ソースファイル名：fdlbm.for，実行ファイル名：fdlbm.out

```
$ cp   ../common/ fdlbm_ori.for   .      ($ cp   ../common/ fdlbm_oripro.for   . )
$ cp   fdlbm_ori.for     fdlbm.for       ($ cp   fdlbm_oripro.for     fdlbm.for )
$ vi    fdlbm.for
$ gfortran   -o   fdlbm.out     fdlbm.for
$ ./ fdlbm.out
```

```
      program fdlbm

      real*8   cx(9), cy(9)
      real*8   f(9,5,35), fn(9,5,35), f0(9,5,35)
      real*8   ux(5,35), uy(5,35), rho(5,35)
      real*8   ux0, uy0, rho0, tau, g, ntime, mu
      real*8   dx , dy , dt
      integer nfnl, nx, ny
      character*8 answer*3

      ux0 = 0.0; uy0 = 0.0; rho0 = 1.0; tau = 0.6; g = 0.0005
      nfnl1 = 100; nfnl2 = 50;ntime = 0.
      dx = 1.0; dy = 1.0; dt = 0.5
      nx = 4; ny = 34
      mu = tau/3.

      call initial(cx,cy,f,f0,ux,uy,rho,nx,ny,ux0,uy0,rho0)

100   do n1 = 1, nfnl1
      do n2 = 1, nfnl2

       do k = 1,9; do i = 1,nx; do j = 1,ny
          fn(k,i,j) =
       end do; end do; end do

       ntime = ntime + 1

       call phys(cx,cy,f,ux,uy,rho,nx,ny)
       call boundary_mac(rho,ux,uy,nx,ny)
       call equiblium(                    )
       call collide(f,fn,f0,tau,nx,ny)
       call force(cx,cy,fn,rho,g,nx,ny)
       call stream(                    )
       call boundary_mic(cx,cy,fn,rho0,nx,ny)
```

```
      do k = 1,9; do i = 1,nx; do j = 1,ny
          f(k,i,j) = f(k,i,j) + fn(k,i,j)*dt
      end do; end do; end do

      call phys(cx,cy,f,ux,uy,rho,nx,ny)
    end do

    call plot(ux,ntime,dt,dy,mu,g,nx,ny)

  end do

  call exceldata(ux,dy,mu,g,nx,ny)

  end
c
  subroutine initial(cx,cy,f,f0,ux,uy,rho,nx,ny,ux0,uy0,rho0)
c-----------------------------------------------------------------------
  real*8   cx(9), cy(9), f(9,5,35), f0(9,5,35), w(9)
  real*8   ux(5,35), uy(5,35), rho(5,35)
  real*8   ux0, uy0, rho0, pi
  integer nx, ny

  pi = 4.0*atan(1.0)
  cx(1) =   0.0
  cy(1) =   0.0
  do k = 2,9
    w(k) = 1.
    if(mod(k, 2) .eq. 1.) w(k) = sqrt(2.)
    cx(k) = w(k)*cos((k-2)*pi/4.0)
    cy(k) = w(k)*sin((k-2)*pi/4.0)
  enddo

    do i = 1,nx
      do j = 1,ny
        ux(i,j) = ux0; uy(i,j) = uy0; rho(i,j) = rho0
      end do
  end do

  call equiblium(                    )
```

```
      do k = 1,9
       do i = 1,nx
        do j = 1,ny
            f(k,i,j) = f0(k,i,j)
        end do
       end do
      end do

      return
      end
c
      subroutine equiblium(                    )
c-----------------------------------------------------------------------
      real*8   cx(9), cy(9), f0(9,5,35)
      real*8   ux(5,35), uy(5,35), rho(5,35)
      real*8   u2, tmp, alp
      integer i, j, k, nx, ny

      do i = 1,nx
       do j = 1,ny
        u2 = ux(i,j)**2 + uy(i,j)**2
        f0(1,i,j) = rho(i,j)*(1. - 3./2.*u2)*4./9.
        do k = 1,4
         m = k*2       ; tmp = cx(m)*ux(i,j) + cy(m)*uy(i,j)
         f0(m,i,j) =
         m = k*2 + 1; tmp = cx(m)*ux(i,j) + cy(m)*uy(i,j)
         f0(m,i,j) = rho(i,j)*(1. + 3.*tmp + 9./2.*tmp**2 - 3./2.*u2)/36.
        end do
       end do
      end do

      return
      end
c
      subroutine boundary_mic(cx,cy,fn,rho0,nx,ny)
c-----------------------------------------------------------------------
      real*8   cx(9), cy(9), fn(9,5,35), ftmp(9,5,35), rho0
      real*8   uxtmp(5,35), uytmp(5,35), rhotmp(5,35)
      integer bflag, nx, ny
```

```
      bflag = 2

c..   bounce-back boudary
      if( bflag .eq. 1) then
        do i = 1,nx
          fn(3,i,1 ) =
          fn(4,i,1 ) =
          fn(5,i,1 ) =
          fn(7,i,ny) =
          fn(8,i,ny) =
          fn(9,i,ny) =
        end do
c..   extrapolation scheme ( 2nd order )
      else if( bflag .eq. 2) then
        do k = 1,9
          do i = 1,nx
            fn(k,i,1)    = 2.*fn(k,i,2)       - fn(k,i,3)
            fn(k,i,ny) =
          enddo
        enddo
      else if( bflag .eq. 3) then
c..   extrapolation scheme ( 1st order )
        do k = 1,9
          do i = 1,nx
            fn(k,i,1 ) = fn(k,i,2)
            fn(k,i,ny) = fn(k,i,ny-1)
          enddo
        enddo
      endif

      return
      end
c
      subroutine boundary_mac(rho,ux,uy,nx,ny)
c-----------------------------------------------------------------------
      real*8    ux(5,35), uy(5,35), rho(5,35)
      integer nx, ny

        do i = 1,nx
```

```
            ux(i,2     ) = 0.
            uy(i,2     ) = 0.
            ux(i,ny-1) = 0.
            uy(i,ny-1) = 0.
          enddo

        return
        end

        subroutine stream(                    )
c-------------------------------------------------------------------------
        real*8   f(9,5,35),fn(9,5,35),cx(9),cy(9)
        real*8   dx,dy,dt
        real*8   tmp1, tmp2
        integer i, j, k, ii, jj, sflag, nx, ny
c..     sflag
c..     centered difference : 1
c..     upwind difference     : 2

        sflag = 1

        if( sflag .eq. 1) then
c.. centered difference
          do k = 1,9
            do i = 1,nx
              do j = 1,ny
                i1 = i + 1; i2 = i - 1
                if(i .eq. 1 ) i2 = nx
                if(i .eq. nx) i1 = 1

                j1 = j + 1; j2 = j - 1
                if(j .eq. 1 ) j2 = ny
                if(j .eq. ny) j1 = 1

                fn(k,i,j) = fn(k,i,j)
     .                    -( cx(k)*(f(k,i1,j) - f(k,i2,j))/dx
     .                     + [                          ])*.5

              enddo
            enddo
```

```
          enddo
        else if(sflag .eq. 2) then
c.. upwind difference
          do k = 1,9
           do i = 1,nx
            do j = 1,ny
             if( cx(k). ge. 0.) then
              i1 = i; i2 = i - 1
              if(i .eq. 1 ) i2 = nx
              tmp1 = cx(k)*(f(k,i1,j) - f(k,i2,j))/dx
             else
              i1 = i + 1; i2 = i
              if(i .eq. nx) i1 = 1
              tmp1 = cx(k)*(f(k,i1,j) - f(k,i2,j))/dx
             endif
             if( cy(k). ge. 0.) then
              j1 = j; j2 = j - 1
              if(j .eq. 1) j2 = ny
             tmp2 = cy(k)*(f(k,i,j1) - f(k,i,j2))/dy
             else
              j1 = j + 1; j2 = j
              if(j .eq. ny) j1 = 1
              tmp2 = cy(k)*(f(k,i,j1) - f(k,i,j2))/dy
             endif
             fn(k,i,j) = fn(k,i,j) - (tmp1 + tmp2)
            enddo
           enddo
          enddo
        endif

        return
        end
c
        subroutine collide(f,fn,f0,tau,nx,ny)
c-----------------------------------------------------------------------
        real*8    f(9,5,35), fn(9,5,35), f0(9,5,35), tau
        integer i, j, k, nx, ny

        do k = 1,9
         do i = 1,nx
```

```
          do j = 1,ny
            fn(k,i,j) = fn(k,i,j) - (f(k,i,j) - [          ])/tau
          end do
        end do
      end do

      return
      end
c
      subroutine force(cx,cy,fn,rho,g,nx,ny)
c-----------------------------------------------------------------------
      real*8    cx(9), cy(9), fn(9,5,35), rho(5,35), g
      integer i, j, k, nx, ny

      do k = 1,9
        do i = 1,nx
          do j = 1,ny
            fn(k,i,j) = fn(k,i,j) + g*rho(i,j)*cx(k)/6.
          end do
        end do
      end do

      return
      end
c
      subroutine phys(cx,cy,f,ux,uy,rho,nx,ny)
c-----------------------------------------------------------------------
      real*8    cx(9), cy(9), f(9,5,35)
      real*8    ux(5,35), uy(5,35), rho(5,35)
      integer i, j, k, nx, ny

      do i = 1,nx
        do j = 1,ny
          ux(i,j) = 0; uy(i,j) = 0; rho(i,j) = f(1,i,j);
          do k = 2,9
            ux(i,j) = [                    ]
            uy(i,j) = [                    ]
           rho(i,j) = [                    ]
          end do
        end do
```

```
      end do

      do i = 1,nx
       do j = 1,ny
        if (rho(i,j) .ne. 0.) then
         ux(i,j) =   ux(i,j)/rho(i,j)
         uy(i,j) =   uy(i,j)/rho(i,j)
        else
         ux(i,j) = 0.
         uy(i,j) = 0.
        endif
        end do
      end do

      return
      end

      subroutine plot(ux,ntime,dt,dy,mu,g,nx,ny)
c-----------------------------------------------------------------------
      real*8 ux(5,35), ntime, dt, dy, mu, g, umax
      integer nx, ny

        uxmax = 0.
       do i = 1, nx
        do j = 1, ny
         if( ux(i,j) .ge. uxmax) uxmax = ux(i,j)
        end do
       end do

       write (*, *) 'Time    = ', ntime*dt,',     ',
     .                    'Error = ',
     . (g*(ny-3)**2*dy**2/8./mu - uxmax)/(g*(ny-3)**2*dy**2/8./mu)
c
       write (*,'(a30,f15.10)') 'U_max (Numerical    Solution)', uxmax
       write (*,'(a30,f15.10)') 'U_max (Analytical Solution)',
     . g*(ny-3)**2*dy**2/8./mu

2010    format(x, 34e15.6)
2020    format(x, 4e15.6)
```

```
      return
      end

      subroutine exceldata(ux,dy,mu,g,nx,ny)
c-----------------------------------------------------------------------
      real*8 ux(5,35), ntime, dy, mu, g
      integer nx, ny

        open(unit = 20, file = 'fdlbmvelocity', status= 'unknown')
         do j = 2, ny-1
         write(20,*) ux(2,j), '    ',
     1                -g/mu/2.*float(j -2)*float(j -(ny - 1))
         enddo
        close(20)

200   format(1x, 20e15.6)

      return
      end
```

《結果》

Time = 2500.00000000000 , Error = 4.950218500305833E-003

U_max (Numerical Solution) 0.2988258898

U_max (Analytical Solution) 0.3003125023