MODULE shock2com

PARAMETER (nx=40,ny=40,nskv=3)

real a(nskv)/nskv\*-10./, a1(nskv)

integer nxskv(nskv)/20,30,25/ ,nyskv(nskv)/20,15,20/,nxs/20/,nys/27/-зеленое не нужно

real dt,dx(nx),dy(ny)

integer nstep

DATA dx /nx\*10./ ,dy/ny\*10./

DATA dt /1/

data nstep/200/

END MODULE shock2com

$ debug

use shock2com

integer,parameter :: nrand=100

integer,parameter :: npar=5

real u(nrand,npar)

real kf,igrad,kmin,kmax,imin,imax,alfamin,alfamax,mmin,mmax,m

real c [allocatable] (:,:)

real cx [allocatable] (:,:)

real vx [allocatable] (:,:)

real cy [allocatable] (:,:)

real vy [allocatable] (:,:)

real c1 [allocatable] (:)

real v1 [allocatable] (:)

real c05 [allocatable] (:)

real crez [allocatable] (:,:)

real q [allocatable] (:,:)

integer\*2 ierr

integer\*4 step

mmin=5

mmax=10

pormin=.2

pormax=.4

imin=.001

imax=.01

alfamin=270

alfamax=330.

kmin=5

kmax=20

allocate (c(nx,ny),stat=ierr)

allocate (cx(nx,ny),stat=ierr)

allocate (cy(nx,ny),stat=ierr)

allocate (vx(nx,ny),stat=ierr)

allocate (vy(nx,ny),stat=ierr)

allocate (crez(nx,ny),stat=ierr)

allocate (q(nx,ny),stat=ierr)

crez=0.

q=0.

call rands\_1(nrand,npar,u)

do irand=1,nrand

igrad=imin+u(irand,1)\*(imax-imin)

alfa=alfamin+u(irand,2)\*(alfamax-alfamin)

m=mmin+u(irand,3)\*(mmax-mmin)

kf=kmin+u(irand,4)\*(kmax-kmin)

por=pormin+u(irand,5)\*(pormax-pormin)

do i=1,nskv

a1(i)=a(i)/(m\*por)

end do

do i=1,nx

do k=1,ny

vx(i,k)=-kf\*cosd(alfa)\*igrad/por

c(i,k)=0.

vy(i,k)=-kf\*sind(alfa)\*igrad/por

cx(i,k)=0.

cy(i,k)=0.

end do

end do

call VEL(ny,nx,dx,dy,VX,VY,A1,NXskv,NYskv,NSKV)

!c(nxs,nys)=100. Этот оператор не нужен

do nsk=1,nskv

q(NXskv(nsk),NYskv(nsk))=-a1(nsk)

end do

loop1 : do step=1,nstep

do nsk=1,nskv

c(NXskv(nsk),NYskv(nsk))=1

end do

allocate (c05(nx+1),stat=ierr)

allocate (v1(nx+1),stat=ierr)

allocate (c1(nx+1),stat=ierr)

loop2 : do k=2,ny-1

do i=1,nx

c1(i)=c(i,k)

v1(i)=vx(i,k)

end do

call shock1(c05,c1,v1,dx,dt,nx)

do i=1,nx

cx(i,k)=c05(i)

end do

end do loop2

deallocate (c05,v1,c1 )

allocate (c05(ny+1),stat=ierr)

allocate (v1(ny+1),stat=ierr)

allocate (c1(ny+1),stat=ierr)

loop3 : do i=2,nx-1

do k=1,ny

c1(k)=c(i,k)

v1(k)=vy(i,k)

end do

call shock1(c05,c1,v1,dy,dt,ny)

do k=1,ny

cy(i,k)=c05(k)

end do

end do loop3

deallocate (c05,v1,c1 )

do k=2,ny-2

do i=2,nx-2

c(i,k)=c(i,k)+dt/(dx(i)\*dy(k))\*(dy(k)\*(vx(i-1,k)\*cx(i-1,k)-vx(i,k)\*cx(i,k))+dx(i)\*(vy(i,k-1)\*cy(i,k-1)-vy(i,k)\*cy(i,k)+ Q(i,k)\*c(i,k) ))

end do

end do

do nsk=1,nskv

! c(NXskv(nsk),NYskv(nsk))=1

!if(nsk.eq.1)write(2,\*)step\*dt,c(NXskv(nsk),NYskv(nsk)),NXskv(nsk),NYskv(nsk)

!if(nsk.eq.2)write(3,\*)step\*dt,c(NXskv(nsk),NYskv(nsk)),NXskv(nsk),NYskv(nsk)

!if(nsk.eq.3)write(4,\*)step\*dt,c(NXskv(nsk),NYskv(nsk)),NXskv(nsk),NYskv(nsk)можно убрать.

end do

end do loop1

do i=1,nx

do k=1,ny

If(c(i,k).ge.0.5) crez(i,k)=crez(i,k)+1.

end do

end do

end do

! do k=1,ny

! write (1,'(40f7.3)') (c(i,k),i=1,nx)

! end do

do i=1,nx-1

do k=1,ny-1

! write (1,'(6g12.5)')i\*10.-5.,400-k\*10.+5., c(i,k),crez(i,k),vx(i,k),vy(i,k)

write (1,'(6g12.5)')i\*10.-5.,k\*10.-5., c(i,k),crez(i,k),vx(i,k),vy(i,k)

end do

end do

end

subroutine shock1(c05,c,v,dx,dt,nx)

real c (nx+1)

real c05(nx)

real dx(nx)

real v (nx)

real r (nx)

real f (nx)

integer sig

c05(1)=c(1)

nstr=nx

do j=1,nx

r(j)=1.

f(j)=0.

end do

iapr=2

! iapr=1

do j=2,nstr-1

sig=sign(1.,v(j))

if (iapr.gt.1) then

r1=(c(j+1-sig)-c(j-sig))

r2= (c(j+1)-c(j))

if(r2\*r1.le.0.)then

r(j)=0.

else

r(j)=r1/r2

end if

if(r(j).le.0.) then

f(j)=0.

else

if(r(j).le.1.) then

f(j)=amin1(2.\*r(j),1.)

else

f(j)=amin1(2.,r(j))

end if

end if

end if ! end iapr

c05(j)=.5\*(c(j)+c(j+1))-.5\*sig\*(c(j+1)-c(j))+ &

& .5\*f(j)\*(sig-dt\*v(j)\*2./(dx(j)+dx(j+1)))\*(c(j+1)-c(j))

end do

end