ソースファイル名：lbmthermMRT.for, 実行ファイル名：lbmthermMRT.out

```
$ vi    lbmthermMRT.for
$ gfortran    -o    lbmthermMRT.out    lbmthermMRT.for
$ ./ lbmthermMRT.out
```

```
      program HalfWayBounceBack

      real*8     g(5,100,100),    g0(5,100,100), gtmp(5,100,100)
      real*8    mg(5,100,100), mg0(5,100,100), mc(5,100,100)
      real*8    cx(5), cy(5), q(5,5), s(5,5), mm(5,5), ms(5,5)
      real*8    u(100,100), v(100,100), e(100,100), ea(100,100)
      real*8    tau, kap, pi, wav, u0, h, emax, emin, err, tmp
      integer i, j, k, m, ip, im, jp, jm, nx, ny, nf1, nf2, time
      complex*8 beta, img, ec(100,100)
      character a(100,100), b(100,100)

c.. discrete velocity
      cx(1) =    0.; cy(1) =    0.
      cx(2) =    1.; cy(2) =    0.; cx(3) =    0.; cy(3) =    1.
      cx(4) = -1.; cy(4) =    0.; cx(5) =    0.; cy(5) = -1.

c.. collision matrix (g)
      mm(1,1)= 1.0;mm(1,2)= 1.0;mm(1,3)= 1.0;mm(1,4)= 1.0;mm(1,5)= 1.0;
      mm(2,1)= 0.0;mm(2,2)= 1.0;mm(2,3)= 0.0;mm(2,4)=-1.0;mm(2,5)= 0.0;
      mm(3,1)= 0.0;mm(3,2)= 0.0;mm(3,3)= 1.0;mm(3,4)= 0.0;mm(3,5)=-1.0;
      mm(4,1)= 4.0;mm(4,2)=-1.0;mm(4,3)=-1.0;mm(4,4)=-1.0;mm(4,5)=-1.0;
      mm(5,1)= 0.0;mm(5,2)= 1.0;mm(5,3)=-1.0;mm(5,4)= 1.0;mm(5,5)=-1.0;

      q(1,1)= 0.2;q(1,2)= 0.0;q(1,3)= 0.0;q(1,4)= 0.20;q(1,5)= 0.00;
      q(2,1)= 0.2;q(2,2)= 0.5;q(2,3)= 0.0;q(2,4)=-0.05;q(2,5)= 0.25;
      q(3,1)= 0.2;q(3,2)= 0.0;q(3,3)= 0.5;q(3,4)=-0.05;q(3,5)=-0.25;
      q(4,1)= 0.2;q(4,2)=-0.5;q(4,3)= 0.0;q(4,4)=-0.05;q(4,5)= 0.25;
      q(5,1)= 0.2;q(5,2)= 0.0;q(5,3)=-0.5;q(5,4)=-0.05;q(5,5)=-0.25;

      tau = 0.503; kap = (tau - 0.5)/3;
      do k = 1,5; do m = 1,5; s(k,m)    = 0.; enddo; enddo;

      s(1,1) = 1.;
      s(2,2) = 1./tau;
      s(3,3) = 1./tau;
      s(4,4) = 1.;
      s(5,5) = 1.;

      do k = 1,5; do m = 1,5;
       ms(k,m) = 0.
       do n = 1,5; ms(k,m) = ms(k,m) + q(k,n)*s(n,m); enddo;
      enddo; enddo;

      time = 0; nx = 31; ny = nx + 1; h = float(ny - 2); u0 = 0.001
      img = (0.0, 1.0); pi = 4.*atan(1.); wav = 2.*pi/float(nx)
```

```
c.. exact solution
        beta = wav*sqrt(1. + img*u0/kap/wav)
        do i = 1, nx; do j = 1, ny
         ec(i,j)=exp(img*wav*float(i-1))/beta/h/(exp(beta*h)-exp(-beta*h))
     .    *( (1. + exp(-beta*h))*exp( beta*(float(j) - 1.5))
     .     + (1. + exp( beta*h))*exp(-beta*(float(j) - 1.5)) )
        end do; end do

        do i = 1, nx; do j = 1, ny
         ea(i,j) = real(ec(i,j)); e(i,j) = ea(i,j);
          u(i,j) = u0; v(i,j) = 0.;
        enddo; enddo

        do i = 1,nx; do j = 1,ny
         g(1,i,j) = e(i,j)/3.
         do k = 2,5
          tmp = cx(k)*u(i,j) + cy(k)*v(i,j)
          g(k,i,j) = e(i,j)*(1. + 3.*tmp)/6.
         end do;
        end do; end do

c.. calculation start
        do nf1 = 1, 20; do nf2 = 1,500
         time = time + 1
c..
         do i = 1,nx; do j = 1,ny
          g0(1,i,j) = e(i,j)/3.
          do k = 2,5
           tmp = cx(k)*u(i,j) + cy(k)*v(i,j)
           g0(k,i,j) = e(i,j)*(1. + 3.*tmp)/6.
          end do;
         end do; end do

c..   multi-relaxation time
         do i = 1, nx; do j = 1, ny;
          do k = 1, 5; mg(k,i,j) = 0.; mg0(k,i,j)= 0.
          do m = 1, 5;
           mg(k,i,j)   =   mg(k,i,j) + mm(k,m)* g(m,i,j)
           mg0(k,i,j) = mg0(k,i,j) + mm(k,m)*g0(m,i,j)
          enddo; enddo;
         enddo; enddo;

         do i = 1, nx; do j = 1, ny;
          do k = 1, 5; mc(k,i,j) = 0.
          do m = 1, 5;
           mc(k,i,j) = mc(k,i,j) + ms(k,m)*(mg(m,i,j) - mg0(m,i,j))
          enddo; enddo;
         enddo; enddo;
```

```
          do i = 1,nx; do j = 1,ny;
            do k = 1,5; g(k,i,j) = g(k,i,j) - mc(k,i,j); enddo;
          enddo; enddo

c..   streaming
          do i = 1,nx; do j = 1,ny;
            do k = 1,5; gtmp(k,i,j) = g(k,i,j); end do;
          end do; end do

          do i = 1,nx;
            ip = mod(i              , nx) + 1;
            im = mod(i + nx - 2, nx) + 1;
            do j = 1,ny       ; g(2,ip,j       ) = gtmp(2,i,j); end do
            do j = 1,ny - 1; g(3,i ,j + 1) = gtmp(3,i,j); end do
            do j = 1,ny       ; g(4,im,j       ) = gtmp(4,i,j); end do
            do j = 2,ny       ; g(5,i ,j - 1) = gtmp(5,i,j); end do
          end do;

c.. boudary condition
          do i = 1,nx
            ip = i + 1; if(i .eq. nx) ip = 1
            im = i - 1; if(i .eq. 1 ) im = nx
            g(3,i,2     ) = g(5,i,1 ) + cos(wav*float(i - 1))/h*kap;
            g(5,i,ny-1) = g(3,i,ny) + cos(wav*float(i - 1))/h*kap;
          end do;

c.. physics
          do i = 1,nx; do j = 1,ny
            e(i,j) = g(1,i,j)
            do k = 2,5
              e(i,j) =     e(i,j) + g(k,i,j)
            enddo
          end do; end do

c.. loop nf2
        end do

c.. error
        err = 0.; tmp = 0.
        do i = 1, nx; do j = 2, ny-1
          err = err + (e(i,j) - ea(i,j))**2;
          tmp = tmp + ea(i,j)**2;
        end do; end do
        err = sqrt(err/tmp)

c..   graphics
        do i = 1, nx; do j = 1, ny
          a(i,j)= '0'; b(i,j)= '0'
        end do; end do
```

```
      do j = 1, ny
       nb = int(u(nx/2,j) / umax * 20)
       do i = 1, nb
         a(i,j)= '-'
       end do
      end do
c..
      emax = -1000; emin = 1000;
      do i = 1, nx; do j = 1, ny;
       if(ea(i,j) .ge. emax) emax = ea(i,j)
       if(ea(i,j) .le. emin) emin = ea(i,j)
      enddo; enddo;

      do i = 1, nx; do j = 1, ny
       if(ea(i,j) .le. emax*1.0                    ) a(i,j)= '9'
       if(ea(i,j) .le. emax*0.9 + emin*0.1) a(i,j)= '8'
       if(ea(i,j) .le. emax*0.8 + emin*0.2) a(i,j)= '7'
       if(ea(i,j) .le. emax*0.7 + emin*0.3) a(i,j)= '6'
       if(ea(i,j) .le. emax*0.6 + emin*0.4) a(i,j)= '5'
       if(ea(i,j) .le. emax*0.5 + emin*0.5) a(i,j)= '4'
       if(ea(i,j) .le. emax*0.4 + emin*0.6) a(i,j)= '3'
       if(ea(i,j) .le. emax*0.3 + emin*0.7) a(i,j)= '2'
       if(ea(i,j) .le. emax*0.2 + emin*0.8) a(i,j)= '1'
       if(ea(i,j) .le. emax*0.1 + emin*0.9) a(i,j)= '0'
      end do; end do
c..
      emax = -1000; emin = 1000;
      do i = 1, nx; do j = 1, ny;
       if(e(i,j) .ge. emax) emax = e(i,j)
       if(e(i,j) .le. emin) emin = e(i,j)
      enddo; enddo;

      do i = 1, nx; do j = 1, ny
       if(e(i,j) .le. emax*1.0                    ) b(i,j)= '9'
       if(e(i,j) .le. emax*0.9 + emin*0.1) b(i,j)= '8'
       if(e(i,j) .le. emax*0.8 + emin*0.2) b(i,j)= '7'
       if(e(i,j) .le. emax*0.7 + emin*0.3) b(i,j)= '6'
       if(e(i,j) .le. emax*0.6 + emin*0.4) b(i,j)= '5'
       if(e(i,j) .le. emax*0.5 + emin*0.5) b(i,j)= '4'
       if(e(i,j) .le. emax*0.4 + emin*0.6) b(i,j)= '3'
       if(e(i,j) .le. emax*0.3 + emin*0.7) b(i,j)= '2'
       if(e(i,j) .le. emax*0.2 + emin*0.8) b(i,j)= '1'
       if(e(i,j) .le. emax*0.1 + emin*0.9) b(i,j)= '0'
      end do; end do

      write(*,*) '*--- Neumann    boundary condition ---*'
      write(*,'(a,i6,a,e17.10)')'time:',time,', err:',err
      write(*,'(a,i3,2f16.10)')'y =',2      ,ea(nx/2,      2),e(nx/2,      2)
      write(*,'(a,i3,2f16.10)')'y =',3      ,ea(nx/2,      3),e(nx/2,      3)
      write(*,'(a,i3,2f16.10)')'y =',ny-2,ea(nx/2,ny-2),e(nx/2,ny-2)
```

```
        write(*,'(a,i3,2f16.10)')'y =',ny-1,ea(nx/2,ny-1),e(nx/2,ny-1)
        do j = ny,1,-2
          write(*,*) (a(i,j),i = 1,nx,2),'    ',(b(i,j),i = 1,nx,2)
        end do

c.. loop nf1
        end do

99      write(*,*)'end'

        stop

        end
```

```
 *--- Neumann  boundary condition ---*
time: 10000, err: 0.2331248184E-01
y =  2    -0.0260352325    -0.0276134471
y =  3    -0.0064376481    -0.0073437965
y = 30    -0.0064376518    -0.0073437952
y = 31    -0.0260352138    -0.0276134457
 9998753100013578  8999764200002468
 6677765432222345  6777765432222345
 4556665554433344  4556666554333344
 4455555555444444  4455555555444444
 4444555555544444  4444555555544444
 4444455555555444  4444455555555444
 4444444555555554  4444444555555554
 5444444455555555  5444444455555555
 5444444445555555  5444444445555555
 4444444555555555  4444444555555555
 4444445555555544  4444445555555544
 4444555555554444  4444555555554444
 4445555555544444  4445555555544444
 4555555554444444  4455566554444344
 5566665543333344  5667766543323344
 7888764321112456  7898764310012357
 end
```