**princlugraf.f90**

! program elgaussdirect

! programa principal para resolver eq lineares - metodo direto

!

use msflib ! biblioteca que contem o comando systemqq

logical chamada

parameter (np=10)

dimension a(np,np)

dimension b(np),aux(np)

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

open(8,file='dados1')

open(9,file='dados2')

!

! Leitura de dados

!

read(1,\*)n

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

write(\*,\*)'a matriz a(i,j) eh:'

do i=1,n

read(1,\*) (a(i,j),j=1,n)

write(\*,\*)(a(i,j),j=1,n)

enddo

!

write(\*,\*)'o vetor b(i) eh:'

read (1,\*) (b(i),i=1,n)

write(\*,\*)'b(i)=', (b(i),i=1,n)

do i=1,n

aux(i)=b(i)

enddo

do i=1,n

do j=1,n

aux(j)=b(j)

enddo

aux(1)=b(1)\*i

call ludcmp(a,n,np,indx,d)

call lubksb(a,n,np,indx,aux)

!

! solucao

!

write(\*,\*)'a solucao eh'

do j=1,n

write(\*,\*)'x(',j,')=',aux(j)

enddo

write(8,\*)i,aux(1)

write(9,\*)i,aux(2)

enddo

close(8)

close(9)

! chamada = systemqq('edit dados1') ! listagem dos dados

! chamada = systemqq('edit dados2') ! listagem dos dados

chamada = systemqq('notepad dados1') ! listagem dos dados

chamada = systemqq('notepad dados2') ! listagem dos dados

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

! stop

end

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

**dados.gnu**

set data style linespoints

set grid

set xlabel 'Parametro P'

set ylabel 'Solucao x(1) e x(2)'

set title 'Estudo parametrico'

plot 'dados1','dados2'

pause -1

**input.txt**

4 ! n

6.d0 -2.d0 2.d0 4.d0 ! a(1,j)

12.d0 -8.d0 6.d0 10.d0 ! a(2,j)

3.d0 -13.d0 9.d0 3.d0 ! a(3,j)

-6.d0 4.d0 1.d0 -18.d0 ! a(4,j)

12.d0 34.d0 27.d0 -38.d0 ! b(i)

**lubksb.f90**

SUBROUTINE lubksb(a,n,np,indx,b)

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

dimension indx(np),a(np,np),b(np)

ii=0

do 12 i=1,n

ll=indx(i)

sum=b(ll)

b(ll)=b(i)

if (ii.ne.0)then

do 11 j=ii,i-1

sum=sum-a(i,j)\*b(j)

11 continue

else if (sum.ne.0.) then

ii=i

endif

b(i)=sum

12 continue

do 14 i=n,1,-1

sum=b(i)

do 13 j=i+1,n

sum=sum-a(i,j)\*b(j)

13 continue

b(i)=sum/a(i,i)

14 continue

return

END

**ludcmp.f90**

SUBROUTINE ludcmp(a,n,np,indx,d)

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

PARAMETER (NMAX=10,TINY=1.0e-20)

dimension indx(np),a(np,np),vv(NMAX)

d=1.

do 12 i=1,n

aamax=0.

do 11 j=1,n

if (abs(a(i,j)).gt.aamax) aamax=abs(a(i,j))

11 continue

if (aamax.eq.0.) then

! do ja=1,n

! do jb=1,n

! write(\*,\*)'a(',ja,',',jb,')=',a(ja,jb)

! enddo

! enddo

pause 'singular matrix in ludcmp'

endif

vv(i)=1./aamax

12 continue

do 19 j=1,n

do 14 i=1,j-1

sum=a(i,j)

do 13 k=1,i-1

sum=sum-a(i,k)\*a(k,j)

13 continue

a(i,j)=sum

14 continue

aamax=0.

do 16 i=j,n

sum=a(i,j)

do 15 k=1,j-1

sum=sum-a(i,k)\*a(k,j)

15 continue

a(i,j)=sum

dum=vv(i)\*abs(sum)

if (dum.ge.aamax) then

imax=i

aamax=dum

endif

16 continue

if (j.ne.imax)then

do 17 k=1,n

dum=a(imax,k)

a(imax,k)=a(j,k)

a(j,k)=dum

17 continue

d=-d

vv(imax)=vv(j)

endif

indx(j)=imax

if(a(j,j).eq.0.)a(j,j)=TINY

if(j.ne.n)then

dum=1./a(j,j)

do 18 i=j+1,n

a(i,j)=a(i,j)\*dum

18 continue

endif

19 continue

return

END