## Head file

```
      implicit double precision (a-h,o-z)
      parameter(im=161,jm=123,kk=8)
       common /parini/ ro0,U,Pr,Re,dt,pi,T0,Tw,tol
       common /grid/ x(im,jm),y(im,jm),a(im,jm,9,0:kk)
       common /physics/ ux(im,jm),uy(im,jm),tem(im,jm),
        $                        ro(im,jm),sm(im,jm),vor(im,jm)
       common /dist1/ f(im,jm,0:kk),feq(im,jm,0:kk),
        $                      ex(0:kk),ey(0:kk)
       common /dist2/ g(im,jm,0:kk),geq(im,jm,0:kk)
```

## Program for mesh generation

```
       include 'drivenhead.for'
       pi=4.0*atan(1.0)
       eta=0.8
       d=35.0
       do 1 j=1,jm
       dr=1.0+d*(1.-eta*atan((1.-real(j-2)/real(jm-3))*tan(1./eta)))
       do 1 i=1,im
       th=2.0*pi*real(i-1)/real(im-1)
       x(i,j)=dr*cos(th)
       y(i,j)=-dr*sin(th)
1      continue
       dmin=x(1,3)-x(1,2)
       dmax=x(1,jm)-x(1,jm-1)
       print *, 'ratio=',dmax/dmin
       open(2,file='mesh.dat')
       do i=1,im
       do j=1,jm
       write(2,*) x(i,j),y(i,j)
       end do
       end do
       close(2)
       end
```

## Main program

```
       include 'drivenhead.for'
       call parainitial
       call acoefficient
       print *, 'tol=', tol
       call initialrouandu
       call equilibrium
```

```
      call initialf
      N=0
      err=1.0e-8
100   call stream
      call density
      call velocity(umax)
      call temperature
      if(mod(N,25).eq.0)         print *, N,umax
      call equilibrium
      call collision
      if(mod(N,50).eq.0) then
            call nusselt
            call cpress
            call streamline
            call vorticity
            call printer
      endif
      N=N+1
      if(N.le.45000) goto 100
      call printer
      end
c
      subroutine parainitial
      include 'drivenhead.for'
      open(2,file='mesh.dat')
      do i=1,im
      do j=1,jm
      read(2,*) x(i,j),y(i,j)
      end do
      end do
      close(2)
      dt=min((x(1,2)-x(1,1)),(sqrt((x(1,1)-x(2,1))**2
     $    +(y(1,1)-y(2,1))**2)))
      pi=4.0*atan(1.0)
      visn=0.01
      Re=100.0
      U=0.15
      Pr=1.4*U/Re/visn
      ro0=1.0
      T0=298.15
      Tw=473.15
      tol=(12.0*U/(Re*dt)+1.0)/2.0
      ex(0)=0.0
      ey(0)=0.0
```

```
      ex(1)=1.0
      ey(1)=0.0
      ex(2)=1.0
      ey(2)=1.0
      ex(3)=0.0
      ey(3)=1.0
      ex(4)=-1.0
      ey(4)=1.0
      ex(5)=-1.0
      ey(5)=0.0
      ex(6)=-1.0
      ey(6)=-1.0
      ex(7)=0.0
      ey(7)=-1.0
      ex(8)=1.0
      ey(8)=-1.0
      end
c
      subroutine acoefficient
      include 'drivenhead.for'
      dimension aa(9),gxy(9,2)
      do 6 i=1,im
      do 6 j=2,jm-1
      if((i.eq.1).or.(i.eq.im)) then
            do 61 jj=j-1,j+1
            gxy(1+jj-j+1,1)=x(im-1,jj)
            gxy(1+jj-j+1,2)=y(im-1,jj)
            gxy(4+jj-j+1,1)=x(1,jj)
            gxy(4+jj-j+1,2)=y(1,jj)
            gxy(7+jj-j+1,1)=x(2,jj)
            gxy(7+jj-j+1,2)=y(2,jj)
61          continue
      else
            i0=0
            do 62 ii=i-1,i+1
            do 62 jj=j-1,j+1
            i0=i0+1
            gxy(i0,1)=x(ii,jj)
            gxy(i0,2)=y(ii,jj)
62          continue
      endif
      p1=gxy(5,1)
      p2=gxy(5,2)
      gxy(5,1)=gxy(1,1)
```

```
      gxy(5,2)=gxy(1,2)
      gxy(1,1)=p1
      gxy(1,2)=p2
      do 7 k=0,kk
      exx=ex(k)
      eyy=ey(k)
      call matrix(aa,exx,eyy,gxy)
      do 8 m=1,9
      a(i,j,m,k)=aa(m)
8     continue
7     continue
6     continue
      end
c
      subroutine matrix(aa,exx,eyy,gxy)
      include 'drivenhead.for'
      dimension aa(9),gxy(9,2)
      dimension m(9,6),n(6,9),s(6,6)
      real m,n
      do 1 i=1,9
      m(i,1)=1.0
      n(1,i)=1.0
      dx=gxy(i,1)+exx*dt-gxy(1,1)
      dy=gxy(i,2)+eyy*dt-gxy(1,2)
      m(i,2)=dx
      n(2,i)=dx
      m(i,3)=dy
      n(3,i)=dy
      m(i,4)=dx*dx/2.0
      n(4,i)=dx*dx/2.0
      m(i,5)=dy*dy/2.0
      n(5,i)=dy*dy/2.0
      m(i,6)=dx*dy
      n(6,i)=dx*dy
1     continue
      call multiply1(m,n)
      call convert(m,s)
      call multiply2(s,n,aa)
      end
c
      subroutine multiply1(m,n)
      include 'drivenhead.for'
      dimension m(9,6),n(6,9)
      real m,n
```

```
      dimension s(6,6)
      do 1 i=1,6
      do 1 j=1,6
      s(i,j)=0.0
      do 2 k=1,9
      s(i,j)=s(i,j)+n(i,k)*m(k,j)
2     continue
1     continue
      do 3 i=1,6
      do 3 j=1,6
      m(i,j)=s(i,j)
3     continue
      end
c
      subroutine convert(m,s)
      include 'drivenhead.for'
      dimension m(9,6),s(6,6)
      real m
      do 1 i=1,6
      do 1 j=1,6
      s(i,j)=m(i,j)
1     continue
      do 2 k=1,6
      do 3 j=1,6
      if(j.ne.k) then
            s(k,j)=s(k,j)/s(k,k)
      end if
3     continue
      s(k,k)=1.0/s(k,k)
      do 4 i=1,6
      if(i.ne.k) then
            do 5 j=1,6
            if(j.ne.k) then
                  s(i,j)=s(i,j)-s(k,j)*s(i,k)
            end if
5           continue
      end if
4     continue
      do 6 i=1,6
      if(i.ne.k) then
            s(i,k)=-s(i,k)*s(k,k)
      end if
6     continue
2     continue
```

```
      end
c
      subroutine multiply2(s,n,aa)
      include 'drivenhead.for'
      dimension s(6,6),n(6,9),aa(9)
      real n
      do 1 i=1,9
      aa(i)=0.0
      do 2 k=1,6
      aa(i)=aa(i)+s(1,k)*n(k,i)
2     continue
1     continue
      end
c
      subroutine initialrouandu
      include 'drivenhead.for'
      do i=1,im
      do j=2,jm-1
      if(j.eq.2) then
            ux(i,j)=0.0
            uy(i,j)=0.0
      else
            ux(i,j)=U
            uy(i,j)=0.0
      end if
      ro(i,j)=ro0
      tem(i,j)=T0
      if(j.eq.2) tem(i,j)=Tw
      end do
      end do
      end
c
      subroutine initialf
      include 'drivenhead.for'
      do i=1,im
      do j=2,jm-1
      do k=0,kk
      f(i,j,k)=feq(i,j,k)
      end do
      end do
      end do
      do i=1,im
      f(i,1,1)=f(i,3,5)
      f(i,1,2)=f(i,3,6)
```

```
      f(i,1,3)=f(i,3,7)
      f(i,1,4)=f(i,3,8)
      f(i,1,5)=f(i,3,1)
      f(i,1,6)=f(i,3,2)
      f(i,1,7)=f(i,3,3)
      f(i,1,8)=f(i,3,4)
      do k=1,kk
      f(i,jm,k)=f(i,jm-1,k)
      end do
      end  do
      end
c
      subroutine equilibrium
      include 'drivenhead.for'
      do i=1,im
      do j=2,jm-1
      uxy=ux(i,j)**2+uy(i,j)**2
      do k=0,kk
      eu=ex(k)*ux(i,j)+ey(k)*uy(i,j)
      if(k.eq.0) then
      w=4.0/9.0
      elseif(mod(k,2).eq.1) then
      w=1.0/9.0
      else
      w=1.0/36.0
      endif
      feq(i,j,k)=w*(ro(i,j)+ro0*(3.0*eu+4.5*eu**2-1.5*uxy))
      end do
      end do
      end do
      end
c
      subroutine stream
      include 'drivenhead.for'
      dimension f1(im,jm,0:kk)
      dimension gg1(9,9)
      do 1 i=1,im
      do 1 j=2,jm-1
      do 2 k=1,kk
      if((i.eq.1).or.(i.eq.im)) then
            do 21 jj=j-1,j+1
            gg1(1+jj-j+1,k)=f(im-1,jj,k)
            gg1(4+jj-j+1,k)=f(1,jj,k)
            gg1(7+jj-j+1,k)=f(2,jj,k)
```

```
21          continue
      else
            i0=0
            do 22 ii=i-1,i+1
            do 22 jj=j-1,j+1
            i0=i0+1
            gg1(i0,k)=f(ii,jj,k)
22          continue
      endif
      s=gg1(1,k)
      gg1(1,k)=gg1(5,k)
      gg1(5,k)=s
      f1(i,j,k)=0.0
      do 23 m=1,9
      f1(i,j,k)=f1(i,j,k)+a(i,j,m,k)*gg1(m,k)
23    continue
2     continue
1     continue
      do 3 i=1,im
      do 3 j=2,jm-1
      do 3 k=1,kk
      f(i,j,k)=f1(i,j,k)
3     continue
      end
c
      subroutine collision
      include 'drivenhead.for'
      do i=1,im
      do j=2,jm-1
      do k=0,kk
      f(i,j,k)=f(i,j,k)+(feq(i,j,k)-f(i,j,k))/tol
      end do
      end do
      end do

      do i=1,im
      ff1=f(i,2,1)
      ff2=f(i,2,2)
      ff3=f(i,2,3)
      ff4=f(i,2,4)
      f(i,2,1)=f(i,2,5)
      f(i,2,2)=f(i,2,6)
      f(i,2,3)=f(i,2,7)
      f(i,2,4)=f(i,2,8)
```

```
      f(i,2,5)=ff1
      f(i,2,6)=ff2
      f(i,2,7)=ff3
      f(i,2,8)=ff4

      f(i,1,1)=f(i,3,5)
      f(i,1,2)=f(i,3,6)
      f(i,1,3)=f(i,3,7)
      f(i,1,4)=f(i,3,8)
      f(i,1,5)=f(i,3,1)
      f(i,1,6)=f(i,3,2)
      f(i,1,7)=f(i,3,3)
      f(i,1,8)=f(i,3,4)
      do k=1,kk
      f(i,jm,k)=f(i,jm-1,k)
      end do
      end do
      end
c
      subroutine density
      include 'drivenhead.for'
      do i=1,im
      do j=2,jm-1
      ro(i,j)=0.0
      do k=0,kk
      ro(i,j)=ro(i,j)+f(i,j,k)
      end do
      if(j.eq.jm-1) ro(i,j)=ro0
      end do
      end do
      end
c
      subroutine velocity(umax)
      include 'drivenhead.for'
      umax=-1.0
      do i=1,im
      do j=3,jm-2
      uu=(f(i,j,1)+f(i,j,2)+f(i,j,8)-f(i,j,4)-f(i,j,5)-f(i,j,6))/ro0
      vv=(f(i,j,2)+f(i,j,3)+f(i,j,4)-f(i,j,6)-f(i,j,7)-f(i,j,8))/ro0
      ut=abs(sqrt(uu**2+vv**2)-sqrt(ux(i,j)**2+uy(i,j)**2))
      if(ut.gt.umax) umax=ut
      ux(i,j)=uu
      uy(i,j)=vv
      end do
```

```
      end do

      do i=1,im
      ux(i,2)=0.0
      uy(i,2)=0.0
      ux(i,jm-1)=U
      uy(i,jm-1)=0.0
      end do
      return
      end
c
      subroutine temperature
      include 'drivenhead.for'
      dimension th(im,jm)
      dtm=Tw-T0
      do i=1,im
      do j=2,jm-1
      th(i,j)=(tem(i,j)-T0)/dtm
      end do
      end do
      do i=1,im
      do j=4,jm-3
      if(i.eq.1.or.i.eq.im) then
            i1=im-1
            i2=2
            i3=im-2
            i4=3
      elseif(i.eq.2) then
            i1=i-1
            i2=i+1
            i3=im-1
            i4=4
      elseif(i.eq.im-1) then
            i1=i-1
            i2=i+1
            i3=im-3
            i4=2
      else
            i1=i-1
            i2=i+1
            i3=i-2
            i4=i+2
      endif
      j1=j-1
```

```
j2=j+1
j3=j-2
j4=j+2
xi=0.5*(x(i2,j)-x(i1,j))
yi=0.5*(y(i2,j)-y(i1,j))
xj=0.5*(x(i,j2)-x(i,j1))
yj=0.5*(y(i,j2)-y(i,j1))
xii=x(i1,j)-2.0*x(i,j)+x(i2,j)
yii=y(i1,j)-2.0*y(i,j)+y(i2,j)
xjj=x(i,j1)-2.0*x(i,j)+x(i,j2)
yjj=y(i,j1)-2.0*y(i,j)+y(i,j2)
xij=(x(i2,j2)-x(i2,j1)-x(i1,j2)+x(i1,j1))/4.0
yij=(y(i2,j2)-y(i2,j1)-y(i1,j2)+y(i1,j1))/4.0
al=xj**2+yj**2
b=xi*xj+yi*yj
c=xi**2+yi**2
yac=xi*yj-xj*yi
Um=ux(i,j)*yj-uy(i,j)*xj
Vm=uy(i,j)*xi-ux(i,j)*yi
ti=0.5*(th(i2,j)-th(i1,j))
tj=0.5*(th(i,j2)-th(i,j1))
tii=th(i1,j)-2.0*th(i,j)+th(i2,j)
tjj=th(i,j1)-2.0*th(i,j)+th(i,j2)
tij=(th(i2,j2)-th(i1,j2)-th(i2,j1)+th(i1,j1))/4.0
f1=Um/yac
f2=Vm/yac
if(f1.ge.0.0) then
ti1=(th(i4,j)-2.*th(i2,j)+9.*th(i,j)-10.*th(i1,j)+2.*th(i3,j))/6.
else
ti1=(-2.*th(i4,j)+10.*th(i2,j)-9.*th(i,j)+2.*th(i1,j)-th(i3,j))/6.
endif
if(f2.ge.0.0) then
tj1=(th(i,j4)-2.*th(i,j2)+9.*th(i,j)-10.*th(i,j1)+2.*th(i,j3))/6.
else
tj1=(-2.*th(i,j4)+10.*th(i,j2)-9.*th(i,j)+2.*th(i,j1)-th(i,j3))/6.
endif
suml=ti1*f1+tj1*f2
sumr=2.0*U*((al*tii-2.0*b*tij+c*tjj)/yac**2+((al*xii-2.0*b*xij+c*xjj)
  $    *(yi*tj-yj*ti)+(al*yii-2.0*b*yij+c*yjj)*(xj*ti-xi*tj))/yac**3)
  $    /(Re*Pr)
tm=(sumr-suml)*dt+th(i,j)
tem(i,j)=tm*dtm+T0
end do
end do
```

```
      do i=1,im
      tem(i,2)=Tw
      tem(i,jm-1)=T0
      tem(i,3)=(tem(i,2)+tem(i,4))/2.0
      tem(i,jm-2)=(tem(i,jm-1)+tem(i,jm-3))/2.0
      end do
      end
c
      subroutine nusselt
      include 'drivenhead.for'
      open(2,file='nuaround.dat')
      do 1 i=1,im
      th=360.0*real(i-1)/real(im-1)
      nu=-(tem(i,2)-tem(i,3))/(x(1,2)-x(1,3))
      write(2,*) th,nu
1     continue
      close(2)
      end
c
      subroutine cpress
      include 'drivenhead.for'
      open(2,file='cparound.dat')
      do 1 i=1,im
      th=360.0*real(i-1)/real(im-1)
      cp=2.0*(ro(i,3)-ro0)/(3.0*ro0*U**2)
      write(2,*) th,cp
1     continue
      close(2)
      end
c
      subroutine streamline
      include 'drivenhead.for'
      dimension ur(im,jm)
      do 1 i=1,im
      th=2.0*pi*real(i-1)/real(im-1)
      do 1 j=2,jm-1
      ur(i,j)=(-ux(i,j)*sin(th)-uy(i,j)*cos(th))/U
1     continue
      do 2 i=1,im
      sm(i,2)=0.0
      do 2 j=2,jm-2
      ds=x(1,j+1)-x(1,j)
      sm(i,j+1)=sm(i,j)+0.5*(ur(i,j)+ur(i,j+1))*ds
2     continue
```

```
      end
c
      subroutine vorticity
      include 'drivenhead.for'
         do 1 j=2,jm-1
         do 1 i=1,im
         i1=i-1
         i2=i+1
         if(i.eq.1) i1=im-1
         if(i.eq.im) i2=2
      xxi=0.5*(x(i2,j)-x(i1,j))
      yxi=0.5*(y(i2,j)-y(i1,j))
      xeta=0.5*(x(i,j+1)-x(i,j-1))
      yeta=0.5*(y(i,j+1)-y(i,j-1))
      yac=xxi*yeta-xeta*yxi
      vxi=0.5*(uy(i2,j)-uy(i1,j))
         uxi=0.5*(ux(i2,j)-ux(i1,j))
         ueta=0.5*(ux(i,j+1)-ux(i,j-1))
         veta=0.5*(uy(i,j+1)-uy(i,j-1))
         vor(i,j)=(-(vxi*yeta-veta*yxi)/yac-(uxi*xeta-ueta*xxi)/yac)/U
1        continue
      end
c
      subroutine printer
      include 'drivenhead.for'
      open(2,file='grid.dat')
      write(2,*) 'TITLE="Fluid flow around Cylinder with Temperature"'
      write(2,*) 'VARIABLES="X","Y","Ux","Uy","P","Stm","Vor","Tem"'
      write(2,*) 'ZONE I=',jm-2,', J=',im,', F=POINT'
      do 1 i=1,im
      do 1 j=2,jm-1
      uu=ux(i,j)/U
      vv=uy(i,j)/U
      P=ro(i,j)/3.0
      write(2,*) x(i,j),y(i,j),uu,vv,P,sm(i,j),vor(i,j),tem(i,j)
1     continue
      close(2)
      end
```