

# 数値流体力学サマースクール課題

富阪幸治

2000.7

## 1 波動方程式

テキスト p.6 の FTCS スキームは

$$\frac{\rho_j^{n+1} - \rho_j^n}{\Delta t} + c \frac{\rho_{j+1}^n - \rho_{j-1}^n}{2\Delta x} = 0 \quad (1)$$

のように差分して

$$\rho_j^{n+1} = \rho_j^n - c \frac{\Delta t}{2\Delta x} (\rho_{j+1}^n - \rho_{j-1}^n), \quad (2)$$

として解く方法である。

また、これは、数値流束を

$$E_{j+1/2} = c \frac{\rho_j + \rho_{j+1}}{2}, E_{j-1/2} = c \frac{\rho_{j-1} + \rho_j}{2}, \quad (3)$$

のようにとって、それを用いて

$$\rho_j^{n+1} = \rho_j^n - \frac{\Delta t}{\Delta x} (E_{j+1/2} - E_{j-1/2}), \quad (4)$$

のように時間推進を行なうと見ても良い。以下のプログラムは、数値流束を式(3)のように決めて、 $j = 1, \dots, 50$  に対して  $\rho = 1$ 、 $j = 51, \dots, 100$  に対して  $\rho = 0$  とクーラン数  $\nu \equiv c\Delta t/\Delta x = 0.25$  で 50 ステップ、100 ステップ計算した時の  $\rho$  を出力するものである。このプログラムを解析し、空間 1 次精度の風上差分のものに書き換えよ。さらに、FTCS スキーム、空間 1 次精度の風上差分の結果をグラフに出力せよ。

```
1
2 !
3 ! Test Program preparing for CFD Summer School
4 ! Q. Rewrite the program to use upwind scheme not FTCS.
5 !           July 2000, by K.T.
6 !
7     integer Nx
8     parameter (Nx=100)
9     real*8 rho(0:Nx),E(0:Nx),c,dtdx
10    integer j,k
11 !
```

```

12      c=1.0d0
13      dtdx=0.25d0
14      !
15      do j=0,50
16          rho(j)=1d0
17      end do
18      do j=51,Nx
19          rho(j)=0d0
20      end do
21      !
22      do j=0,Nx
23          write(*,'(i3,3x,f6.3,3x,f6.3)') j,rho(j)
24      end do
25      !
26      do k=1,100
27          do j=0,Nx-1
28              E(j)=0.5d0*c*(rho(j+1)+rho(j))
29          end do
30          E(Nx)=E(Nx-1)
31          do j=1,Nx
32              rho(j)=rho(j)-dtdx*(E(j)-E(j-1))
33          end do
34          rho(0)=rho(1)
35          if (mod(k,50).eq.0) then
36              write(*,*)
37              do j=0,Nx
38                  write(*,'(i3,3x,1e10.3,3x,f6.3)') j,rho(j)
39              end do
40          end if
41      end do
42      end

```

## 2 等温気体の流体力学

以下は、等温気体に対する1次精度の「RoeのFDS法」のプログラムの一部で、時間を $\Delta t$ だけ進めるサブルーチン副プログラムである。これを、MUSCL内挿を使うことによって空間2次精度のものに書き換えよ。p.22 図2-1と同じ計算を行ない得られた結果を図示せよ

```

1
2      subroutine FDS()
3      integer    Nx,Jm
4      parameter (Nx=100)    ! dim of variable

```

```

5     parameter (Jm=Nx)
6     real*8 rho,m,u
7     common /array/ rho(0:Nx), m(0:Nx), u(0:Nx)
8     real*8 cs,cs2,dtdx,dt,dx,t
9     integer iwrite,loopmax
10    common /var/cs,cs2,dtdx,dt,dx,t,iwrite,loopmax
11
12    ! Local Variable
13        real*8 rhop(0:Nx), up(0:Nx), E1(0:Nx), E2(0:Nx)
14        integer j,n1,n2,k
15    ! Matrix
16        real*8 Lambda(2),R(2,2),L(2,2),w(2,2)
17
18    ! Roe Average
19
20        do j=0, jm-1
21            rhop(j)=Sqrt(rho(j)*rho(j+1))
22            up(j)=(Sqrt(rho(j))*u(j)+Sqrt(rho(j+1))*u(j+1))
23            *      /(Sqrt(rho(j))+Sqrt(rho(j+1)))
24        end do
25        rhop(jm)=rhop(jm-1)
26        up(jm)=up(jm-1)
27
28        do j=0, jm-1
29            Lambda(1)=Abs(up(j)-cs)
30            Lambda(2)=Abs(up(j)+cs)
31
32            R(1,1)= 1d0
33            R(1,2)= 1d0
34            R(2,1)= up(j)-cs
35            R(2,2)= up(j)+cs
36
37            L(1,1)= 0.5d0*(up(j)/cs+1d0)
38            L(1,2)= -0.5/cs
39            L(2,1)= -0.5d0*(up(j)/cs-1d0)
40            L(2,2)= 0.5/cs
41
42        do n1=1, 2
43            do n2=1, 2
44                w(n1,n2)=0.0
45                do k=1, 2

```

```

46         w(n1,n2)=w(n1,n2)+R(n1,k)*Lambda(k)*L(k,n2)
47     end do
48 end do
49 end do
50 E1(j)=0.5*(m(j+1)+m(j)
51 *           -w(1,1)*(rho(j+1)-rho(j))
52 *           -w(1,2)*(m(j+1)-m(j)))
53 E2(j)=0.5*(m(j+1)**2/rho(j+1)+m(j)**2/rho(j)
54 *           +cs2*(rho(j+1)+rho(j))
55 *           -w(2,1)*(rho(j+1)-rho(j))
56 *           -w(2,2)*(m(j+1)-m(j)))
57 end do
58
59 do j=1, jm-1
60     rho(j)=rho(j)-dtdx*(E1(j)-E1(j-1))
61     m(j)=m(j)-dtdx*(E2(j)-E2(j-1))
62 end do
63 rho(0)=rho(1)
64 m(0)=m(1)
65 rho(jm)=rho(jm-1)
66 m(jm)=m(jm-1)
67 do j=0, jm
68     u(j)=m(j)/rho(j)
69 end do
70
71 end

```

### 3 時間 2 次精度化

前の節で空間 2 次化されて FDS 法のプログラムを、時間 2 次精度を保つように書き換えよ。これを p.27 の図 2-3 と比較せよ。