

## VI. 数値計算による偏微分方程式の解：双曲型方程式

### 15. 差分法による波動伝搬の数値シミュレーション

ここでは、1次元平面波の伝搬の式を数値的に解くことを考える。

#### 15.1 高階の時間微分の扱いと中点蛙跳び法

平面波の伝搬は、変位を  $u$  とすると波動方程式

$$\frac{\partial^2 u}{\partial t^2} = c^2 \frac{\partial^2 u}{\partial x^2} \quad (15.1)$$

で表される。差分法を用いて、この式を解くことを考える。

時間微分が2階以上の場合は、1階微分の複数の方程式に分ける。(15.1)は変位の速度  $v$  を使って、

$$\frac{\partial v}{\partial t} = c^2 \frac{\partial^2 u}{\partial x^2} \quad (15.2)$$

$$\frac{\partial u}{\partial t} = v \quad (15.3)$$

と書くことができる。オイラー法を(15.2)(15.3)に適用すると、

$$\frac{v_i^{n+1} - v_i^n}{\Delta t} = c^2 \frac{u_{i+1}^n + u_{i-1}^n - 2u_i^n}{\Delta x^2} \quad (15.4)$$

$$\frac{u_i^{n+1} - u_i^n}{\Delta t} = v_i^n \quad (15.5)$$

となるが、時間差分が1次誤差を持つ方法では正確に波を計算することができない。正確に計算するためには2次の方法が必要である。2次の1階差分は中心差分であるので、2つの式(15.2)(15.3)の時間を  $\Delta t/2$  ずらして右辺が2つの時間の真ん中になるように計算すればよい。すなわち、

$$\frac{v_i^{n+1/2} - v_i^{n-1/2}}{\Delta t} = c^2 \frac{u_{i+1}^n + u_{i-1}^n - 2u_i^n}{\Delta x^2} \quad (15.6)$$

$$\frac{u_i^{n+1} - u_i^n}{\Delta t} = v_i^{n+1/2} \quad (15.7)$$

のようになる。この方法では時間差分が中心差分となっていて2次の誤差を持つ。交互にずれた時間でとびとびに値を計算するので、この方法は**中点蛙跳び法** (leap-frog method)と呼ばれる。(15.6)に(15.7)を代入すると、

$$\frac{u_i^{n+1} + u_i^{n-1} - 2u_i^n}{\Delta t^2} = c^2 \frac{u_{i+1}^n + u_{i-1}^n - 2u_i^n}{\Delta x^2} \quad (15.8)$$

が得られる。これは(15.1)の2階の時間微分を2次の中心差分で置き換えたものと一致する。(15.8)を書き換えると、

$$u_i^{n+1} = 2u_i^n - u_i^{n-1} + \Delta t^2 c^2 \frac{u_{i+1}^n + u_{i-1}^n - 2u_i^n}{\Delta x^2} \quad (15.9)$$

となる。 $t=0$  と  $t=-\Delta t$  における2つの初期値が必要である。

## 15.2 波動の数値シミュレーション

(15.9)を用いると波動のシミュレーションを行うことが可能である。ここでは、次のような場合を考える。

- (1) 左端は自由境界, 右端は固定境界である
- (2) 初期条件はすべての点で変位0である。
- (3) 中心に波の発生源があり半周期分の正弦波を発生する。

これらの条件は以下のように与えればよい。以下の通りである。

### [1] 初期条件

2つの時間  $t = -\Delta t$  と  $t = 0$  での初期条件をすべての点に与える。

For  $i = 0, m+1$

$$u_0^n = 0 \quad (15.10)$$

$$u_m^{-1} = 0 \quad (15.11)$$

### [2] 境界条件

震源：動く時間内( $t < t_L$ )に対してのみ。

- ・変位震動源-中心点

$$u_{m/2+1}^n = U_0 \sin \omega t^n \quad (15.12)$$

境界：全ての時間ステップに対して与える。

- ・左端：固定端=ディリクレ型境界条件

この場合は温度一定の境界条件の時と同様に扱う。すなわち、

$$u_0^n = 0 \quad (15.13)$$

となる。

- ・右端：自由端(応力ゼロ)=ノイマン型境界条件

自由端の場合には熱流量0の境界条件と同じように扱えばよい。つまり、

$$\frac{\partial u}{\partial x} = 0 \quad (15.14)$$

より

$$u_m^n = u_{m+1}^n \quad (15.15)$$

である。応力が与えられる場合にも同様である。

ここで考えた境界条件のほかに、速度で与える境界条件も考えることができる。この場合、境界に対して、(15.5)を計算し変位を求めればよい。元々中点蛙跳び法なので、速度の時間ステップの位置がずれていることには注意する必要がある。

### 問題 VI-1

- (1) プログラムを入力して、コンパイルせよ。
- (2) プログラムを実行し、波の反射と衝突の様子を観察せよ。
- (3) 両方とも固定端になるようにプログラムを変更して実行せよ。波の衝突の様子を観察せよ。

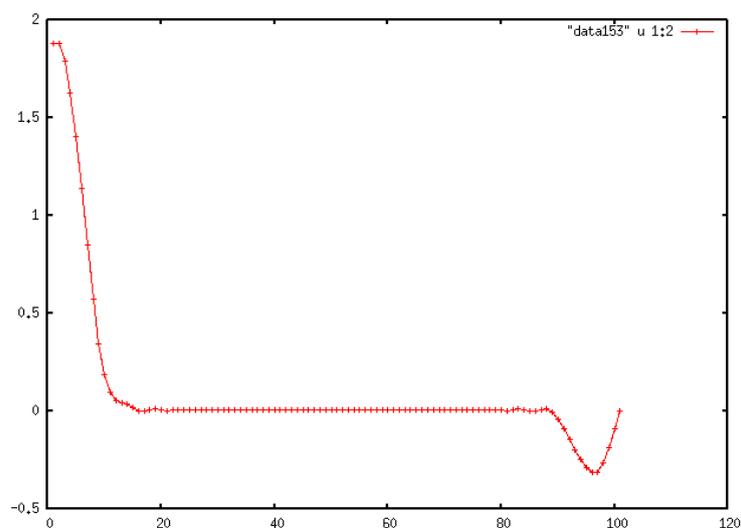


図 15.1 平面波の反射

問題の出展：Shearer: Introduction to seismology

## 問題 VI-1 波動伝播の数値シミュレーション

1次元の波動方程式を解くシンプルなプログラムである。

メインプログラム plane-wave.f90

```
! *****
! *
! * 1-D Plane wave propagation with fixed and free boundaries *
! * wave equation is solved by 2nd order (leap-frog) method *
! *
! *****

program planewave

!-----initialize t, dx, dt, tlen, beta and u1,u2,u3 arrays-----c

implicit none
integer:: i,counter,nx,nx1
integer:: it,ntmax
real(8):: t,dt,dt2,dx,dx2,beta,beta2,tlen,tmax,rhs
real(8),allocatable:: u1(:),u2(:),u3(:)
character(10):: data
character(3):: nf

!----- parameters -----c
! dt must be less than (dx/beta) for numerical stability

nx = 100 ! number of grid
dt = 0.10d0 ! time interval
dx = 1.0d0 ! grid interval
beta = 4.0d0 ! wave velocity
tlen = 5.0d0 ! time length of wave source
```

```
ntmax = 1000 ! maxmum time step (finish at it = ntmax, or)
tmax = 33.0d0 ! maximum calculation time (finish at t<tmax)

!----- allocate dimension variables -----c

nx1 = nx+1
allocate (u1(nx1),u2(nx1),u3(nx1))

!----- initialize t,counter, u1, u2, u3 -----c

t=0.0d0
counter=1

do i=1,nx1
  u1(i)=0.0d0
  u2(i)=0.0d0
  u3(i)=0.0d0
end do

!----- calculate c^2, dt^2, dx^2 -----c

beta2 = beta**2
dt2 = dt**2
dx2 = dx**2

! ===== Time marching loop =====

do it = 1,ntmax

  t=t+dt

!----- calculate u3(i) -----c
```

```
do i=2,nx
  rhs=beta2*(u2(i+1)-2.0d0*u2(i)+u2(i-1))/dx2
  u3(i)=dt2*rhs+2.0d0*u2(i)-u1(i)
end do

!---- free boundary (du/dx=0) -----c

u3(1)=u3(2)

!---- fixed boundary (u=0) -----c

u3(nx1)=0.0d0

!----- source time function c-----c

if (t.le.tlen) then
  u3(51)=sin(pi7*t/tlen)**2
end if

!---- change new and old variables -----c

do i=1,nx1
  u1(i)=u2(i)
  u2(i)=u3(i)
end do

!----make data file -----c

write(nf,'(i3.3)') counter
data = "data"//nf

open(7,file=data,status='replace')
do i= 1,nx1
```

```
        write(7,*) i,u2(i)
    end do

!-----output u2 at desired intervals, stop when t is big enough-----c

        write(6,'(a5,1x,i3,1x,f6.3)') 'loop', counter, t

        if (t.gt.tmax) exit

        counter = counter + 1

    end do

! ===== Loop end =====

    stop
end program planewave
```

このプログラムではパラメータは代入文で入力している。できればファイルから読み取るように修正した方がよい。

### 図の描き方

(1) gnuplot を使用する

```
gnuplot
```

```
gnuplot> plot "data" u 1:2 with linespoints
```

(2) Generic Mapping Tools (GMT)を使用する

```
psxy -R0/100/-2/2 -JX8/5 -W1.0 -Ba50f10/a0.5 -x2 -y15 data > data.ps
```

Postscript file の表示

```
gs data.ps
```