

Jan 04, 13 12:11

projeto3A_02Jan13.f90

Page 1/3

```

PROGRAM projeto3A
! Map05 2012/2 Terceiro Projeto Computacional
! necessidade de armazenagem:  $(4n^2 + n)$  8 bytes
!  $n = 10^3$  ::  $4 \cdot 10^6 \cdot 8 = 32$  MB
!  $n = 3 \cdot 10^3$  ::  $4 \cdot 9 \cdot 10^6 \cdot 8 = 288$  MB
!  $n = 10^4$  ::  $4 \cdot 10^8 \cdot 8 = 3.2$  GB

integer, parameter :: nmax = 3000
real (kind=8) :: vb(nmax*nmax), An(nmax*nmax)
real (kind=8) :: Li(nmax*nmax), Ei(nmax*nmax), aux(nmax)
real (kind=8) :: h, xi, yj
integer :: n, i, j, k, info
integer :: nargin, un
character (len=120) :: arqinp, arqout
character (len=20) :: non, noibin
!
! gfortran projeto3A.f90 -llapack -lblas -o projeto3a
!
nargin = command_argument_count()
if (nargin < 1) then
    un=6
    write(un, '(A)') 'ERRO: nome do arquivo de entrada deve ser fornecido.'
    write(un, '(A)') 'USO: ./projeto3a arq_entrada'
    write(un, '(A)') 'USO: ./projeto3a arq_entrada arq_saida'
    write(un, '(A)') 'Primeira linha de arq_entrada deve conter n ler_LiEi'
    write(un, '(A)') 'n: inteiro parametro do problema'
    write(un, '(A)') 'ler_LiEi (inteiro): se 0 : gerar Li,Ei e salvar no diretorio'
    write(un, '(A)') 'senao: ler Li,Ei salvos no diretorio'
    write(un, '(A)') 'As 4(n+2) linhas subsequentes deverao conter as condicoes de contorno'
    write(un, '(A)') 'em y=0,x=1,y=1,x=0, respectivamente.'
    write(un, '(A)') 'As (n+2)(n+2) linhas subsequentes deverao conter os valores de'
    write(un, '(A)') 'g(i*h,j*h),i=0,...,n+1;j=0,...,n+1'
    STOP
elseif (nargin < 2) then
    arqout='stdout'
    un=6
else
    call get_command_argument(2, arqout)
    open(unit=2, file=arqout, status='new')
    un=2
end if
call get_command_argument(1, arqinp)
open(unit=1, file=arqinp, status='old')
write(un, '(A)') 'Relatorio do programa projeto3a'
write(un, '(A19,2x,A40)') 'arquivo de entrada:', arqinp
write(un, '(A17,2x,A40)') 'arquivo de saida:', arqout
read(1, *) n
write(un, '(A12,2x,I2)') 'parametro n:', n
write(non, *) n
!
! n+1 = numero de intervalos na discretizacao do intervalo [0,1]
h = 1.0D0/(n+1.0D0)
! Passo 1: construo do grande vetor do lado direito
! primeiramente: o termo nao-homogeneo
do j=1,n
    do i=1,n
        vb(i+(j-1)*n) = -h*h*f(i*h,j*h)
    end do
end do
! armazenaremos a solucao por linhas, e assim nodos vizinhos em
! x sao tambem vizinhos na estrutura de dados
do j=1,n
    yj = j*h
    do i=1,n
        xi=i*h
        k = i+(j-1)*n
        if (j.eq.1) then
! vizinho de baixo estah no contorno
        vb(k) = vb(k) + g( xi,yj-h )
        
```

Jan 04, 13 12:11

projeto3A_02Jan13.f90

Page 2/3

```

    end if
    if (j.eq.n) then
!      vizinho de cima estah no contorno
      vb(k) = vb(k) + g( xi,yj+h )
    end if
    if (i.eq.1) then
!      vizinho da esquerda estah no contorno
      vb(k) = vb(k) + g( xi-h,yj )
    end if
    if (i.eq.n) then
!      vizinho da direita estah no contorno
      vb(k) = vb(k) + g( xi+h,yj )
    end if
  end do
end do
! agora vamos imprimir o vetor do lado direito
!write(*,*) 'Passo 1: vetor do lado direito: '
!do k=1,n*n
!  write(*,*) vb(k)
!end do

! construcao da matriz An
! primeiro: An = 4*I
call dlaset('All',n,n,0.0D0,4.0D0,An,n)
! agora introduziremos as 2 subdiagonais nao-nulas
do i=2,n
  k = i+(i-1)*n
  An(k-1)=-1.0D0
  An(k-n)=-1.0D0
end do
! estamos prontos com o bloco An
! Passo 2:  $L1 * L1' = An$  , primeiramente  $Li \leftarrow An$ 
call dcopy(n*n,An,1,Li,1)
! abaixo:  $L1 * L1' = A1$ ,  $L1$  armazenado em  $Li$ 
call dpotrf('Lower',n,Li,n,info)
!
! Passo 3: resolvemos  $L1 * e1 = b1$ 
call dtrsv('Lower','Notrans','Nondiag',n,Li,n,vb,1)
do i=1,n-1
! iteracao avancada: geramos  $L1, E1, L2, E2, \dots, Ln$  ao mesmo tempo em que
! aplicamos substituicao direta
! Passo 4: abaixo:  $Ei = -I$ 
call dlaset('All',n,n,.0D0,-1.0D0,Ei,n)
! Passo 4: agora resolveremos  $Ei * Li' = -I$ 
call dtrsm('Right','Lower','Transpose','NUnit',n,n,1.0D0,Li,n,Ei,n)
! bom momento para salvar ( $Li, Ei$ )
write(noibin,*) i
noibin='LiEi'//trim(adjustl(non))//trim(adjustl(noibin))//'.bin'
open(unit=3,file=noibin,status='replace',form='unformatted')
do k=1,n*n
  write(3) Li(k)
end do
do k=1,n*n
  write(3) Ei(k)
end do
close(3)
! Passo 5:: calcular novo  $Li$  :  $Li * Li' = An - Ei * Ei'$ 
call dcopy(n*n,An,1,Li,1)
call dsyrk('Lower','Notrans',n,n,-1.0D0,Ei,n,1.0D0,Li,n)
! agora  $Li$  armazena  $An - Ei * Ei'$ 
call dpotrf('Lower',n,Li,n,info)
! agora temos o  $Li$  final do Passo 5
! Passo 6: agora devemos fazer  $b_{(i+1)} = b_{(i+1)} - Ei * ei$ 
call dgemv('Notrans',n,n,-1.0D0,Ei,n,vb((i-1)*n+1),1,1.0D0,vb(i*n+1),1)
! Passo 7: resolvemos  $L_{(i+1)} * e_{(i+1)} = b_{(i+1)}$ 
call dtrsv('Lower','Notrans','Nondiag',n,Li,n,vb(i*n+1),1)
end do
write(noibin,*) n
noibin='LiEi'//trim(adjustl(non))//trim(adjustl(noibin))//'.bin'

```

Jan 04, 13 12:11

projeto3A_02Jan13.f90

Page 3/3

```

open(unit=3,file=noibin,status='replace',form='unformatted')
do k=1,n*n
  write(3) Li(k)
end do
close(3)
write(un,*) 'primeira solucao vb()'
do k=1,n*n
  write(un,*) vb(k)
end do
! Passo 8: resolvemos  $L_n' * x_n = e_n$ 
call dtrsv('Lower','Trans','Nondiag',n,Li,n,vb((n-1)*n+1),1)
do i=n-1,1,-1
  ! iteracao retrasada: resgataremos  $L_{n-1}, E_{n-1}, \dots, L_2, E_1, L_1$ 
  ! ao mesmo tempo em que aplicamos substituicao reversa
  write(noibin,*) i
  noibin='LiEi'//trim(adjustl(non))//trim(adjustl(noibin))//'.bin'
  open(unit=3,file=noibin,status='old',form='unformatted')
  do k=1,n*n
    read(3) Li(k)
  end do
  do k=1,n*n
    read(3) Ei(k)
  end do
  close(3)
  ! Passo 9: agora devemos fazer  $x_i = x_i - Ei' * x_{i+1}$ 
  call dcopy(n,vb(i*n+1),1,aux,1)
  call dtrmv('Upper','Trans','Nondiag',n,Ei,n,aux,1)
  call daxpy(n,-1.0D0,aux,1,vb((i-1)*n+1),1)
  ! Passo 10: resolvemos  $L_i' * x_i = e_i$ 
  call dtrsv('Lower','Trans','Nondiag',n,Li,n,vb((i-1)*n+1),1)
end do
write(un,*) 'segunda solucao vb()'
do k=1,n*n
  write(un,*) vb(k)
end do
write(un,*) n
do j=0,n+1
  do i=0,n+1
    if (j.eq.0) then
      write(un,*) g(i*h,.0D0)
    else if (j.eq.(n+1)) then
      write(un,*) g(i*h,1.0D0)
    else if (i.eq.0) then
      write(un,*) g(.0D0,j*h)
    else if (i.eq.(n+1)) then
      write(un,*) g(1.0D0,j*h)
    else
      write(un,*) vb(i+(j-1)*n)
    end if
  end do
end do
END PROGRAM

FUNCTION f(x,y)
  real (kind=8) :: f,x,y
  f = -x*x*y*y
END FUNCTION

FUNCTION g(x,y)
  real (kind=8) :: g,x,y
  g = 273.0-50.0*x*x*y*y
END FUNCTION

```