if j < cols
137                 sum_not_diagonal += Dend[i, j+1]
138                 if i < rows
139                     sum_diagonal += Dend[i+1, j+1]
140                 end
141             end
142         end
143     end
144 end
```

Рис. 4: Codeblock 4

Код проекта Алгоритм роста дендритов

```
140         end
141     end
142     k = sum_diagonal + sum_not_diagonal
143     if k >= 1
144         S[i, j] = sum_not_diagonal + w * sum_diagonal - (2.5 + 2.5 * w)
145         T = Temp[div(rows - 1, 2)+1, div(cols - 1, 2)+1] * (1 + nu[i, j] * delta + 1 * S[i, j])
146
147         if Temp[i, j] < T
148             Dend[i, j] = 1 #1 - состояние, при достижении которой ячейка превращается в дендрит
149             Temp[i, j] = Temp[div(rows - 1, 2)+1, div(cols - 1, 2)+1]
150         end
151     end
152 end
153 end
154 y += 1
155
156 rgba = cmap(norm(Dend))
157 plt.pause(1e-12)
158 plt.clf()
159 plt.imshow(rgba, interpolation="nearest")
160 cal_grad_tew() # вызывает снова A, для повторной проверки T уже не ячейки, а дендрита
161 end
162
163 cmap = matplotlib.cm.gray
164 norm = matplotlib.colors.Normalize(vmin=minimum(Dend), vmax=maximum(Dend))
165 rgba = cmap.(norm.(Dend))
166
167 pyplot.title("Dendritic Growth.")
168 pyplot.imshow(rgba, interpolation="nearest")
169
170 cal_grad_tew()
171 rgba = cmap.(norm.(Dend))
172 pyplot.title("Final Dendritic Growth.")
173 pyplot.imshow(rgba, interpolation="nearest")
```

Рис. 5: Codeblock 5

Код проекта Алгоритм роста дендритов

```
81 algorithm
82   if  $y < n$  then
83     if  $x < n$  then
84       for  $i$  in 1:rows loop
85         for  $j$  in 1:cols loop
86            $T[i,j]_{sum} := 0$ ;
87            $T[i,j]_{sum,w} := 0$ ;
88           if  $i > 1$  then
89              $T[i,j]_{sum} := T[i,j]_{sum} + Temp[i-1, j]$ ;
90           if  $j > 1$  then
91              $T[i,j]_{sum,w} := T[i,j]_{sum,w} + Temp[i-1, j-1]$ ;
92           end if;
93         end if;
94       end if;
95       if  $j > 1$  then
96          $T[i,j]_{sum} := T[i,j]_{sum} + Temp[i, j-1]$ ;
97         if  $i < rows$  then
98            $T[i,j]_{sum,w} := T[i,j]_{sum,w} + Temp[i+1, j-1]$ ;
99         end if;
100       end if;
101       if  $i < rows$  then
102          $T[i,j]_{sum} := T[i,j]_{sum} + Temp[i+1, j]$ ;
103         if  $j > 1$  then
104            $T[i,j]_{sum,w} := T[i,j]_{sum,w} + Temp[i+1, j-1]$ ;
105         end if;
106       end if;
107       if  $j < cols$  then
108          $T[i,j]_{sum} := T[i,j]_{sum} + Temp[i, j+1]$ ;
109         if  $i < rows$  then
110            $T[i,j]_{sum,w} := T[i,j]_{sum,w} + Temp[i+1, j+1]$ ;
111         end if;
112       end if;
113        $T[i,j] := (T[i,j]_{sum} + w * T[i,j]_{sum,w}) / (4 + 4 * w)$ ;
114        $Grad[i, j] := (T[i,j] - Temp[i, j]) / ((4 + 4 * w) * (1 + 2 * w) * (h * h))$ ;
115     end for;
116   end for;
117   upd_temp(xn1, y, Dend, Temp, Grad);
118 else
119   status(yn1, Dend, Temp, S, n1);
120 end if;
121 end if;
```

Рис. 6: Codeblock 6

```
125 function upd_temp
126     input Integer x;
127     input Integer y;
128     input Real Dend[rows, cols];
129     input Real Temp[rows, cols];
130     input Real Grad[rows, cols];
131     output Real Temp_out[rows, cols];
132     Real delta_T;
133     Real sum_not_diagonal;
134     Real sum_diagonal;
135
136 algorithm
137     if y < N then
138         if x < M then
139             delta_T := (deltaT * k) / (cp * l * Dend[x, y]);
140             sum_not_diagonal := Grad[x-1, y] + Grad[x, y-1] + Grad[x+1, y] + Grad[x, y+1];
141             sum_diagonal := Grad[x-1, y-1] + Grad[x-1, y+1] + Grad[x+1, y-1] + Grad[x+1, y+1];
142             Temp_out[x, y] := Temp[x, y] + delta_T * (sum_not_diagonal + w * sum_diagonal);
143         else
144             cal_grad_tem(1, y+1, Dend, Temp, Grad, S, nu);
145         end if;
146     end if;
147 end upd_temp;
```

Рис. 7: Codeblock 7

Код проекта Алгоритм роста дендритов

```
150 function status;
151   input Integer y;
152   input Real dend[rows, cols];
153   input Real temp[rows, cols];
154   output Real S_out[rows, cols];
155   output Real ru_out[rows, cols];
156   Real Tr;
157   Real delta_T;
158   Real u;
159   Real hu;
160   Real hu;
161   Real h_out;
162   Real sum_S;
163
164 algorithm
165   if y < n then
166     Tr := temp[M, y];
167     delta_T := Tr - Trref;
168     u := h1 * delta_T;
169     hu := u * L / m[1, y];
170     hu := 0.023 * power(hu, 0.8) * power(1 / L, 0.4);
171     h_out := hu * K / L;
172     S_out[M, y] := h_out * delta_T;
173     ru_out[1, y] := Modelica.Math.Random.normal(0, 1) * delta / sqrt(2 * deltaT);
174     for i in 1:n-1 loop
175       delta_T := temp[i, y] - Trref;
176       u := h1 * delta_T;
177       hu := u * L / m[i, y];
178       hu := 0.023 * power(hu, 0.8) * power(1 / L, 0.4);
179       h_out := hu * K / L;
180       S_out[i, y] := h_out * delta_T;
181       ru_out[i, y] := Modelica.Math.Random.normal(0, 1) * delta / sqrt(2 * deltaT);
182     end for;
183     sum_S := sum(S_out);
184   end if;
185 end status;
```

Рис. 8: Codeblock 8

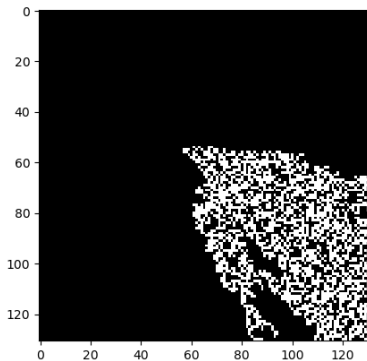


Рис. 9: График 1

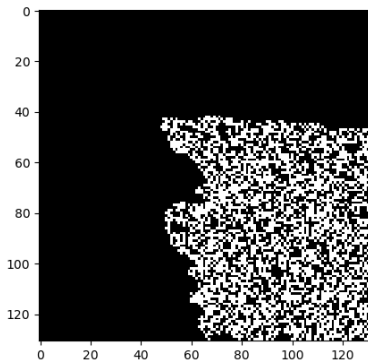


Рис. 10: График 2

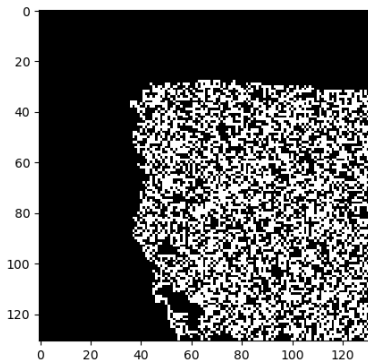


Рис. 11: График 3

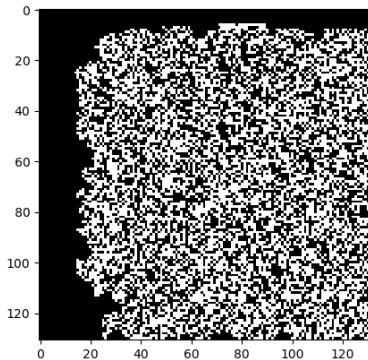


Рис. 12: График 4

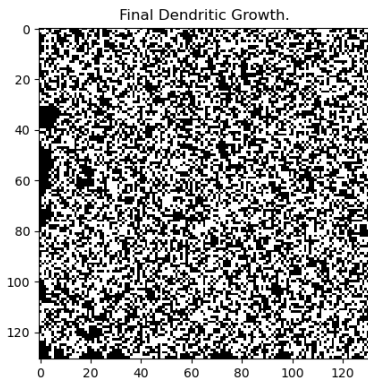


Рис. 13: График 5

Вывод

На данном этапе рассмотрели алгоритм по созданию и росту дендритов
Изучили каждый из его этапов и построили схему.