渡辺　力先生　雪氷学実習ー Fortran 実習　08/01/2012
『拡散方程式を陰解法で解く』　76123042　箕輪　昌紘

1. 拡散方程式

微分方程式の一つである拡散（熱伝導）方程式

$$\frac{\partial y}{\partial t}=f(x,y)\frac{\partial^2 y}{\partial x^2} \quad \dots (1)$$

の数値積分の例として、拡散方程式

$$\frac{\partial u}{\partial t}=\frac{\partial^2 u}{\partial x^2} \quad \dots (2)$$

を任意の初期条件を与え積分して、関数 u(x, t)を求める。

2. 陰解法

(2)を離散化すると、

$$r\,T_{i-1}^{n+1}-(1+2\mathrm{r})\,T_i^{n+1}+r\,T_{i+1}^{n+1}=-T_i^n \quad \dots (3)$$

と表せる。また、(3)は形式的に

$$A_i T_{i-1}^{n+1}+B_i T_i^{n+1}+C_i T_{i+1}^{n+1}=T_i^n \quad \dots (4)$$

と書ける。ここで、

$$A_i=-r\,,B_i=1+2\mathrm{r}\,,C_i=-r \quad \dots (5)$$

であり、

$$A_1=0,C_{i-1}=0 \quad ...(6)$$

とした。i=1 のときは初期条件として定められる。i = 2 のとき、(4)において、 $A_2 T_1^{n+1}$ は定数項であるとみなせるので、

$$B'_2 T_i^{n+1}+C_2 T_3^{n+1}=T'^n_2 \quad \dots (7)$$

を得る。ここで、

$$B'_2=B_2,\quad T'_2=T_2^n-A_2 T_1^n \quad \dots (8)$$

を得る。この式と、(4)において、i = 3 とした式を用いて、 T_2^{n+1} を消去すると、

$$B'_3 T_3^{n+1}+C_3 T_4^{n+1}=T'_3 \quad \dots (9)$$

を得る。ここで、

$$B'_3 = B_3 - \frac{A_3 C_2}{B_2}, \quad T'_3 = T_3^n - \frac{A_3 T'^n_2}{B_2} \quad \dots (10)$$

である。同様の処理を繰り返すと、

$$B'_i T_i^{n+1} + C_i T_{i+1}^{n+1} = T'^n_i \quad \dots (11)$$

$$B'_i = B_i - \frac{A_i C_{i-1}}{B_{i-1}}, \quad T'_i = T_i^n - \frac{A_i T'^n_{i-1}}{B_{i-1}} \quad \dots (12)$$

を得る。これを i = m – 2 まで繰り返すと、

$$B'_{m-2} T_{m-2}^{n+1} + C_{m-2} T_{m-1}^{n+1} = T'^n_{m-2} \quad \dots (13)$$

となる。これを、(4)で i = m – 1 とした式

$$A_{m-1} T_{m-2}^{n+1} + B_{m-1} T_{m-1}^{n+1} = T_{m-1}^n \quad \dots (14)$$

と連立させて解けば、

$$B'_{m-1} T_{m-1}^{n+1} = T'^n_{m-1}, \therefore T_{m-1}^{n+1} = \frac{T'^n_{m-1}}{B'_{m-1}} \quad \dots (15)$$

を得る。このように求められた T_{m-1}^{n+1} と漸化式(11)をつかえば、以下の式により i = m – 2 から、i =2 までの T_i^{n+1} を求めることができる。

$$T_i^{n+1} = \frac{T'^n_i - C_i T_{i+1}^{n+1}}{B'_i} \quad \dots (16)$$

3. 解析結果

積分を行う時の時間間隔 $\Delta t = 0.55$ としたとき、陽解法では、解が不安定となることが、前回までの講義で示された(図 1)。

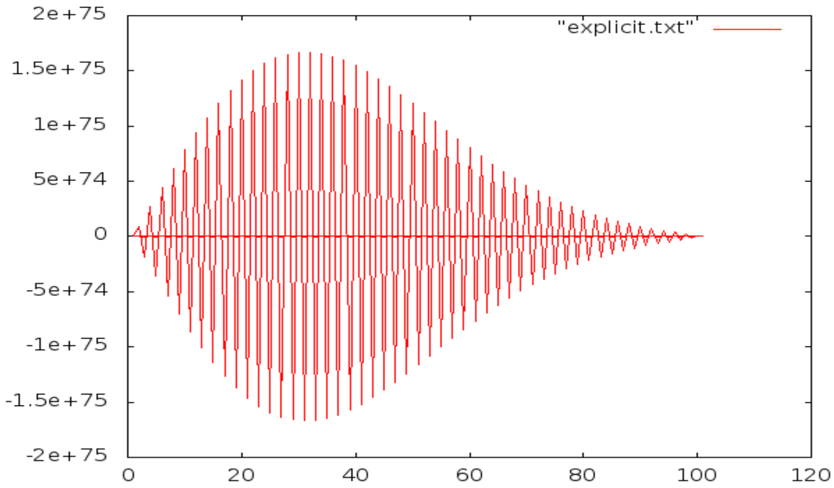

図 1 . 時間間隔を 0.55 とした時の陽解法による結果。

一方で同様の Δt を設定し陰解法で計算を行った場合、解は安定する(図 2)。

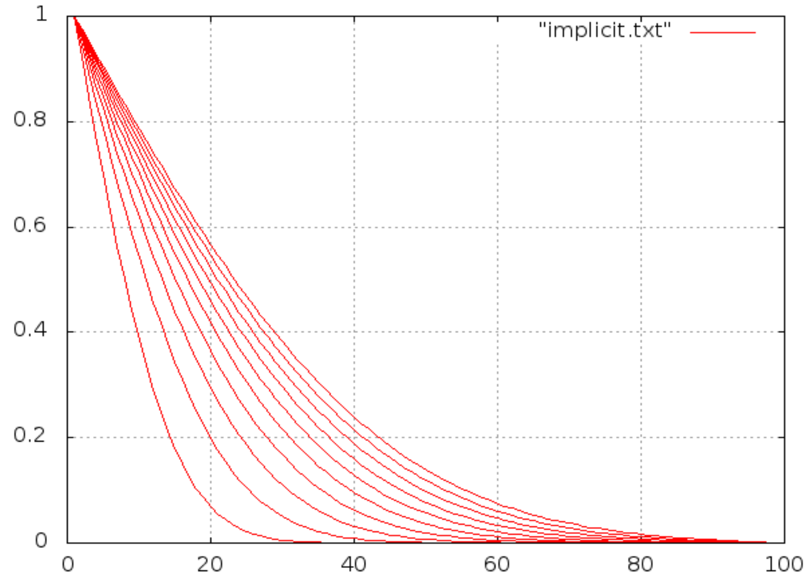

図 2. 時間間隔を 0.55 とした時の陰解法による結果。

次に、計算回数を増やした時、陽解法と陰解法でどのような違いが見られるかを考察する。ここで、Δt は 0.5 とし、5,000 回計算を行った。その結果を図 3 に示す。

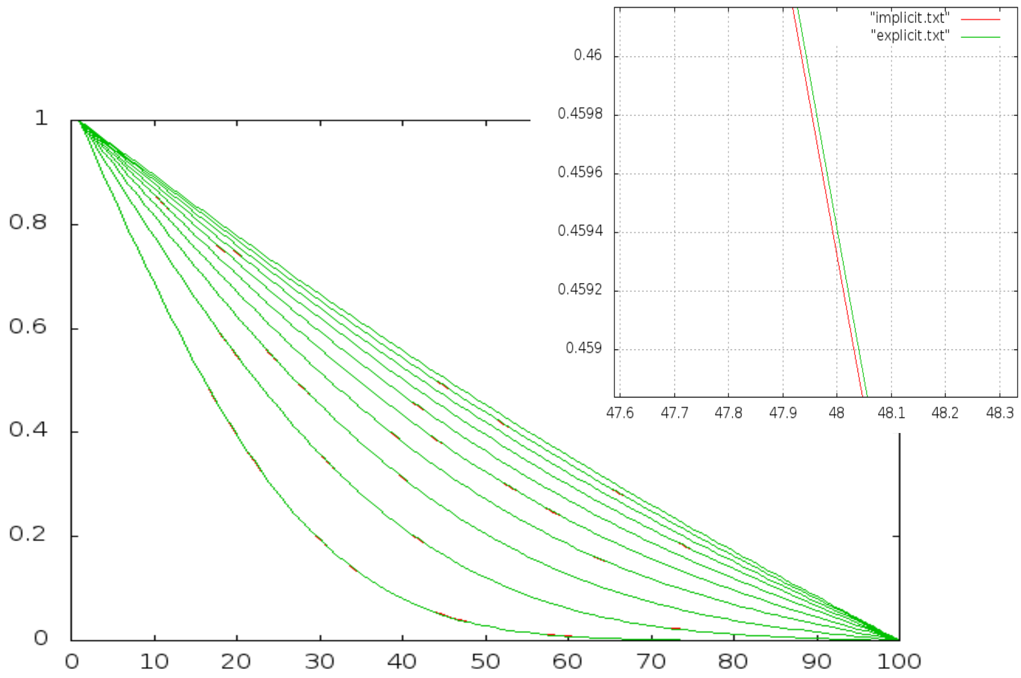

図 3. 時間間隔を 0.5 とし、5,000 回計算を行った時の陽解法(緑)、陰解法（赤）それぞれの結果。右上に解析結果の一部分の拡大図を示す。

陽解法による結果を緑色、陰解法による結果を赤色で示してある。この解析結果より、陽解法や陰解法どちらでも一定の回数を繰り返すと定常状態に達することがわかる。さらに、拡大して解析結果を見てみると、陽解法の方が陰解法よりもわずかに早く定常状態に達していることが見て取れる。

List.1 陰解法ソースコード

```
! --------------------------------
! Created by M. Minowa
! Lastupdate 2012/7/31
! --------------------------------
program netuKadai
        implicit none
        integer, parameter :: dp = kind(1.0d0)
        integer, parameter :: m = 100

        ! 変数の定義
        integer i, j, time
        real(dp) T(m), Tn(m)
        real(dp) dt, dx
        real(dp) T0, Tm, T1
        real(dp) k, h
        real(dp) r, f
        real(dp) a(m), b(m), c(m), bb(m)
        real(dp) d(m), dd(m)
        !
        k = 1.0d0-3    ! 熱伝導率
        h = 10      !
        T0 = 0.0d0          !
        T1 = 1.0d0
        T = T0
        dx = h / (m - 1)
        dt = 0.5d0 * dx * dx / k          !
        Tm = 1
        time = 5000
        ! 熱伝導係数
        r = k * dt / dx**2
 do i = 1, m
```

```
 ! 係数の初期条件
 if (i==1) then
  a(i) = 0
 else
   a(i) = -r
 end if
   b(i) = 1.0d0 + 2.0d0 * r
  if (i==m-1) then
   c(i) = 0
   !print *, "c : ", c(i)
  else
    c(i) = -r
    !print *, "c : ", c(i)
  end if
    dd(i) = 0.0d0
   d(i) = 0.0d0 ! 初期温度
  end do

        ! 時間微分の始まり
        do j = 1, time
                ! -------- 前進消去 -------------
                ! i = 2 のとき
            bb(2) = b(2)
            dd(2) = d(2) - a(2) * d(1)
            !print *, "dd(2) : ", dd(2)
            !
                do i = 3, m-1
                  f = a(i) / bb(i-1)
                    !print *, "f : ", f
                  bb(i) = b(i) - c(i-1) * f
                    !print *, "bb : ", bb(i)
                  dd(i) = d(i) - dd(i-1) * f
                        ! print *, "dd : ", dd(i)
                end do
                !
         Tn(m-1) = dd(m-1) / bb(m-1)
                 !print *, "Tn(m-1)", Tn(m-1)
         ! 初期条件と境界条件
         Tn(1) = T1
         Tn(m) = T0
         ! -------- 後退代入 ------------
        do i= m-2, 2, -1
            Tn(i) = (dd(i) - c(i) * Tn(i+1)) / bb(i)
        end do
        ! 温度を更新する
        do i = 1, m
             d(i) = Tn(i)
        end do
         if (mod(j,500)==0) then ! 出力の間引き
                do i = 1, m
                        print *, i, Tn(i) ! データの出力
                end do
                print *       ! 空行 データ表示のため
          end if
          end do
end program netuKadai
```

参考資料

1.「拡散方程式の数値積分[改訂版]」www.astr.tohoku.ac.jp/~chinone/pdf/Diffusion_equation.pdf
   2012/7/31 閲覧
2. 川崎晴久(1993), C & FORTRAN による数値解析の基礎, 共立出版株式会社