**Main.f90**

! program Regenerator heat exchanger - 20 Jan 2013

use msflib ! biblioteca que contem o comando systemqq

logical chamada

parameter (nelmax=1000)

dimension tp(nelmax),t0(nelmax)

external fcn,rkqc,fcnad

common /const/ ht2

common /adim1/ rmgm,rmrm,rkad,rmad

common /adim2/ rmrad,hgmad,htmad,rmmad,cpfiad

common /temp/ tfeinad,tfiinad

common /disc/ ncel

common /vazao/ iflag2

common /param1/ dx,rk,ag

common /param2/ hfe,hfi,agl,atroca,rmatriz,cg

common /param4/ rm,tfein,cpfe,rmassfe,cvfe

common /param5/ rmfi,tfiin,cpfi,rmr,cvfi

common /tempo/ rmtrans

common /parede/ uaext,tzero

open(1,file='inp-dados.txt')

open(15,file='out-read.txt')

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

open(3,file='tempsist1.txt')

open(4,file='tempsist2.txt')

open(5,file='tempsist3.txt')

open(6,file='tempsist1-x.txt')

open(7,file='tempsist2-x.txt')

open(8,file='tempsist3-x.txt')

open(9,file='vazao.txt')

!

! Leitura de dados

!

read(1,\*)ncel

write(15,\*)'ncel=',ncel

read(1,\*)xl

write(15,\*)'xl=',xl

read(1,\*)dti

write(15,\*)'dti=',dti

read(1,\*)dte

write(15,\*)'dte=',dte

read(1,\*)dar

write(15,\*)'dar=',dar

read(1,\*)phi

write(15,\*)'phi=',phi

read(1,\*)cg

write(15,\*)'cg=',cg

read(1,\*)rhog

write(15,\*)'rhog=',rhog

read(1,\*)rm

write(15,\*)'rm=',rm

read(1,\*)tt

write(15,\*)'tt=',tt

read(1,\*)cpfe

write(15,\*)'cpfe=',cpfe

read(1,\*)cvfe

write(15,\*)'cvfe=',cvfe

read(1,\*)rhofi

write(15,\*)'rhofi=',rhofi

read(1,\*)cvfi

write(15,\*)'cvfi=',cvfi

read(1,\*)rk

write(15,\*)'rk=',rk

read(1,\*)rmfi

write(15,\*)'rmfi=',rmfi

read(1,\*)hfe

write(15,\*)'hfe=',hfe

read(1,\*)hfi

write(15,\*)'hfi=',hfi

read(1,\*)rhofe

write(15,\*)'rhofe=',rhofe

read(1,\*)cpfi

write(15,\*)'cpfi=',cpfi

read(1,\*)href

write(15,\*)'href=',href

read(1,\*)rmref

write(15,\*)'rmref=',rmref

read(1,\*)tzero

write(15,\*)'tzero=',tzero

read(1,\*)ht2

write(15,\*)'ht2=',ht2

read(1,\*)teta0

write(15,\*)'teta0=',teta0

read(1,\*)htime

write(15,\*)'htime=',htime

read(1,\*)tend

write(15,\*)'tend=',tend

!

! parametros de operacao

!

read(1,\*)tfein

write(15,\*)'tfein=',tfein

read(1,\*)tfiin

write(15,\*)'tfiin=',tfiin

read(1,\*)tini

write(15,\*)'tini=',tini

read(1,\*)tol1

write(15,\*)'tol1=',tol1

read(1,\*)iflag

write(15,\*)'iflag=',iflag

read(1,\*)iflag2

write(15,\*)'iflag2=',iflag2

read(1,\*)iperiod

write(15,\*)'iperiod=',iperiod

read(1,\*)tol2

write(15,\*)'tol2=',tol2

read(1,\*)isol

write(15,\*)'isol=',isol

read(1,\*)dpe

write(15,\*)'dpe=',dpe

read(1,\*)rkisol

write(15,\*)'rkisol=',rkisol

read(1,\*)hzero

write(15,\*)'hzero=',hzero

!

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

! calculo de parametros do modelo

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

!

write(15,\*)'\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*'

write(15,\*)'calculo de parametros do modelo'

write(15,\*)'\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*'

dtint=dti-tt

write(15,\*)'dtint=',dtint

pi=4\*atan(1.)

write(15,\*)'pi=',pi

dx=xl/ncel

write(15,\*)'dx=',dx

agt=pi\*(dti\*dti-dtint\*dtint)/4

write(15,\*)'agt=',agt

vtotal=xl\*pi\*(dte\*dte-dti\*dti)/4

write(15,\*)'vtotal=',vtotal

vmatriz=(1.-phi)\*vtotal

write(15,\*)'vmatriz=',vmatriz

vgrid=vmatriz+agt\*xl

write(15,\*)'vgrid=',vgrid

xlar=4\*vmatriz/pi/dar/dar

write(15,\*)'xlar=',xlar

agl=pi\*dar\*xlar

write(15,\*)'agl=',agl

atl=pi\*dti\*xl

write(15,\*)'atl=',atl

atroca=agl+atl

write(15,\*)'atroca=',atroca

rmg=rhog\*vgrid

write(15,\*)'rmg=',rmg

rmt=rhog\*pi\*dti\*tt\*xl

write(15,\*)'rmt=',rmt

rmatriz=rmg+rmt

write(15,\*)'rmatriz=',rmatriz

rmr=rhofi\*pi\*dtint\*dtint\*xl/4

write(15,\*)'rmr=',rmr

ags=(1.-phi)\*pi\*(dte\*dte-dti\*dti)

write(15,\*)'ags=',ags

rmassfe=rhofe\*phi\*vtotal

write(15,\*)'rmassfe=',rmassfe

ag=ags+agt

write(15,\*)'ag=',ag

apl=pi\*dte\*xl

write(15,\*)'apl=',apl ! calcula area molhada de parede entre isolamento e fluido quente

aple=pi\*dpe\*xl

write(15,\*)'aple=',aple ! calcula area molhada de parede externa

uaext=1./(1./apl/hfe+log(dpe/dte)/2/pi/rkisol/xl+1./aple/hzero) ! overall heat transfer coefficient

write(15,\*)'uaext=',uaext

!

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

! calculo de "n" - tamanho do vetor a integrar

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

n=3\*ncel

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

! initial values

!

time=teta0 !tempo adimensional inicial

do i=1,n

! if(isol.eq.0) then

tp(i)=tini !(temperaturas)

! else

! tp(i)=tini/tzero !(temperaturas)

! endif

enddo

k=0

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

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

write(\*,\*)' time T(i) '

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

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

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

write(3,\*)time,tp(1)

write(4,\*)time,tp(2)

write(5,\*)time,tp(3)

! if(isol.eq.0) then

rms=rm

! else

! rms=rmad

! endif

write(9,\*)time,rms

!

! beginning of time loop

!

50 k=k+1

tendi=time+htime

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

!

! armazena em t0(i) as temperaturas do instante anterior

!

do 110 i=1,n

t0(i)=tp(i)

110 continue

!

if(iflag.eq.0) then

! if(isol.eq.0) then

call odeint(tp,n,time,tendi,tol1,ht2,1.e-20,id1,id2,nelmax,fcn,rkqc)

! else

! call odeint(tp,n,time,tendi,tol1,ht2,1.e-20,id1,id2,nelmax,fcnad,rkqc)

! endif

else

if(iflag.eq.1) then

k=0

500 k=k+1

time=min(time+ht2,tendi)

call rk4ord(tp,n,time,ht2,fcn,nd)

if (time.lt.tendi) goto 500

endif

if (iflag.eq.2) call fore(n,fcn,time,tp,tendi,nelmax)

endif

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

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

write(3,\*)tendi,tp(1)

write(4,\*)tendi,tp(2)

write(5,\*)tendi,tp(3)

write(9,\*)tendi,rmtrans

!

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

!

if(iperiod.eq.0) then

t0norm=rnorm2(n,t0,nelmax)

do 120 l=1,n

t0(l)=tp(l)-t0(l)

120 continue

dtnorm=rnorm2(n,t0,nelmax)

! write(\*,\*)'dtnorm=',dtnorm,'t0norm=',t0norm

if(dtnorm.lt.tol2\*htime) then

! write(\*,\*) 'dtnorm=',dtnorm,'ratio norms=',

! \* dtnorm/t0norm

time=tendi

goto 300

endif

else

!

if(tendi.ge.tend) then

time=tendi

goto 300

endif

endif

time=tendi

goto 50

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

! end of time loop

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

300 continue

jj=1

dx=xl/ncel

dxi=dx/2

do j1=1,ncel

write(6,\*)dxi,tp(jj)

write(7,\*)dxi,tp(jj+1)

write(8,\*)dxi,tp(jj+2)

!

! incremento em cada celula

!

dxi=dxi+dx

jj=jj+3

enddo

!

! calculo da efetividade do trocador de calor

!

cq=rm\*cpfe !capacidade calorífica do fluido externo

cf=rmfi\*cpfi !capacidade calorífica do fluido frio

! if(cf.lt.cq) then

! cmin=cf

! else

! cmin=cq

! endif

! if(isol.eq.0) then

tpn=tp(n-1)

tp3=tp(3)

! else

! tpn=tp(n-1)\*tzero

! tp3=tp(3)\*tzero

! endif

epsilonq=cq\*(tfein-tpn)/cf/(tfein-tfiin)

write(\*,\*)'efetividadeq=',epsilonq

write(15,\*)'efetividadeq=',epsilonq

epsilonf=(tp3-tfiin)/(tfein-tfiin)

write(\*,\*)'efetividadef=',epsilonf

write(15,\*)'efetividadef=',epsilonf

qquente=cq\*(tfein-tpn)

write(15,\*)'qquente=',qquente,' Watts'

qfrio= cf\*(tp3-tfiin)

write(15,\*)'qfrio=',qfrio,' Watts'

!

close(2)

close(3)

close(4)

close(5)

close(6)

close(7)

close(8)

close(9)

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

chamada = systemqq('wgnuplot dados.gnu') ! gráfico temp 1a celula x tempo

chamada = systemqq('wgnuplot dados-x.gnu') ! gráfico temp x posicao

chamada = systemqq('wgnuplot dados-vazao.gnu') ! gráfico vazao fluido externo x tempo

stop

end

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

function rnorm2(n,x,nd)

!

! compute euclidean norm of a vector

!

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

dimension x(nd)

sum=0.d0

do i=1,n

sum=sum+x(i)\*x(i)

enddo

aux=sqrt(sum)

rnorm2=aux

return

end

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

!

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

dimension fi(nelmax),f(nelmax)

common /adim1/ rmgm,rmrm,rkad,rmad

common /adim2/ rmrad,hgmad,htmad,rmmad,cpfiad

common /temp/ tfeinad,tfiinad

common /disc/ ncel

common /vazao/ iflag2

common /param1/ dx,rk,ag

common /param2/ hfe,hfi,agl,atroca,rmatriz,cg

common /param4/ rm,tfein,cpfe,rmassfe,cvfe

common /param5/ rmfi,tfiin,cpfi,rmr,cvfi

common /tempo/ rmtrans

common /parede/ uaext,tzero

!

! write(\*,\*)'dx=',dx

if(iflag2.eq.1) then

rmtrans=(rm/2)\*cos(t/60.)+rm/2

else

rmtrans=rm

endif

! calculo das derivadas das temperaturas

!

jj=1

do j1=1,ncel

if(j1.eq.1.and.ncel.gt.1) then

qcond=-rk\*ag\*(fi(jj)-fi(jj+3))/dx !qcond,a

endif

if(j1.eq.ncel.and.ncel.gt.1) then

qcond=-rk\*ag\*(fi(jj)-fi(jj-3))/dx !qcond,b

endif

if(ncel.eq.1) qcond=0.

if(j1.gt.1.and.j1.lt.ncel) then

qcond=rk\*ag\*(fi(jj+3)-2\*fi(jj)+fi(jj-3))/dx

endif

! write(\*,\*)'qcond=',qcond

!

! calculo da derivada do sistema 1

!

qg=hfe\*agl\*(fi(jj+1)-fi(jj))/ncel

qt=hfi\*atroca\*(fi(jj+2)-fi(jj))/ncel

! write(\*,\*)'qg=',qg

! write(\*,\*)'qt=',qt

f(jj)=(qg+qcond+qt)/(rmatriz\*cg/ncel)

!

! calculo da derivada do sistema 2

!

if(j1.eq.1) then

tin=tfein

else

tin=fi(jj+1-3)

endif

! compute heat leak across insulation

qp=uaext\*(tzero-fi(jj+1))/ncel

f(jj+1)=(rmtrans\*cpfe\*(tin-fi(jj+1))-qg+qp)/(rmassfe\*cvfe/ncel)

!

! calculo da derivada do sistema 3

!

if(j1.eq.ncel) then

trin=tfiin

else

trin=fi(jj+2+3)

endif

f(jj+2)=(rmfi\*cpfi\*(trin-fi(jj+2))-qt)/(rmr\*cvfi/ncel)

!

! incremento em cada celula

!

jj=jj+3

enddo

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

!234567890123456789012345678901234567890123456789012345678901234567890

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=1000)

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=1000)

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=1000)

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=1000)

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-vazao.gnu

set data style linespoints

set xlabel ' '

set ylabel ' '

set title ' '

plot 'vazao.txt'

pause -1

Dados-x.gnu

set data style linespoints

set xlabel ' '

set ylabel ' '

set title ' '

plot 'tempsist1-x.txt','tempsist2-x.txt','tempsist3-x.txt'

pause -1

Dados.gnu

set data style linespoints

set xlabel ' '

set ylabel ' '

set title ' '

plot 'tempsist1.txt','tempsist2.txt','tempsist3.txt'

pause -1

inp-dados.txt

20 ! ncel = numero de celulas

1. ! xl = comprimento do regenerador [m]

5.e-2 ! dti = diametro do tubo interno [m]

10.e-2 ! dte = diametro do tubo externo [m]

2.e-3 ! dar = diametro do arame da matriz [m]

0.95 ! phi = porosidade da matriz

896. ! cg = calor especifico do material da matriz [J/(kg.K)]

2707. ! rhog = densidade do material da matriz [kg/m^3]

0.1 ! rm = vazao massica do fluido externo [kg/s]

1.e-3 ! tt = espessura do tubo interno [m]

1.e3 ! cpfe = calor esp. pres. const. fluido externo [J/(kg.K)]

713. ! cvfe = calor esp. vol. const. fluido externo [J/(kg.K)]

1000. ! rhofi = densidade do fluido interno [kg/m^3]

4.18e3 ! cvfi = calor esp. vol. const. fluido interno [J/(kg.K)]

204. ! rk = condutividade termica do material da matriz [W/(m.K)]

0.01 ! rmfi = vazao massica de fluido interno [kg/s]

100. ! hfe = coeficiente transf. calor conveccao fluido ext. [W/(m^2.K)]

100. ! hfi = coeficiente transf. calor conveccao fluido int. [W/(m^2.K)]

1.165 ! rhofe = densidade do fluido externo [kg/m3]

4.18e3 ! cpfi = calor esp. pres. const. fluido interno [J/(kg.K)]

200. ! href = coeficiente transf. calor conveccao de referencia [W/(m^2.K)]

1. ! rmref = vazao massica de referencia [kg/s]

298.15 ! tzero = temperatura ambiente externa [K]

0.01 ! ht2 = passo de tempo adimensional inicial

0. ! teta0 = tempo adimensional inicial

10. ! htime = passo externo de tempo adimensional

1000. ! tend = tempo adimensional final de integracao

373.15 ! tfein = temperatura de entrada do fluido externo [K]

293.15 ! tfiin = temperatura de entrada do fluido interno [K]

293.15 ! tini = temperatura inicial do aparato [K]

1.e-4 ! tol1 = tolerancia para o RK de passo adaptativo

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

1 ! iflag2 = 1 - vazao variavel com o cos do tempo; 0 - vazao fixa

1 ! iperiod = 0 - simula até regime permanente; 1 - simula até tend especificado

1.e-3 ! tol2 - tolerancia para entrar em regime permanente

0 ! isol = 0 - mod dimensional; 1 - modelo não-dimensional

12.e-2 ! dpe = diametro do isolamento do tubo externo [m]

1. ! rkisol = condutividade termica do material do isolamento [W/(m.K)]

5. ! hzero = coeficiente transf. calor conveccao ambiente ext. [W/(m^2.K)]