**main.f90**

! program Exemplo PVC - 1 valor inicial desconhecido

use msflib ! systemqq

logical calling

 parameter (nmax=20000)

 common /param/ eta0,deta,etaend,f10,f20

 common /noeq/ n

 common /noeqt/ nt

 common /div/ ndiv

 common /prandtl/ pr

 common /etavec/ eta(nmax)

 common /f1vec/ f1(nmax)

 common /param1/ t10

 external ff,tf

 ! common /const/ ht2

 open(1,file='inpclass.txt')

 open(2,file='f1.txt')

 open(3,file='f2.txt')

 open(4,file='f3.txt')

 open(7,file='f.txt')

! Solution of a Boundary Value Problem - BVP

!

! data input

!

 read(1,\*)n

 write(\*,\*)'n=',n

 read(1,\*)eta0

 write(\*,\*)'eta0=',eta0

 read(1,\*)etaend

 write(\*,\*)'etaend=',etaend

 read(1,\*)deta

 write(\*,\*)'deta=',deta

 read(1,\*)f10

 write(\*,\*)'f10=',f10

 read(1,\*)f20

 write(\*,\*)'f20=',f20

 read(1,\*)f30a

 write(\*,\*)'f30a=',f30a

 read(1,\*)f30b

 write(\*,\*)'f30b=',f30b

 read(1,\*)maxit

 write(\*,\*)'maxit=',maxit

 read(1,\*)tol

 write(\*,\*)'tol=',tol

 ! PVC solution

 call secante(f30a,f30b,maxit,tol,c,ff)

 ndiv=etaend/deta

 write(\*,\*)'ndiv=',ndiv

 write(\*,\*) 'f20=',c

 close(2)

 close(3)

 close(4)

 close(7)

calling = systemqq('notepad f1.txt') ! list of data

calling = systemqq('notepad f2.txt') ! list of data

! calling = systemqq('notepad f3.txt') ! list of data

calling = systemqq('notepad f.txt') ! list of f-table

calling = systemqq('wgnuplot data.gnu') ! graph - velocities

 stop

 end

!\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

 subroutine fcn(n,t,fi,f,nelmax)

 dimension fi(nelmax),f(nelmax)

!

 f(1)=fi(2)

 f(2)=t-2\*fi(2)+sin(fi(1))

! f(2)=fi(3)

! f(3)=-fi(1)\*fi(3)/2

!

 return

 end

!\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

 subroutine fore(n,fcn,time,fi,tend,nelmax)

!

! implicit real \*8 (a-h,o-z)

 parameter (nd1=100)

 dimension fi(nelmax),f(nd1)

 common /const/ ht2

 external fcn

 k=0

 50 k=k+1

 time=min(time+ht2,tend)

 call fcn(n,time,fi,f,nelmax)

 do 100 i=1,n

 fi(i)=fi(i)+ht2\*f(i)

 100 continue

 if (time.lt.tend) goto 50

 return

 end

!------------------------------------------------------------------

 function ff(x)

 parameter (nmax=100)

 dimension tp(nmax)

 external fcn,rkqc

 common /noeq/ n

 common /flag/ iflag

 common /etavec/ eta1(1)

 common /f1vec/ f1(1)

 common /param/ eta0,deta,etaend,f10,f20

!

! initial values

!

 eta=eta0

 tp(1)=f10

 tp(2)=x

! tp(3)=x

!

 k=0

 if (iflag.eq.1) then

 eta1(k+1)=eta

 f1(k+1)=tp(1)

 write(2,\*)eta,tp(1)

 write(3,\*)eta,tp(2)

! write(4,\*)eta,tp(3)

 write(7,\*)'--------------------------------------------'

 write(7,\*)'Nr do passo eta f1 f2 f3'

 write(7,\*)'--------------------------------------------'

 write(7,\*)k,eta,(tp(l),l=1,n)

 endif

 ht2=deta/1

 nd=nmax

!

! beginning of time loop

!

 50 k=k+1

 tendi=eta+deta

! write(\*,\*)'-------------eta=',tendi

 call rk4ord(tp,n,eta,deta,fcn,nd)

! call fore(n,fcn,time,tp,tendi,nmax)

 if (iflag.eq.1) then

 if (k.eq.100.or.k.eq.200.or.k.eq.300.or.k.eq.400.or.k.eq.500.or.k.eq.600.or.k.eq.700.or.k.eq.800.or.k.eq.900.or.k.eq.1000) write(7,\*)k,tendi,(tp(l),l=1,n)

 eta1(k+1)=tendi

 f1(k+1)=tp(1)

 write(2,\*)tendi,tp(1)

 write(3,\*)tendi,tp(2)

! write(4,\*)tendi,tp(3)

 endif

!

 if (tendi.lt.etaend) then

 eta=tendi

 goto 50

 endif

 aux=tp(1)-1.

 ff=aux

 return

 end

!------------------------------------------------------

**rk4ord.f90**

 subroutine rk4ord(y,n,x,h,derivs,nd)

!

! rk4

!

 parameter (nmax=100,nd3=100)

 dimension y(nd),dydx(nd3),yt(nd3),dyt(nd3),dym(nd3)

 external derivs

 hh=h\*.5

 h6=h/6

 xh=x+hh

 call derivs(n,x,y,dydx,nd)

 do 11 i=1,n

 yt(i)=y(i)+hh\*dydx(i)

 11 continue

 call derivs(n,xh,yt,dyt,nd)

 do 12 i=1,n

 yt(i)=y(i)+hh\*dyt(i)

 12 continue

 call derivs(n,xh,yt,dym,nd)

 do 13 i=1,n

 yt(i)=y(i)+h\*dym(i)

 dym(i)=dyt(i)+dym(i)

 13 continue

 call derivs(n,x+h,yt,dyt,nd)

 do 14 i=1,n

 y(i)=y(i)+h6\*(dydx(i)+dyt(i)+2\*dym(i))

 14 continue

 return

 end

!234567890123456789012345678901234567890123456789012345678901234567890

**secante.f90**

 subroutine secante(xa0,xa1,maxit,tol,xatual,ff)

 common /convergence/ auxi3,rtol

 common /flag/ iflag

!

 iflag=0

 rtol=tol

 xantes = xa0

 auxi1 = ff(xantes)

 xpos = xa1

 auxi2 = ff(xpos)

 if (abs(auxi1).le.tol) then

 xatual = xantes

 auxi3=auxi1

 goto 333

 endif

 if (abs(auxi2).le.tol) then

 xatual = xpos

 auxi3 = auxi2

 goto 333

 endif

 do i=1,maxit

 xatual=xpos-auxi2\*(xpos-xantes)/(auxi2-auxi1)

 auxi3 = ff(xatual)

 ! write(10,\*)i,auxi3

 ! write(12,\*)i,xatual,auxi3

 if (abs(auxi3).le.tol) goto 333

 xantes = xpos

 auxi1=auxi2

 xpos = xatual

 auxi2 = auxi3

 enddo

 if (i.ge.maxit) then

 write (\*,\*)'Nao convergiu, F=',abs(auxi3)

 return

 endif

333 CONTINUE

 iflag=1

 aux5=ff(xatual)

 write (\*,\*)'A solucao secante eh x=',xatual,' f=',abs(auxi3)

 RETURN

 END

!--------------------------------------------------------------

**data.gnu**

set data style linespoints

set grid

set xlabel 'Tempo (eta)'

set ylabel 'Solução numérica x(eta)'

set title 'PVC Exemplo (f)'

plot 'f1.txt'

pause -1

**inpclass.txt**

2 ! n=number of equations (momentum)

0. ! eta0 = initial eta

5. ! etaend = eta at infinity

0.01 ! deta = eta-stepsize

1. ! f10 = f initial value

0. ! f20 = f-prime initial value

0.1 ! f30a = first f-2prime guessed initial value

1.2 ! f30b = second f-2prime guessed initial value

100 ! maxit - max number of iterations

1.e-6 ! tol = tolerance for secant method