

温度  $T_b = 0\text{ }^{\circ}\text{C}$ 、一定の無限平板 (厚さ 20 cm) の片側表面を急に  $T_a = 1000\text{ }^{\circ}\text{C}$ にした場合の温度変化に関して、後退差分法のトーマス法による数値解析をソースコード 1 に示す。時間ステップは 250 s、解析結果を 1000 s ごとに 5000 s ごとに出力したものを図 1 に示す。

ソースコード 1: 後退差分法のトーマス法による数値解析プログラム

```

1  #include<stdio.h>
2
3  #define N 10 // メッシュ数
4  #define L 0.2 // 棒の長さ
5  #define dx (L / N) // メッシュの間隔
6  #define T0 1000 // 左端の温度
7  #define T1 0 // 右端の温度
8  #define dt 250 // 時間刻み幅
9
10 int T_final = 5000; // 解析終了時間
11 double alpha = 1.26/(1600*1050); // 熱伝達率
12 double T_old[N+1]; // 未知数の温度分布
13 double T_new[N+1]; // 次の時間ステップの温度分布
14 double a[N-1]; // トーマス法の行列の係数a~d
15 double b[N-1];
16 double c[N-1];
17 double d[N-1];
18 double r = (alpha*dt/(dx*dx)); //収束判定
19
20 void ThomasAlgorithm(){
21
22     // 時間ステップごとの反復
23     int t = 250;
24     while(t <= T_final){
25         //トーマス法の計算
26         for(int x = 1; x < N-1; x++){
27             double e = a[x]/b[x-1];
28             b[x] = b[x] - e * c[x-1];
29             d[x] = d[x] - e * d[x-1];
30         }
31         d[N-2] = d[N-2]/b[N-2];
32         for(int x = N - 3; x >= 0; x--){
33             d[x] = (d[x] - c[x] * d[x+1])/b[x];
34         }
35         for(int x = 1; x < N; x++){
36             T_new[x] = d[x-1];
37         }
38         //1000秒ごとに表示
39         if(t % 1000 == 0){
40             printf("%d[s]\n",t);
41             for(int x = 0; x < N+1; x++){
42                 printf("%5.1f\t",T_new[x]);
43             }
44             printf("\n");
45         }
46         for(int x = 0; x < N+1; x++){

```

```

47     T_old[x] = T_new[x];
48 }
49 for(int x = 0; x < N-1; x++){
50     a[x] = -r;
51     b[x] = 1 + 2 * r;
52     c[x] = -r;
53     if(x == 0){
54         d[x] = r * T_old[x] + T_old[x+1];
55     }else if(x == N - 2){
56         d[x] = r * T_old[x+2] + T_old[x+1];
57     }else{
58         d[x] = T_old[x+1];
59     }
60 }
61 t+=dt;
62 }
63 }
64
65 int main(){
66     //境界条件
67     for(int x = 1; x < N; x++){
68         T_old[x] = T1;
69         T_new[x] = T1; // 初期温度分布を 0°Cで初期化
70     }
71     T_old[0] = T0;
72     T_new[0] = T0;
73
74     for(int x = 0; x < N-1; x++){
75         a[x] = -r;
76         b[x] = 1 + 2 * r;
77         c[x] = -r;
78         if(x == 0){
79             d[x] = r * T_old[x] + T_old[x+1];
80         }else if(x == N - 2){
81             d[x] = r * T_old[x+2] + T_old[x+1];
82         }else{
83             d[x] = T_old[x+1];
84         }
85     }
86     ThomasAlgorithm();
87     return 0;
88 }

```

```

C:\Users\%s_takahashi\takahashi_workspace\c_workspace\computational_mechanics\kadi3>kadi3.exe
1000[s] 1000.0 571.2 280.6 123.8 50.5 19.4 7.1 2.5 0.9 0.3 0.0
2000[s] 1000.0 700.6 447.4 262.5 142.9 72.9 35.2 16.1 6.9 2.5 0.0
3000[s] 1000.0 757.5 538.9 360.3 227.0 135.3 76.5 41.0 20.4 8.3 0.0
4000[s] 1000.0 791.0 596.9 429.5 294.8 193.1 120.6 71.5 39.0 17.0 0.0
5000[s] 1000.0 813.5 637.6 480.8 348.7 242.9 162.1 102.5 59.3 26.9 0.0

```

図 1: 実行結果