The following FORTRAN program describes a model of low voltage RF glow discharge between a pair of plates.

The basic **physical equations** used in the program are as follows:

$$\frac{\partial n_i}{\partial t}+\nabla\cdot\mathbf{J}_i=S$$

$$\frac{\partial n_e}{\partial t}+\nabla\cdot\mathbf{J}_e=S$$

$$\mathbf{J}_i=-D_i\nabla n_i+\mu_i n_i\mathbf{E}$$

$$\mathbf{J}_e=-D_e\nabla n_e-\mu_e n_e\mathbf{E}$$

$$\frac{\partial}{\partial t}\left(n_e\frac{3}{2}kT_e\right)+\nabla\cdot\mathbf{q}_e-p_c+p_l=0$$

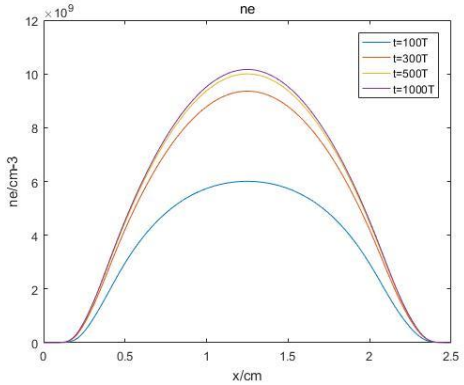
$$\mathbf{q}_e=-\frac{3}{2}kD_e n_e\nabla T_e+\frac{5}{2}kT_e\mathbf{J}_e$$

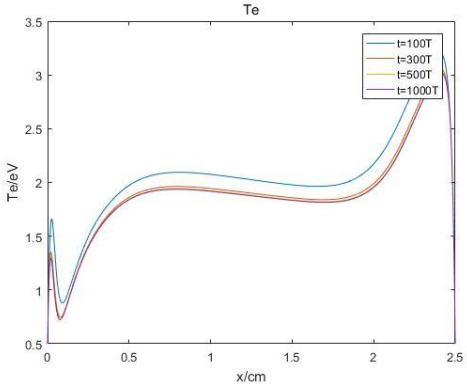
$$p_c=-e\mathbf{J}_e\cdot\mathbf{E}$$

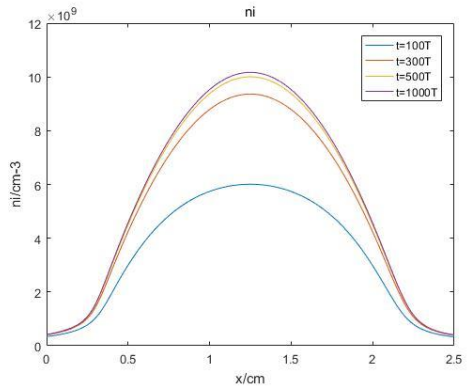
$$p_l=H_i S$$

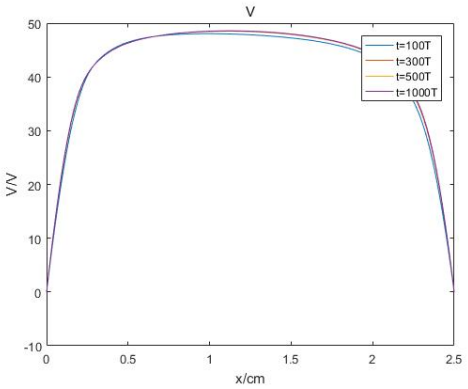
$$\nabla\cdot\mathbf{E}=\frac{e}{\varepsilon_0}\left(n_i-n_e\right),\ \ \mathbf{E}=-\nabla V$$

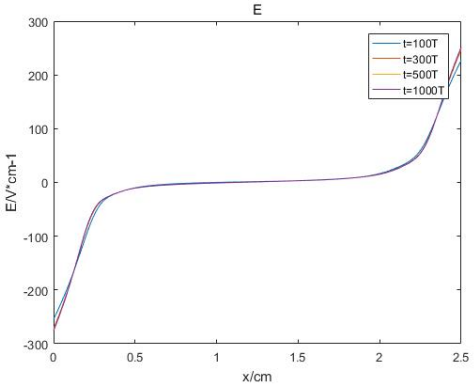
Program **running results**:

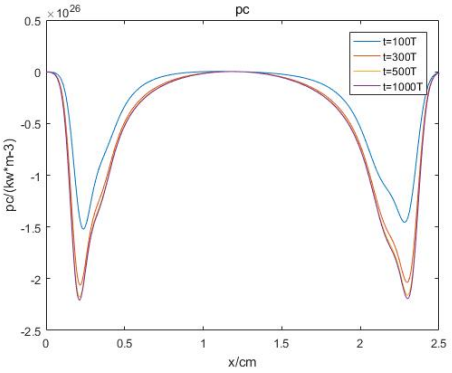


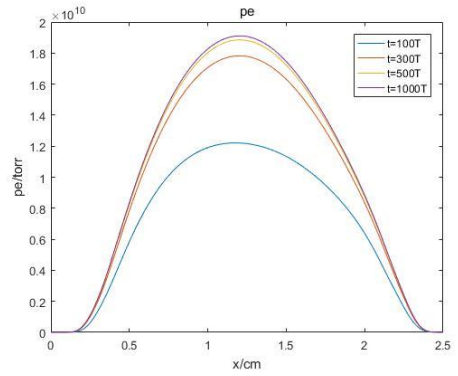



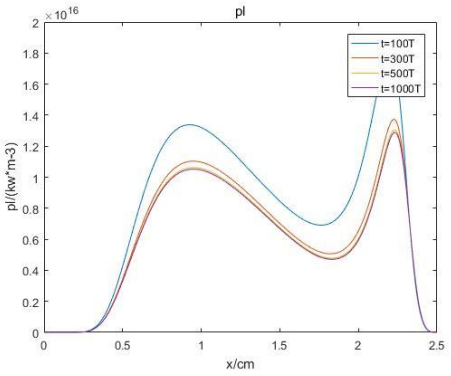

ni
×10^9
ni/cm-3
x/cm
t=100T
t=300T
t=500T
t=1000T

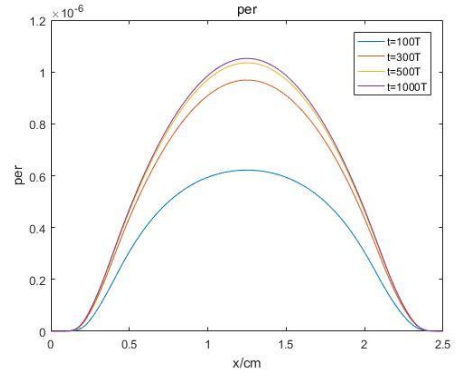

V
V/V
x/cm
t=100T
t=300T
t=500T
t=1000T

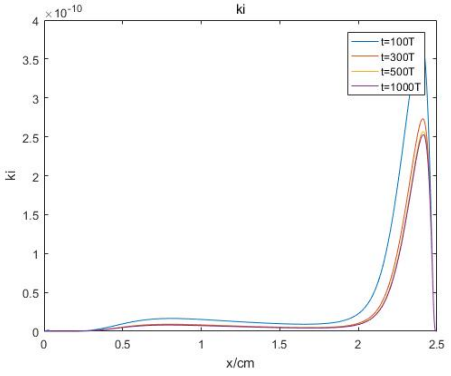

E
E/V*cm-1
x/cm
t=100T
t=300T
t=500T
t=1000T


pc
×10^26
pc/(kw*m-3)
x/cm
t=100T
t=300T
t=500T
t=1000T


pe
×10^10
pe/torr
x/cm
t=100T
t=300T
t=500T
t=1000T


pl
×10^16
pl/(kw*m-3)
x/cm
t=100T
t=300T
t=500T
t=1000T


per
×10^-6
per
x/cm
t=100T
t=300T
t=500T
t=1000T


ki
×10^-10
ki
x/cm
t=100T
t=300T
t=500T
t=1000T

```
program plasma
    implicit none
    !!=============================================================
    real*8,dimension(:,:),allocatable :: nih,neh,Teh,peh,Vh,Eh,Ehh
    real*8 :: Dih,Deh,muih,mueh,alpha,Hih,beta,nnh!constant
    real*8,dimension(:,:),allocatable :: Jeh,Jih,Sh,kih,pph,phh,pch,plh
    real*8 :: nepsh,Teih,ksh,Tebh,h!constant
    real*8 :: xh,th
    real*8,dimension(:,:),allocatable :: Te,ki
    real*8,dimension(:),allocatable :: a,b,c,f0,Vhtemp,pehtemp !tri
 !  real*8 :: time_begin , time_end
    !!nondimensionalization=======================================
    integer :: i,j,n,m,k
    real*8,parameter :: d=2.5,pn=0.3,Hi=15.7
    real*8,parameter :: e0=1.6022d-19,eps0=8.8542d-14
    real*8,parameter :: neps=1.d8,Tei=1.d0
    real*8,parameter :: gamma=0.01,ks=1.19d7,Teb=0.5,Va=100.0,f=13.56d6
    real*8,parameter :: ns=1.d11,Tes=1.d0
    real*8,parameter :: deltat=1.d-4

    real*8,parameter :: nn=9.66d15,Di=214.285714,De=3.99585921d6
    real*8,parameter :: mui=4813.6646,mue=1.d6
    real*8,parameter :: pi=3.14159265
    real*8 :: sf=0.01
    !! parameter=================================================
    h=1.d-3  !space step
    n=int(1/h)+1
    m=5000000!time

    write(*,*) 'm=',m,'n=',n
    allocate(nih(2,n))
    ! allocate(nihh(m/1000,1001))
    allocate(neh(2,n))
    ! allocate(nehh(m/1000,1001))
    allocate(Teh(2,n))
    ! allocate(Tehh(m/1000,1001))
    allocate(Te(2,n))
    allocate(peh(2,n))
    ! allocate(pehh(m/1000,1001))
    allocate(Vh(2,n))
    ! allocate(Vhh(m/1000,1001))
    allocate(Eh(2,n))
    allocate(Ehh(2,n))
    ! allocate(Ehhh(m/1000,1001))
    allocate(Jeh(2,n))
    allocate(Jih(2,n))
    allocate(Sh(2,n))
    allocate(kih(2,n))
    allocate(ki(2,n))
    allocate(pph(2,n))
    allocate(phh(2,n))
    allocate(pch(2,n))
    allocate(a(n))
    allocate(b(n))
    allocate(c(n))
    allocate(f0(n))
```

```
    allocate(Vhtemp(n-2))
    allocate(pehtemp(n-2))
    allocate(plh(2,n))!time*space
    Dih=Di/(f*d*d);Deh=De/(f*d*d)
    muih=mui*Va/(f*d*d);mueh=mue*Va/(f*d*d)
    alpha=100;Hih=15.7;beta=e0*ns*d*d/(eps0*Va)
    nnh=nn/ns;nepsh=neps/ns
    Teih=1;ksh=ks/(f*d);Tebh=0.5
  !!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
    do i=1,n
        Teh(1,i)=Teih
        xh=(i-1)*h
        neh(1,i)=16.*nepsh*xh*(1-xh)*xh*(1-xh)
        nih(1,i)=16.*nepsh*xh*(1-xh)*xh*(1-xh)
        !Teh(1,i)=Teih
        Vh(1,i)=0
        Te(1,i)=Teh(1,i)*Tes
        peh(1,i)=neh(1,i)*Teh(1,i)
        ki(1,i)=1.235d-7*exp(-18.687/Te(1,i))
        kih(1,i)=ki(1,i)*ns/f
        Sh(1,i)=kih(1,i)*nnh*neh(1,i)
    end do

    do i=2,n
       Ehh(1,i)=(Vh(1,i-1)-Vh(1,i))/h
    end do

        Eh(1,1)=-(-Vh(1,3)+4.*Vh(1,2)-3.*Vh(1,1))/(2.*h)
        Eh(1,n)=-(Vh(1,n-2)-4.*Vh(1,n-1)+3.*Vh(1,n))/(2.*h)
        pph(1,1)=alpha*Deh*Eh(1,1)*(-neh(1,3)+4.*neh(1,2)-3.*neh(1,1))/(2.*h)
        pph(1,n)=alpha*Deh*Eh(1,n)*(neh(1,n-2)-4.*neh(1,n-1)+3.*neh(1,n))/(2.*h)


   do i=2,n-1
       Eh(1,i)=(Ehh(1,i+1)+Ehh(1,i))/2.
       pph(1,i)=alpha*Deh*Eh(1,i)*(neh(1,i+1)-neh(1,i-1))/(2.*h)
   end do
     do i=1,n
        phh(1,i)=alpha*mueh*Eh(1,i)*Eh(1,i)*neh(1,i)
        pch(1,i)=pph(1,i)+phh(1,i)
        plh(1,i)=Hih*Sh(1,i)
   end do


    !!t=0                                                                    ↵
      nih,neh,Teh,Vh,peh,Eh,ki,kih,Sh,pph,pch,phh,plh=====================↵
      ==========
    a(1)=0;
    c(n)=0;
    do i=2,m
        !!=================================================================
        a(n)=-deltat/(h*h)*(2.*Dih+h*muih*Ehh(1,n))
        b(n)=1.+deltat/(h*h)*(2.*Dih+h*muih*Ehh(1,n))
        b(1)=1.+deltat/(h*h)*(2.*Dih-h*muih*Ehh(1,2))
        c(1)=-deltat/(h*h)*(2.*Dih-h*muih*Ehh(1,2))
        f0(1)=deltat*Sh(1,1)+nih(1,1)
```

```
        f0(n)=deltat*Sh(1,n)+nih(1,n)
        do  j=2,n-1
            a(j)=-deltat/(2.*h*h)*((2.*Dih+h*muih*Ehh(1,j)))
            b(j)=1.+deltat/(2.*h*h)*(4.*Dih+h*muih*(Ehh(1,j+1)-Ehh(1,j)))
            c(j)=-deltat/(2.*h*h)*((2.*Dih-h*muih*Ehh(1,j+1)))
            f0(j)=nih(1,j)+deltat*Sh(1,j)
        end do
        nih(2,:)=tri(a,b,c,f0,n)
        !!nih=============================================================
        a(n)=-(deltat/(h*h))*(2.*Deh-h*mueh*Ehh(1,n))
        b(n)=1.+(deltat/(h*h))*(2.*Deh+2.*h*ksh+h*mueh*Ehh(1,n))
        b(1)=1.+(deltat/(h*h))*(2.*Deh+2.*h*ksh-h*mueh*Ehh(1,2))
        c(1)=-(deltat/(h*h))*(2.*Deh+h*mueh*Ehh(1,2))
        f0(1)=deltat*(Sh(1,1)-2.*gamma*muih/h*nih(2,1)*Ehh(1,2))+neh(1,1)
        f0(n)=deltat*(Sh(1,n)+2.*gamma*muih/h*nih(2,n)*Ehh(1,n))+neh(1,n)
        do  j=2,n-1
            a(j)=-(deltat/(2.*h*h))*(2.*Deh-h*mueh*Ehh(1,j))
            b(j)=1.+(deltat/(2.*h*h))*(4.*Deh-h*mueh*(Ehh(1,j+1)-Ehh(1,j)))
            c(j)=-(deltat/(2.*h*h))*(2.*Deh+h*mueh*Ehh(1,j+1))
            f0(j)=neh(1,j)+deltat*Sh(1,j)
        end do
        neh(2,:)=tri(a,b,c,f0,n)
        !!neh=============================================================
        peh(2,1)=neh(2,1)*Tebh
        peh(2,n)=neh(2,n)*Tebh

        do  j=2,n-1
            a(j)=-(5.*deltat/(6.*h*h))*(2.*Deh-mueh*h*Ehh(1,j))
            b(j)=1.+(5.*deltat/(6.*h*h))*(4.*Deh-mueh*h*(Ehh(1,j+1)-Ehh(1,j)))
            c(j)=-(5.*deltat/(6.*h*h))*(2.*Deh+mueh*h*Ehh(1,j+1))
            f0(j)=peh(1,j)+(2.*deltat/3.)*(pch(1,j)-plh(1,j))-deltat*Deh/(3.*h*h)*((neh
              (2,j)+neh(2,j+1))*(Teh(1,j+1)-Teh(1,j))-(neh(2,j-1)+neh(2,j))*(Teh(1,j)-Teh
              (1,j-1)))
        end do
        f0(2)=f0(2)-a(2)*peh(2,1)
        f0(n-1)=f0(n-1)-c(n-1)*peh(2,n)
        a(2)=0;c(n-1)=0
        pehtemp=tri(a(2:n-1),b(2:n-1),c(2:n-1),f0(2:n-1),n-2)
        do  j=2,n-1
            peh(2,j)=pehtemp(j-1)
        end do
        !!peh=============================================================
        th=deltat*(i-1)
        Vh(2,1)=sin(2.*pi*th)
        Vh(2,n)=0

        do  j=2,n-1
            a(j)=-1.
            b(j)=2.
            c(j)=-1.
            f0(j)=beta*h*h*(nih(2,j)-neh(2,j))
        end do
        f0(2)=f0(2)-a(2)*Vh(2,1)
        a(2)=0;c(n-1)=0
        Vhtemp=tri(a(2:n-1),b(2:n-1),c(2:n-1),f0(2:n-1),n-2)
        do  j=2,n-1
```

```
            Vh(2,j)=Vhtemp(j-1)
          end do

  !!Vh=================================================================

   !      call smooth(n,sf,neh,nih,peh,vh)
          do j=2,n
              Ehh(2,j)=(Vh(2,j-1)-Vh(2,j))/h
          end do

            Eh(2,1)=-(-Vh(2,3)+4.*Vh(2,2)-3.*Vh(2,1))/(2.*h)
            Eh(2,n)=-(Vh(2,n-2)-4.*Vh(2,n-1)+3.*Vh(2,n))/(2.*h)
            pph(2,1)=alpha*Deh*Eh(2,1)*(-neh(2,3)+4.*neh(2,2)-3.*neh(2,1))/(2.*h)
            pph(2,n)=alpha*Deh*Eh(2,n)*(neh(2,n-2)-4.*neh(2,n-1)+3.*neh(2,n))/(2.*h)


          do j=2,n-1
              Eh(2,j)=(Ehh(2,j+1)+Ehh(2,j))/2.
              pph(2,j)=alpha*Deh*Eh(2,j)*(neh(2,j+1)-neh(2,j-1))/(2.*h)
          end do

          do j=1,n
              Teh(2,j)=peh(2,j)/neh(2,j)
              Te(2,j)=Teh(2,j)*Tes
              ki(2,j)=1.235d-7*exp(-18.687/Te(2,j))!reverse
              kih(2,j)=ki(2,j)*ns/f
              Sh(2,j)=kih(2,j)*nnh*neh(2,j)
              phh(2,j)=alpha*mueh*Eh(2,j)*Eh(2,j)*neh(2,j)
              pch(2,j)=pph(2,j)+phh(2,j)
              plh(2,j)=Hih*Sh(2,j)
          end do

          if(i==1000000) then

       open(6,file='ni100.dat',status='new')
       open(7,file='ne100.dat',status='new')
       open(8,file='pe100.dat',status='new')
       open(9,file='V100.dat',status='new')
       open(10,file='Te100.dat',status='new')
       open(11,file='E100.dat',status='new')
          do j=1,1001
              write(6,*) nih(2,j)
              write(7,*) neh(2,j)
              write(8,*) peh(2,j)
              write(9,*) Vh(2,j)
              write(10,*) Te(2,j)
             write(11,*) Eh(2,j)
          end do
       close(6)
       close(7)
      close(8)
      close(9)
       close(10)
       close(11)

        else  if(i==3000000) then
```

```
    open(12,file='ni300.dat',status='new')
    open(13,file='ne300.dat',status='new')
    open(14,file='pe300.dat',status='new')
    open(15,file='V300.dat',status='new')
    open(16,file='Te300.dat',status='new')
    open(17,file='E300.dat',status='new')
        do j=1,1001
            write(12,*) nih(2,j)
            write(13,*) neh(2,j)
            write(14,*) peh(2,j)
           write(15,*) Vh(2,j)
            write(16,*) Te(2,j)
           write(17,*) Eh(2,j)
        end do
    close(12)
    close(13)
   close(14)
   close(15)
    close(16)
    close(17)

 !    else if(i==1500000) then

   ! open(18,file='ni150.dat',status='new')
    !open(19,file='ne150.dat',status='new')
   ! open(20,file='pe150.dat',status='new')
    !open(21,file='V150.dat',status='new')
   ! open(22,file='Te150.dat',status='new')
  !  open(23,file='E150.dat',status='new')
 !       do j=1,1001
!            write(18,*) nih(2,j)
        !    write(19,*) neh(2,j)
       !     write(20,*) peh(2,j)
      !      write(21,*) Vh(2,j)
     !       write(22,*) Te(2,j)
    !       write(23,*) Eh(2,j)
   !     end do
   ! close(18)
  !  close(19)
  ! close(20)
  ! close(21)
  !  close(22)
  !  close(23)

        end if
        !!renew parameter=====================================================
        nih(1,:)=nih(2,:)
        neh(1,:)=neh(2,:)
        Teh(1,:)=Teh(2,:)
        Te(1,:)=Te(2,:)
        peh(1,:)=peh(2,:)
        Vh(1,:)=Vh(2,:)
        Eh(1,:)=Eh(2,:)
        Ehh(1,:)=Ehh(2,:)
        Jeh(1,:)=Jeh(2,:)
```

```
        Jih(1,:)=Jih(2,:)
        Sh(1,:)=Sh(2,:)
        kih(1,:)=kih(2,:)
        ki(1,:)=ki(2,:)
        pph(1,:)=pph(2,:)
        pch(1,:)=pch(2,:)
        plh(1,:)=plh(2,:)
        !!===========
    end do
    !!
      t=============================================================================================
      ==

        do j=1,n
            nih(1,j)=nih(1,j)*ns
            neh(1,j)=neh(1,j)*ns
            Vh(1,j)=Vh(1,j)*Va
            peh(1,j)=peh(1,j)*ns*Tes
            Eh(1,j)=Eh(1,j)*Va/d
        end do



    !!reverse=====================================================
    open(24,file='ni500.dat',status='new')
    open(25,file='ne500.dat',status='new')
    open(26,file='pe500.dat',status='new')
    open(27,file='V500.dat',status='new')
    open(28,file='Te500.dat',status='new')
    open(29,file='E500.dat',status='new')
!    do i=1,m/1000
        do j=1,1001
            write(24,*) nih(1,j)
            write(25,*) neh(1,j)
            write(26,*) peh(1,j)
            write(27,*) Vh(1,j)
            write(28,*) Te(1,j)
            write(29,*) Eh(1,j)
        end do
 !   end do!prevent overflow
    close(24)
    close(25)
    close(26)
    close(27)
    close(28)
    close(29)
    !!output ni ne Te V=========================================================
 !   call cpu_time(time_end)
  !      write(*,*) time_begin-time_end
        !!end time counting

    contains
     function tri(a,b,c,f,n)
       integer :: n,i,j
       real*8,dimension(n) :: a,b,c,f,tri,e,d
     e(1)=c(1)/b(1);d(1)=f(1)/b(1)
```

```
    do i=2,n
          e(i)=c(i)/(b(i)-a(i)*e(i-1))
           d(i)=(f(i)-a(i)*d(i-1))/(b(i)-a(i)*e(i-1))
        end do
        tri(n)=d(n)
       do i=n-1,1,-1
           tri(i)=d(i)-e(i)*tri(i+1)
       end do
    end function tri
        !!function tri=============================================================
    !       function tri(a,b,c,f,n)
    !    integer :: n,i,j
   !     real*8,dimension(n) :: a,b,c,f,tri,e,d
  !       e(1)=b(1)
  !       d(1)=f(1)
 !    do i=2,n
 !     e(i)=b(i)-c(i-1)*a(i)/(e(i-1)+1d-10)
 !         d(i)=f(i)-d(i-1)*a(i)/(e(i-1)+1d-10)
  !      end do
 !       tri(n)=d(n)/e(n)
  !      do j=n-1,1,-1
  !          tri(j)=(d(j)-c(j)*tri(j+1))/(e(j)+1d-10)
  !      end do
  !      end function tri

        !!========SMOOTH=========
        subroutine smooth(n,sf,ne,ni,p,v)
        integer :: n,i
        real*8,dimension(2,n) :: ne,ni,p,v
        real*8,dimension(n) ::y1,y2,y3,y4
        real*8 :: sf
        do i=2,n-1
            y1(i)=ne(2,i)+sf*(ne(2,i-1)-2.*ne(2,i)+ne(2,i+1))
            y2(i)=ni(2,i)+sf*(ni(2,i-1)-2.*ni(2,i)+ni(2,i+1))
            y3(i)=p(2,i)+sf*(p(2,i-1)-2.*p(2,i)+p(2,i+1))
            y4(i)=v(2,i)+sf*(v(2,i-1)-2.*v(2,i)+v(2,i+1))
        end do
        do i=2,n-1
            ne(2,i)=y1(i)
            ni(2,i)=y2(i)
            p(2,i)=y3(i)
            v(2,i)=y4(i)
        end do
        return
        end
        !!========================

end program plasma
```