**Project workspace: reservatorio**

**main.f90**

! program reservatorio

use msflib ! biblioteca que contem o comando systemqq

logical chamada

parameter (nelmax=100)

dimension tp(nelmax)

external fcn,rkqc

common /const/ ht2

common /param/ rps,vswept,r,rn,vs,cv,rar

common /param2/ temp0,vres,p0

common /param3/ vm

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

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

open(3,file='out-read.txt')

open(10,file='out-m.txt')

open(11,file='out-p.txt')

open(12,file='out-vm.txt')

!

! integrador de EDO's com passo fixo - RK 4a ordem

read(1,\*)n

write(3,\*)'n=',n

read(1,\*)tau0

write(3,\*)'tau0=',tau0

read(1,\*) tend

write(3,\*)'tend=',tend

read(1,\*) dtau

write(3,\*)'dtau=',dtau

read(1,\*) p0

write(3,\*)'p0=',p0

read(1,\*) iflag

write(3,\*)'iflag=',iflag

read(1,\*) ht2

write(3,\*)'ht2=',ht2

read(1,\*) rps

write(3,\*)'rps=',rps

read(1,\*) vswept

write(3,\*)'vswept=',vswept

read(1,\*) r

write(3,\*)'r=',r

read(1,\*) rn

write(3,\*)'rn=',rn

read(1,\*) rar

write(3,\*)'rar=',rar

read(1,\*) cv

write(3,\*)'cv=',cv

read(1,\*) vres

write(3,\*)'vres=',vres

read(1,\*) temp0

write(3,\*)'temp0=',temp0

! compute constants

rm0=p0\*vres/rar/temp0

write(3,\*)'rm0=',rm0

rhoar=rm0/vres

write(3,\*)'rhoar=',rhoar

vs=1./rhoar

write(3,\*)'vs=',vs

! initial values

time=tau0

tp(1)=rm0

tp(2)=p0

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

! loop para apresentar resultados intermediarios

! em cada intervalo dtau durante a integracao das equacoes

!

k=0

write(\*,\*) ' Table of results'

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

write(\*,\*)' Passo Nr time x(i) v(i) '

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

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

write(2,\*)time,(tp(l),l=1,n)

write(10,\*)time,tp(1)

write(11,\*)time,tp(2)

write(12,\*)time,vm

!

! beginning of time loop

!

50 k=k+1

tendi=time+dtau

write(\*,\*)'-------------time=',tendi

if (iflag.eq.0) then ! resolve por RK adaptativo

call odeint(tp,n,time,tendi,1.e-6,ht2,1.e-20,id1,id2,nelmax,fcn,rkqc)

else

if (iflag.eq.1) then ! resolve por RK passo fixo

call rk4ord(tp,n,time,dtau,fcn,nd)

else ! resolve por Forward Euler

call fore(n,fcn,time,tp,tendi,nelmax)

endif

endif

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

write(2,\*)tendi,(tp(l),l=1,n)

write(10,\*)tendi,tp(1)

write(11,\*)tendi,tp(2)

write(12,\*)tendi,vm

!

if (tendi.lt.tend) then

time=tendi

goto 50

endif

close(2)

close(3)

close(10)

close(11)

close(12)

chamada = systemqq('notepad outnum.txt') ! listagem dos dados

chamada = systemqq('wgnuplot dados-m.gnu') ! gráfico m

chamada = systemqq('wgnuplot dados-p.gnu') ! gráfico p

chamada = systemqq('wgnuplot dados-vm.gnu') ! gráfico vm

stop

end

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

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

dimension fi(nelmax),f(nelmax)

common /param/ rps,vswept,r,rn,vs,cv,rar

common /param2/ temp0,vres,p0

common /param3/ vm

!

efv=1-r\*((fi(2)/p0)\*\*(1./rn)-1.)

f(1)=vswept/vs\*rps\*efv\*cv

vm=f(1)

f(2)=rar\*temp0/vres\*f(1)

return

end

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

!234567890123456789012345678901234567890123456789012345678901234567890

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

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

**ode.f90**

subroutine odeint(ystart,nvar,x1,x2,eps,h1,hmin,nok,nbad,nd,derivs,rkqc)

parameter (maxstp=10000,nmax=100,two=2.0,zero=0.0,tiny=1.d-30)

parameter (nd1=100)

common /path/ kmax,kount,dxsav

dimension ystart(nd),yscal(nd1),y(nd1),dydx(nd1)

external derivs,rkqc

x=x1

h=sign(h1,x2-x1)

nok=0

nbad=0

kount=0

do 11 i=1,nvar

y(i)=ystart(i)

11 continue

if (kmax.gt.0) xsav=x-dxsav\*two

do 16 nstp=1,maxstp

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

do 12 i=1,nvar

yscal(i)=abs(y(i))+abs(h\*dydx(i))+tiny

12 continue

if ((x+h-x2)\*(x+h-x1).gt.zero) h=x2-x

call rkqc(y,dydx,nvar,x,h,eps,yscal,hdid,hnext,derivs,nd)

if (hdid.eq.h) then

nok=nok+1

else

nbad=nbad+1

endif

if ((x-x2)\*(x2-x1).ge.zero) then

do 14 i=1,nvar

ystart(i)=y(i)

14 continue

return

endif

if (abs(hnext).lt.hmin) then

write(\*,\*) 'stepsize small',hmin

stop

endif

h=hnext

16 continue

write(\*,\*) 'too many steps',nstp

stop

end

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

**rk.f90**

subroutine rkqc(y,dydx,n,x,htry,eps,yscal,hdid,hnext,derivs,nd)

!

! fifth-order RK

!

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

parameter (nmax=100, pgrow=-.20,pshrnk=-.25,fcor=1.d0/15.,one=1., safety=.9, errcon=6.e-4,nd2=100)

external derivs

dimension y(nd),dydx(nd),yscal(nd),ytemp(nd2),ysav(nd2),dysav(nd2)

xsav=x

do 11 i=1,n

ysav(i)=y(i)

dysav(i)=dydx(i)

11 continue

h=htry

1 hh=0.5\*h

call rk4(ysav,dysav,n,xsav,hh,ytemp,derivs,nd)

x=xsav+hh

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

call rk4(ytemp,dydx,n,x,hh,y,derivs,nd)

x=xsav+h

if (x.eq.xsav) then

write(\*,\*) 'stepsize not significant in rkqc',x

stop

endif

call rk4(ysav,dysav,n,xsav,h,ytemp,derivs,nd)

errmax=0.

do 12 i=1,n

ytemp(i)=y(i)-ytemp(i)

dummy=abs(ytemp(i)/yscal(i))

errmax=max(errmax,dummy)

12 continue

errmax=errmax/eps

if(errmax.gt.one) then

h=safety\*h\*(errmax\*\*pshrnk)

goto 1

else

hdid=h

if (errmax.gt.errcon) then

hnext=safety\*h\*(errmax\*\*pgrow)

else

hnext=4.d0\*h

endif

endif

do 13 i=1,n

y(i)=y(i)+ytemp(i)\*fcor

13 continue

return

end

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

subroutine rk4(y,dydx,n,x,h,yout,derivs,nd)

!

! rk4

!

parameter (nmax=100,nd3=100)

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

external derivs

hh=h\*.5

h6=h/6

xh=x+hh

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

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

14 continue

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

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

**dados-m.gnu**

set data style linespoints

set grid

set xlabel 'Tempo t'

set ylabel 'Solucao numerica m(t)'

set title 'Massa de ar'

plot 'out-m.txt'

pause -1

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

**dados-p.gnu**

set data style linespoints

set grid

set xlabel 'Tempo t'

set ylabel 'Solucao numerica p(t)'

set title 'Pressao do reservatorio'

plot 'out-p.txt'

pause -1

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

**dados-vm.gnu**

set data style linespoints

set grid

set xlabel 'Tempo t'

set ylabel 'Solucao numerica vm(t)'

set title 'vazao do ar'

plot 'out-vm.txt'

pause -1

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

**inpaula.txt**

2 ! n = numero de equacoes

0. ! tau0 = instante inicial [s]

200000. ! tend = tempo final de simulacao [s]

1000. ! dtau = intervalo de tempo para apresentar res. interm. [s]

0.1e6 ! p0 = pressao inicial do reservatorio [Pa]

1 ! iflag = 0 - RK adaptativo; 1 - RK passo fixo; 2 - For. Euler

1.e-3 ! ht2 = valor de passo inicial para RK adaptativo, RK passo fixo e FE

8.33 ! rps = rotacao do compressor [ciclo por seg]

0.1e-3 ! vswept = vol varrido no compressor [m^3]

0.1 ! r = razao espaco morto do compressor = Vc/Vs

1.2 ! rn = coeficiente politropico

287. ! rar = constante do gas ar [J/(kg.K)]

0.8 ! cv = coef. de valvulas

1. ! vres = volume do reservatorio [m^3]

298.15 ! temp0 = temperatura do ambiente [K]