

```

1  program autovalor
2
3  ! Projeto 1 - Introdução à Programação
4  ! Nome: Henrique Krastins Okuti
5  ! Contato: henrique.okuti@usp.br
6
7  implicit none
8
9      real*8, dimension(:), allocatable :: x,iaux
10     real*8 r,vepsilon
11     real*8 lambda,term
12     real*8 raiz2, raiz7, pi, um_menosraiz5, um_maisraiz5, raiz10, raiz3
13     integer n, i, j, k, contador, m
14     logical :: cont= .True.
15     real*8, dimension(3,3) :: matriz1
16     real*8, dimension(4,4) :: matriz2
17     real*8, dimension(5,5) :: matriz3
18
19     ! Algumas operações úteis para definirmos os elementos das matrizes
20     raiz2 = sqrt(2.d0)
21     raiz3 = sqrt(3.d0)
22     raiz7 = sqrt(7.d0)
23     raiz10 = sqrt(10.d0)
24     pi = 4*atan(1.d0)
25     um_menosraiz5 = 1.d0 - sqrt(5.d0)
26     um_maisraiz5 = 1 + sqrt(5.d0)
27
28     ! Definimos a matriz1:
29     matriz1 = reshape((/ (raiz2), 8.d0, 13.d0 ,8.d0, 3.d0, 5.d0, 13.d0, 5.d0,
30     (raiz7) /), (/3,3/))
31
32     ! Agora confirmamos ela escrevendo ela na tela
33     write(*,*) "Matriz 1:"
34
35     i = 1
36     m = 3
37     n = 3
38     do i=1,m
39         write(*,*) (matriz1(i,j),j=1,3)
40     enddo
41
42
43     ! Definimos a matriz2 membro a membro pois tive problemas com o tamanho da
44     expressão no compilador
45     matriz2(1,1) = raiz10
46     matriz2(1,2) = -2
47     matriz2(1,3) = 3
48     matriz2(1,4) = 2
49     matriz2(2,1) = -2
50     matriz2(2,2) = 11
51     matriz2(2,3) = -3
52     matriz2(2,4) = pi
53     matriz2(3,1) = 3
54     matriz2(3,2) = -3
55     matriz2(3,3) = 6
56     matriz2(3,4) = 3.5
57     matriz2(4,1) = 2
58     matriz2(4,2) = pi
59     matriz2(4,3) = 3.5
60     matriz2(4,4) = 6
61
62     ! Confirmamos os seus elementos escrevendo eles na tela
63     write(*,*) "Matriz 2:"
64
65     i = 1
66     m = 4
67     n = 4
68     do i=1,m

```

```

68     write(*,*) (matriz2(i,j),j=1,4)
69 enddo
70
71 ! Definimos a matriz3 membro a membro pelo mesmo motivo que a matriz2
72 matriz3(1,1) = -10.d0
73 matriz3(1,2) = um_maisraiz5
74 matriz3(1,3) = -3.d0
75 matriz3(1,4) = (-13.d0/15.d0)
76 matriz3(1,5) = -1.d0
77 matriz3(2,1) = um_maisraiz5
78 matriz3(2,2) = -10.d0
79 matriz3(2,3) = raiz3
80 matriz3(2,4) = -4.d0
81 matriz3(2,5) = -27.d0
82 matriz3(3,1) = -3.d0
83 matriz3(3,2) = raiz3
84 matriz3(3,3) = um_menosraiz5
85 matriz3(3,4) = -3.d0
86 matriz3(3,5) = -3.d0
87 matriz3(4,1) = -(13.d0/15.d0)
88 matriz3(4,2) = -4.d0
89 matriz3(4,3) = -3.d0
90 matriz3(4,4) = -6.d0
91 matriz3(4,5) = -4.d0
92 matriz3(5,1) = -1.d0
93 matriz3(5,2) = -27.d0
94 matriz3(5,3) = -3.d0
95 matriz3(5,4) = -4.d0
96 matriz3(5,5) = -13.d0
97
98 ! Confirmamos os elementos escrevendo a matriz na tela
99 write(*,*)"Matriz 3:"
100 i = 1
101 m = 5
102 n = 5
103 do i=1,m
104     write(*,*) (matriz3(i,j),j=1,5)
105 enddo
106
107 write(*,*)"Dimensao da matriz:"
108 read(*,*)n
109 allocate(x(n))
110 allocate(xaux(n)) ! Leio a dimensão n da matriz que se deseja calcular o
autovalor e gero
111
! os vetores x(n) e xaux(n), que serão utilizados para
multiplicar a matriz
112
113 write(*,*)"Numero de interacoes:"
114 read(*,*)k ! Número máximo de loops de multiplicação que serão
realizados antes de finalizar o programa
115 write(*,*)"Precisao desejada (casas decimais):"
116 read(*,*)vepsilon ! Número da precisao desejada
117
118 i = 1
119 do while (i.LE.n)
120     call random_number(r)
121     x(i) = r ! Gero um vetor aleatório x com n elementos
122     i = i+1
123 end do
124
125
126 write(*,*)"Vetor utilizado para o chute inicial:"
127 write(*,*)(x(i),i=1,n)
128 write(*,*) "CALCULANDO"
129
130 contador = 0
131 do while(cont)
132     contador = contador +1 ! O contador será usado para ser comparado com o
número máximo de loops, "K",

```

```

133                                     ! lido anteriormente. Ele será incrementado em 1
                                     a cada loop
134 do i= 1, n
135     term = 0.0d0      ! 'term' representa cada linha resultante da
                        multiplicação entre a matriz e x, para cada uma das i linhas
136
137     if (n.EQ.3) then
138         do j= 1, n
139             term = (term + x(j)*matriz1(i,j))    ! Faço a operação descrita
                                                acima
140
141         end do
142     else if (n.EQ.4) then
143         do j= 1, n
144             term = (term + x(j)*matriz2(i,j))    ! Faço a operação descrita
                                                acima
145
146         end do
147     else if (n.EQ.5) then
148         do j= 1, n
149             term = (term + x(j)*matriz3(i,j))    ! Faço a operação descrita acima
150
151         end do
152     endif
153
154     xaux(i) = term    ! Defino a linha "i" do vetor xaux como o termo 'term'.
155     ! Ainda não podemos alterar o valor do vetor original x pois ele está
                        sendo usado na multiplicação,
156     ! então xaux está sendo usado para armazenar todos os valores que serão
                        substituídos
157     ! de uma vez após o término da operação
158
159 end do
160 x = xaux      ! Após a multiplicação faço a substituição dos valores de x
                pelos armazenados em xaux
161
162 ! Caso a diferença entre a última aproximação do autovalor dominante e a
nova seja menor que um valor epsilon
163 ! OU caso o número máximo de loops 'k' tenha sido atingido, o programa é
interrompido
164 if (n.NE.5) then
165     if (abs(lambda-maxval(x)).LE.(10.d0**(-vepsilon)) .OR. contador.GE.k)
        then
166         cont = .False.
167     end if
168     ! Mesmo caso da interrupção mas para a matriz 5x5 usamos minval(x) por
questão de convergência
169 else
170     if (abs(lambda-minval(x)).LE.(10.d0**(-vepsilon)) .OR. contador.GE.k)
        then ! Caso a diferença entre a última aproximação do autovalor
dominante e a nova seja menor que um valor epsilon OU caso o número
máximo de loops 'k' tenha sido atingido, o programa é interrompido
171         cont = .False.
172     end if
173 endif
174 write(*,*) "Vetor:", (x(i), i=1, n)
175 if (n.NE.5) then
176     lambda = maxval(x)    ! o autovalor dominante lambda é igual ao maior
número do vetor x
177     x = x/lambda    ! divido x pelo autovalor para ajustá-lo para o próximo
ciclo
178     write(99,*) contador, lambda    ! Escrevo o autovalor e o numero da
interação no loop
179 else
180     lambda = minval(x)    ! Para o caso n = 5 tomamos o menor valor para o
cálculo do vetor x
181     ! Caso contrário a solução fica oscilando sem saber a
direção de convergência
182     x = x/lambda    ! divido x pelo autovalor para ajustá-lo para o próximo

```

```
183         ciclo
           write(99,*)contador,lambda ! Escrevo o número da interação no loop e o
           autovalor
184     endif
185 end do
186
187 write(*,*)"O valor do autovalor dominante:"
188 write(*,*)lambda ! Retorna no terminal o autovalor dominante
189
190 end program autovalor
```