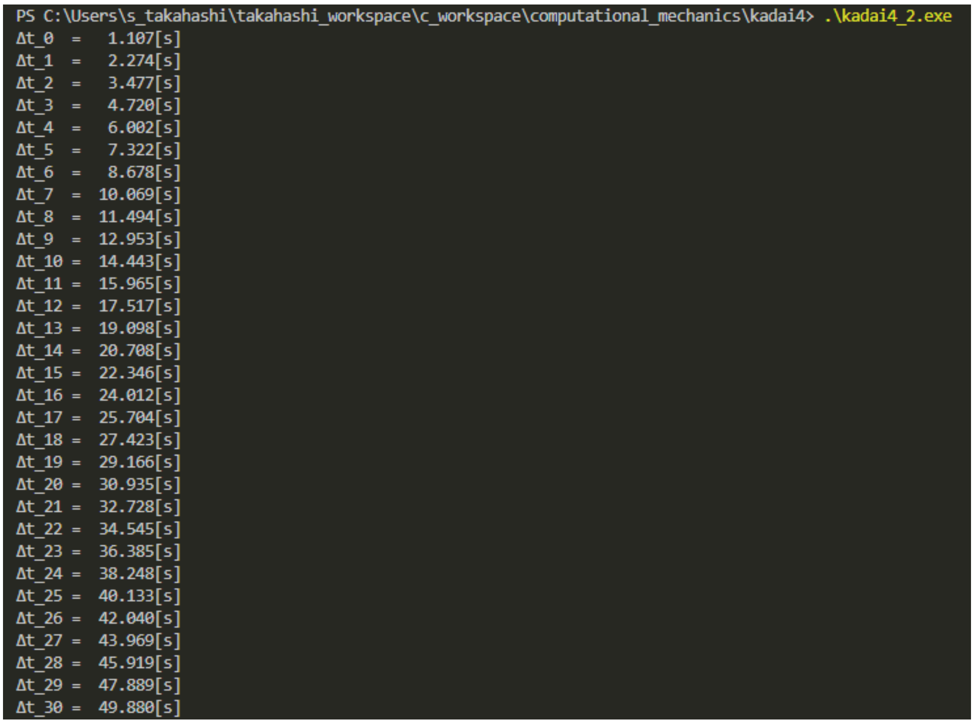
外径 50 [mm]、内径 10 [mm] の円筒内に満たされている水が全て凍結するまでの時間を後退差分法の VTS 法で求めるプログラムをソースコード 1 に示す。各時間ステップを出力したものを実行結果として、図 1〜4 に示す。

ソースコード 1: 円筒状物体が凍結するまでの時間を後退差分法の VTS 法で求めるプログラム

```
#include <stdio.h>
#include <math.h>

#define N 100
#define lambda_s 2.22
#define cp_s 2067
#define rho_s 917
#define alpha_s lambda_s / (rho_s * cp_s)
#define L 334880
#define Ta -5
#define Tm 0
#define A 0.005
#define B 0.025
#define eps 1e-5

double dt[N], old_dt[N];
double t_s[N][N] = {Tm};

const double dr = (B - A) / N; // dr の算出

void thomas_init(int n, double a[], double b[], double c[], double d[]){
  double r_s = (alpha_s * dt[n]) / (dr * dr);

  for(int i=0;i<n;i++){
    a[i] = -r_s * (1 - 1 / (2 * (A / dr + i + 1)));
    b[i] = 1 + 2 * r_s;
    c[i] = -r_s * (1 + 1 / (2 * (A / dr + i + 1)));
    d[i] = t_s[i+1][n];
  }
  d[0] = t_s[1][n] + r_s * (1 - 1 / (2 * (A / dr + 1))) * Ta;
  d[n-1] = 0;
}

void thomas(int n, double a[], double b[], double c[], double d[]){
  for(int i=1;i<n;i++){
    double e = a[i] / b[i-1];
    b[i] = b[i] - e * c[i-1];
    d[i] = d[i] - e * d[i-1];
  }
  d[n-1] = d[n-1] / b[n-1];
  for(int i=n-2;i>=0;i--){
    d[i] = (d[i] - c[i] * d[i+1]) / b[i];
  }

  for(int i=1;i<=n;i++){
```

```
    t_s[i][n+1] = d[i-1];
  }
}

int is_convergence(double old_val, double new_val) {
  // 収束判定
  double e_temp = fabs((new_val - old_val) / old_val);
  if(e_temp > eps){
    return 1;
  } else {
    return 0;
  }
}

int main(){
  double t_sum = 0;
  for(int i=0;i<N;i++) {
    dt[i] = 0;
    old_dt[i] = 0;
  }

  // dt_0
  t_s[0][1] = Ta;
  dt[0] = ((rho_s * L) / lambda_s) * ((dr * dr) / (Tm - Ta));
  printf("Δt_0  = %7.3lf[s]\n", dt[0]);
  t_sum += dt[0];

  // dt_1
  dt[1] = dt[0];
  do {
    old_dt[1] = dt[1];

    double r_s = (alpha_s * dt[1]) / (dr * dr);
    double q = 1 / (2 * (A / dr + 1)); // t_s1 計算の簡略化変数
    t_s[1][2] = 1 / (1 + 2 * r_s) * ((1 + r_s * (1 + q)) * Tm + r_s * Ta * (1 - q));
    dt[1] = ((rho_s * L) / lambda_s) * ((dr * dr) / (Tm - t_s[1][2]));
  } while(is_convergence(old_dt[1], dt[1]));
  printf("Δt_1  = %7.3lf[s]\n", dt[1]);
  t_sum += dt[1];

  // dt_2 ->
  for(int i=2;i<N;i++){
    dt[i] = dt[i-1];
    double a[i], b[i], c[i], d[i];

    do {
      old_dt[i] = dt[i];

      thomas_init(i, a, b, c, d);
      thomas(i, a, b, c, d);
      dt[i] = ((rho_s * L) / lambda_s) * ((dr * dr) / (Tm - t_s[i][i+1]));
```

```c
    } while(is_convergence(old_dt[i], dt[i]));

    printf("Δt_%-2d␣=␣%7.3lf[s]\n", i, dt[i]);
    t_sum += dt[i];
  }

  printf("\nt␣=␣%7.3lf[s]\n", t_sum);
  return 0;
}
```



```
PS C:\Users\s_takahashi\takahashi_workspace\c_workspace\computational_mechanics\kadai4> .\kadai4_2.exe
Δt_0  =   1.107[s]
Δt_1  =   2.274[s]
Δt_2  =   3.477[s]
Δt_3  =   4.720[s]
Δt_4  =   6.002[s]
Δt_5  =   7.322[s]
Δt_6  =   8.678[s]
Δt_7  =  10.069[s]
Δt_8  =  11.494[s]
Δt_9  =  12.953[s]
Δt_10 =  14.443[s]
Δt_11 =  15.965[s]
Δt_12 =  17.517[s]
Δt_13 =  19.098[s]
Δt_14 =  20.708[s]
Δt_15 =  22.346[s]
Δt_16 =  24.012[s]
Δt_17 =  25.704[s]
Δt_18 =  27.423[s]
Δt_19 =  29.166[s]
Δt_20 =  30.935[s]
Δt_21 =  32.728[s]
Δt_22 =  34.545[s]
Δt_23 =  36.385[s]
Δt_24 =  38.248[s]
Δt_25 =  40.133[s]
Δt_26 =  42.040[s]
Δt_27 =  43.969[s]
Δt_28 =  45.919[s]
Δt_29 =  47.889[s]
Δt_30 =  49.880[s]
```

図 1: 実行結果 1

```
Δt_31 =  51.891[s]
Δt_32 =  53.922[s]
Δt_33 =  55.971[s]
Δt_34 =  58.040[s]
Δt_35 =  60.127[s]
Δt_36 =  62.233[s]
Δt_37 =  64.356[s]
Δt_38 =  66.497[s]
Δt_39 =  68.656[s]
Δt_40 =  70.832[s]
Δt_41 =  73.025[s]
Δt_42 =  75.234[s]
Δt_43 =  77.460[s]
Δt_44 =  79.702[s]
Δt_45 =  81.960[s]
Δt_46 =  84.234[s]
Δt_47 =  86.523[s]
Δt_48 =  88.827[s]
Δt_49 =  91.147[s]
Δt_50 =  93.481[s]
Δt_51 =  95.830[s]
Δt_52 =  98.194[s]
Δt_53 = 100.571[s]
Δt_54 = 102.963[s]
Δt_55 = 105.369[s]
Δt_56 = 107.789[s]
Δt_57 = 110.222[s]
Δt_58 = 112.669[s]
Δt_59 = 115.128[s]
Δt_60 = 117.601[s]
Δt_61 = 120.087[s]
Δt_62 = 122.586[s]
Δt_63 = 125.097[s]
```

図 2: 実行結果 2

```
Δt_64 = 127.621[s]
Δt_65 = 130.158[s]
Δt_66 = 132.706[s]
Δt_67 = 135.267[s]
Δt_68 = 137.839[s]
Δt_69 = 140.424[s]
Δt_70 = 143.020[s]
Δt_71 = 145.628[s]
Δt_72 = 148.247[s]
Δt_73 = 150.878[s]
Δt_74 = 153.519[s]
Δt_75 = 156.172[s]
Δt_76 = 158.836[s]
Δt_77 = 161.511[s]
Δt_78 = 164.197[s]
Δt_79 = 166.893[s]
Δt_80 = 169.600[s]
Δt_81 = 172.318[s]
Δt_82 = 175.046[s]
Δt_83 = 177.784[s]
Δt_84 = 180.532[s]
Δt_85 = 183.291[s]
Δt_86 = 186.059[s]
Δt_87 = 188.838[s]
Δt_88 = 191.626[s]
Δt_89 = 194.424[s]
Δt_90 = 197.232[s]
Δt_91 = 200.049[s]
Δt_92 = 202.876[s]
Δt_93 = 205.713[s]
```

図 3: 実行結果 3

```
Δt_94 = 208.558[s]
Δt_95 = 211.413[s]
Δt_96 = 214.277[s]
Δt_97 = 217.150[s]
Δt_98 = 220.033[s]
Δt_99 = 222.924[s]

t = 9878.036[s]
```

図 4: 実行結果 4