

UMA APLICAÇÃO DE SIMETRIZADORES
PARA O PROBLEMA DE AUTOVALORES

HELOÍSA GUEDES DE ALCÂNTARA

ORIENTADOR

JOSÉ VITÓRIO ZAGO

Dissertação apresentada no Instituto
de Matemática, Estatística e Ciência
da Computação, UNICAMP, como requisí-
to parcial para obtenção do título
de Mestre em Matemática Aplicada.

Julho 1980.

Aos meus pais

/

AGRADECIMENTOS:

Ao professor Dr. JOSÉ VITÓRIO ZAGO, pela orientação e esclarecimentos dados durante a pesquisa.

Ao professor Dr. BISWA NATH DATTA, pelo incentivo e entusiasmo demonstrados em todo desenvolvimento deste trabalho.

Aos professores do IMECC, pela orientação durante o curso de pós-graduação.

À VALÉRIA DE PODESTÁ GOMES, pelos ensinamentos computacionais.

Aos colegas e funcionários, pela atenção dispensada e convivência diária agradável.

Aos amigos, por terem tornado gratificante minha estada em Campinas.

À minha família, pela compreensão e apoio durante todos os momentos.

Ao ÁLVARO JOSÉ DAMIÃO, pelos favores prestados.

À VERA MARIA DE QUEIROZ MOURA, pela datilografia do tex-
to.

Aos professores da FAFI de Sorocaba, pelo alicerce matemático que contribuiu para a realização deste trabalho.

À CAPES, pela bolsa de estudos concedida para o desenvolvimento desta pesquisa.

SUMÁRIO

Este trabalho tem por finalidade determinar os autovaleores reais e distintos de uma matrizcompanheira real.

No Capítulo zero, são dadas as definições e notações usadas durante o trabalho.

No Capítulo 1, é proposto um novo método, usando-se símetrizadores.

No Capítulo 2, é descrito o método QR, com perturbação na matriz original.

Finalmente, no Capítulo 3, é feita a comparação entre os dois métodos e são apresentados os resultados dos testes computacionais.

No Apêndice, constam as listagens dos algoritmos programados.

SUMMARY

The scope of this work is to compute the distinct and real eigenvalues of a companion real matrix.

First the definitions and notations used in the work are given.

In the 1st chapter a new method that uses symmetrizers is suggested.

In the 2nd chapter the QR method with shift in the original matrix is described.

Finally in the 3rd chapter the comparison between the two methods is done and the result of computational tests are presented, too.

The appendix contains the listing of the programs of the algorithms.

ÍNDICE

	<u>Página</u>
Capítulo 0: Notações e definições	1
Capítulo 1: Método proposto	6
1.1. Introdução	6
1.2. Matrizes de Hankel	7
1.3. Algoritmos usados:	15
1.3.1. Construção da matriz Hankel	15
1.3.2. Construção da matriz A	17
1.3.3. Redução de $Ax = \lambda Bx$ para $Ez = \lambda z$	18
1.3.4. Método de Householder	20
1.3.5. Variação racional do QL	23
1.4. Diagrama de blocos	29
1.5. Aplicação	30
Capítulo 2: Método QR	35
2.1. Introdução	35
2.2. Algoritmos usados	35
2.2.1. Balanceamento de uma matriz	35
2.2.2. Método QR para matrizes reais de Hessenberg	40
2.3. Diagrama de blocos	49
2.4. Aplicação	50
Capítulo 3: Comparação	63
3.1. Experiência computacional	63

	<u>Página</u>
3.2. Conclusão	65
Capítulo 4: Apêndice (Listagens)	66
4.1. HANKEL	66
4.2. PRODUT	67
4.3. REDUZ	68
4.4. HOUSE	70
4.5. QLMUD	72
4.6. DATTA	75
4.7. BALANC	76
4.8. HOR	79
4.9. QR	83
Referências	84

CAPÍTULO 0

NOTAÇÕES E DEFINIÇÕES:

O problema algébrico fundamental de autovalores está na determinação dos valores de λ para os quais o conjunto de n equações lineares homogêneas de n incógnitas

$$Ax = \lambda x \quad (1)$$

tem uma solução não-trivial. A equação (1) pode ser escrita na forma

$$(A - \lambda I)x = 0$$

e para λ arbitrário, este conjunto de equações tem apenas a solução $x = 0$. Dizemos que existe solução não-trivial se e somente se a matriz $(A - \lambda I)$ é singular, isto é,

$$\det(A - \lambda I) = 0.$$

Desenvolvendo o determinante, teremos a equação polinomial explícita

$$\alpha_0 + \alpha_1 \lambda + \dots + \alpha_{n-1} \lambda^{n-1} + (-1)^n \lambda^n = 0. \quad (2)$$

Definição 1: Chamamos de equação característica da matriz A à equação (2) e de polinômio característico, o polinômio à esquerda da igualdade.

Definição 2: As raízes λ da equação (2) são chamadas de autovalores da matriz A.

Definição 3: Uma solução simétrica X da equação matricial $XS = S^t X$ é dita simetrizadora de S, e a matriz S é dita simétrizável nesse caso.

Definição 4: Chamamos de matriz companheira do polinômio característico de A, a matriz:

$$C = \begin{bmatrix} 0 & 1 & 0 & \dots & 0 \\ 0 & 0 & 1 & \dots & 0 \\ \vdots & \vdots & \vdots & & \vdots \\ 0 & 0 & 0 & \dots & 1 \\ c_1 & c_2 & c_3 & \dots & c_n \end{bmatrix}$$

cuja equação característica:

$$(-1)^n [\lambda^n - c_n \lambda^{n-1} - c_{n-1} \lambda^{n-2} - \dots - c_1] = 0 ,$$

é idêntica à equação característica de A.

Definição 5: Definimos forma quadrática $x^t Ax$, de n variáveis x_1, x_2, \dots, x_n , associada à matriz simétrica real A por:

$$x^t Ax = \sum_{i,j=1}^n a_{ij} x_i x_j$$

onde x é o vetor com componentes x_i .

Definição 6: Uma forma quadrática com a matriz A é definida:

positiva definida	de acordo com $x^t Ax$	> 0
negativa definida		< 0
positiva semi definida		≥ 0 , para todo $x \neq 0$.
negativa semi definida		≤ 0

Definição 7: A matriz A de uma forma quadrática positiva - (negativa) definida $x^t Ax$ é chamada uma matriz positiva (negativa) definida.

Definição 8: Uma forma quadrática $x^t Ax$ é positiva definida se - e somente se - todos os autovalores de A são positivos.

Definição 9: Dizemos que as matrizes quadradas A e B são similares, quando existe uma matriz T não-singular tal que $TAT^{-1}=B$. A transformação TAT^{-1} é chamada transformação de similaridade.

Definição 10: Uma matriz é dita não-derrogatória, quando - ela é similar à matriz companheira de seu polinômio característico.

Definição 11: As matrizes:

$$L = \begin{bmatrix} l_{11} & & & & \\ l_{21} & l_{22} & & & \\ \vdots & \ddots & \ddots & & \\ l_{n1} & \cdots & \cdots & l_{nn} & \end{bmatrix} \quad U = \begin{bmatrix} u_{11} & u_{12} & \cdots & u_{1n} \\ & \ddots & & \\ & & \ddots & \\ & & & u_{nn} \end{bmatrix}$$

são chamadas respectivamente, triangular inferior e triangular superior.

Definição 12: A matriz:

$$T = \begin{bmatrix} t_{11} & t_{12} & & & \\ t_{21} & t_{22} & \ddots & & \\ \vdots & \ddots & \ddots & \ddots & \\ & & & t_{n-1,n} & \\ 0 & & \ddots & \ddots & \\ & & & t_{n,n-1} & t_{nn} \end{bmatrix}$$

é chamada matriz tridiagonal.

Definição 13: A matriz:

$$V = \begin{bmatrix} 1 & 1 & \cdots & 1 \\ \lambda_1 & \lambda_2 & \cdots & \lambda_n \\ \lambda_1^2 & \lambda_2^2 & \cdots & \lambda_n^2 \\ \vdots & \vdots & & \vdots \\ \lambda_1^{n-1} & \lambda_2^{n-1} & \cdots & \lambda_n^{n-1} \end{bmatrix}$$

é dita matriz de Vandermonde.

Definição 14: A matriz:

$$D = dg(d_{11}, d_{22}, \dots, d_{nn}) = \begin{bmatrix} d_{11} & & & \\ & \ddots & & \\ & & d_{22} & \dots d_{nn} \\ & & & \ddots \end{bmatrix}$$

é chamada matriz diagonal.

Definição 15: As matrizes:

$$A = \begin{bmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ a_{31} & a_{32} & \dots & a_{3n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n,n-1} & a_{nn} \end{bmatrix} \text{ e } B = \begin{bmatrix} b_{11} & b_{12} & & \\ b_{21} & b_{22} & \ddots & \\ \vdots & \vdots & \ddots & b_{n-1,n} \\ b_{n1} & b_{n2} & \dots & b_{nn} \end{bmatrix}$$

são chamadas respectivamente, de Hessenberg superior e inferior.

Definição 16: A matriz real P é dita ortogonal se e somente se $P = P^t = P^{-1}$.

Notações:

I - Matriz identidade;

$[0]$ - Matriz nula;

A^t - Transposta da matriz A ;

A^{-1} - Inversa da matriz A ;

$tr(C)$ - Traço da matriz C

$L_1(x) = ||x||_1 = |x_1| + |x_2| + \dots + |x_n|$, para $x \in \mathbb{R}^n$

CAPÍTULO 1

MÉTODO PROPOSTO:

1.1. Introdução:

Seja a equação matricial:

$$Cx = \lambda x \quad (3)$$

onde C é uma matriz companheira real e x um vetor coluna.

Tomando a simetrizadora X de C e premultiplicando em (3), teremos:

$$XCx = \lambda Xx.$$

Chamando $A = XC$ e $B = X$, então,

$$Ax = \lambda Bx \quad (4)$$

onde A e B são matrizes simétricas reais.

Quando B é uma matriz positiva definida, o problema $Ax = \lambda Bx$ pode ser reduzido ao problema simétrico padrão $Ez = \lambda z$, onde E é uma matriz simétrica, usando-se a fatorização de Cholesky em B , da seguinte maneira [3,4]:

Se L é uma matriz triangular inferior tal que $LL^t = B$, então, podemos escrever a equação (4) como:

$$\begin{aligned} Ax &= \lambda L L^t x \text{ ou} \\ L^{-1} Ax &= \lambda L^t x \text{ ou ainda} \\ L^{-1} A (L^t)^{-1} L^t x &= \lambda L^t x. \end{aligned}$$

Chamando $E = L^{-1} A (L^t)^{-1}$ e $z = L^t x$ temos finalmente

$$Ez = \lambda z.$$

Logo E é uma matriz conhecida, real e simétrica.

Podemos, então, computar os autovalores λ de E , que correspondem àqueles dados pela equação (3). O caminho indicado pelo "Eispack Guide" [5, 6] é aquele em que usamos a combinação do método de Householder que transforma E numa matriz tridiagonal T , e, de uma variação racional do QL, que computa os autovalores de T .

A dificuldade que esse método poderia apresentar, seria na construção da matriz simetrizadora X , de forma que ela fosse positiva definida. Tal problema é evitado, usando-se as matrizes de Hankel, que são definidas por relações recursivas simples.

1.2. Matrizes de Hankel:

Seja $f(x) = x^n + c_n x^{n-1} + c_{n-1} x^{n-2} \dots + c_2 x + c_1$, um polinômio real de grau n e seja C a matriz companheira de $f(x)$. Então, a matriz:

$$H = \begin{bmatrix} S_0 & S_1 & \cdots & S_{n-1} \\ S_{-1} & S_2 & \cdots & S_n \\ \vdots & \vdots & & \vdots \\ S_{n-1} & S_n & \cdots & S_{2n-2} \end{bmatrix} \quad (5)$$

onde $S_i = \text{tr}(C^i)$ (Somas de Newton) é chamada de matriz Hankel das somas de Newton, associada a $f(x)$.

Para mostrarmos que a construção de H é fácil, usaremos um outro tipo de matriz Hankel [2] , que num caso particular, é igual à matriz Hankel das somas de Newton, com a vantagem de ter relações simples, definindo-a.

Sejam:

$$f(x) = c_{n+1}x^n + c_nx^{n-1} + c_{n-1}x^{n-2} + \cdots + c_2x + c_1, \quad (6)$$

$$(c_{n+1} = 1)$$

e

$$g(x) = b_{m+1}x^m + b_mx^{m-1} + b_{m-1}x^{m-2} + \cdots + b_2x + b_1, \quad (7)$$

dois polinômios de grau n e m , com coeficientes reais; $m \leq n$. Então, as quantidades s_i , $i = -1, 0, 1, 2, \dots$, definidas por:

$$\frac{g(x)}{f(x)} = s_{-1} + \frac{s_0}{x} + \frac{s_1}{x^2} + \frac{s_2}{x^3} + \dots \quad (8)$$

são chamadas parâmetros de Markov associados à função racional $R(x) = g(x) / f(x)$; e as matrizes:

$$H_{kk} = \begin{bmatrix} s_0 & s_1 & \cdots & s_{k-1} \\ s_1 & s_2 & \cdots & s_k \\ \vdots & \vdots & \ddots & \vdots \\ s_{k-1} & s_k & \cdots & s_{2k-2} \end{bmatrix}, \quad (k = 1, 2, \dots, n)$$

são as matrizes de Hankel, associadas aos parâmetros de Markov.

Existem relações recursivas simples para a geração dos coeficientes das matrizes de Hankel. Para o caso $n = m$, em (8), se livrarmos do denominador e equiparamos as potências de x em ambos os lados da equação, obteremos:

$$\begin{aligned} s_{-1} &= b_{n+1} \\ s_0 - c_n s_{-1} &= -b_n \\ s_1 - c_n s_0 - c_{n-1} s_{-1} &= -b_{n-1} \\ &\vdots \\ s_{n-1} - c_n s_{n-2} - \cdots - c_1 s_{-1} &= -b_1 \\ s_t - c_n s_{t-1} - \cdots - c_1 s_{t-n} &= 0 \quad (t = n, n+1, \dots, 2n-2). \end{aligned}$$

Agora, se $g(x) = f'(x)$, onde $f'(x)$ é a primeira derivada de $f(x)$, temos:

$$\begin{aligned}
 s_{-1} &= 0 \\
 s_0 &= n \\
 s_1 - c_n s_0 &= -(n-1) c_n \\
 s_2 - c_n s_1 - c_{n-1} s_0 &= -(n-2) c_{n-1} \\
 &\vdots \\
 s_{n-1} - c_n s_{n-2} - \cdots - c_2 s_0 &= -c_2 \\
 s_t - c_n s_{t-1} - \cdots - c_1 s_{t-n} &= 0 \quad (t = n, n+1, \dots, 2n-2)
 \end{aligned}$$

ou ainda:

$$\left. \begin{aligned}
 s_0 &= n = \text{tr}(C^0) \\
 s_1 &= c_n = \text{tr}(C^1) \\
 s_2 &= c_n^2 + 2 c_{n-1} = \text{tr}(C^2) \\
 s_3 &= c_n^3 + 3 c_n c_{n-1} + 3 c_{n-2} = \text{tr}(C^3) \\
 &\vdots \\
 \text{etc.}
 \end{aligned} \right\} \quad (9)$$

A relação (9) estabelece o fato da matriz Hankel das somas de Newton ser justamente igual à matriz Hankel dos parâmetros de Markov.

Em [1, p. 611] é estabelecido o seguinte lema, que forma uma parte da demonstração do nosso resultado:

Lema 1: A matriz Hankel H definida por (5) é tal que:

$$CH = HC^t \quad (10)$$

onde C é a matriz companheira de $f(x)$.

Demonstração: Seja:

$$C = \begin{bmatrix} 0 & 1 & 0 & \dots & 0 \\ 0 & 0 & 1 & 0 & \dots & 0 \\ \vdots & \ddots & \ddots & & & \vdots \\ \vdots & \ddots & \ddots & & & \vdots \\ \vdots & \ddots & \ddots & & & 1 \\ c_1 & c_2 & c_3 & \dots & c_n \end{bmatrix}$$

Considere-se agora, a equação matricial:

$$CX = XC^T. \quad (11)$$

Desde que C seja não-derrogatória, pelo resultado de Taussky e Zassenhaus [13], toda solução X de (11) é simétrica. Sejam x_1, x_2, \dots, x_n as n linhas da matriz X . Então, a equação matricial (11) é equivalente a:

$$x_i = x_{i-1}C^T, \quad i = 2, 3, \dots, n$$

$$c_1x_1 + c_2x_2 + \dots + c_nx_n = x_nC^T.$$

Eliminando-se x_2, x_3, \dots, x_n , obteremos:

$$x_1f(C) = [0].$$

Desde que, pelo teorema de Cayley-Hamilton [3,p.38-39], $f(C) = [0]$, x_1 pode ser escolhido arbitrariamente. Assim, as linhas x_1, x_2, \dots, x_n da solução simétrica X da equação matri-

cial (11) são tais que:

x_1 pode ser escolhida arbitrariamente

x_2, \dots, x_n satisfazem às relações recursivas

$$x_i = x_{i-1} C^t, \quad i = 2, 3, \dots, n.$$

Sejam h_1, h_2, \dots, h_n as n linhas da matriz H.

Então, das conhecidas identidades de Newton [14,p.39] ,

$$s_i = c_n s_{i-1} + c_{n-1} s_{i-2} + \dots + c_2 s_1 + c_1 s_0, \quad i = 1, 2, \dots, n$$

$$s_i = c_n s_{i-1} + c_{n-1} s_{i-2} + \dots + c_2 s_{i-n+1} + c_1 s_{i-n}, \quad (i > n)$$

é fácil ver-se que:

$$h_i = h_{i-1} C^t, \quad i = 2, 3, \dots, n.$$

Assim, da segunda até a n-ésima linha de H, são satisfeitas as mesmas relações recursivas de X dadas em (11). Desde que x_1 é arbitrária e, em particular, pode ser tomada como a primeira linha h_1 de H, o lema é provado.

Logo, H é uma simetrizadora de C^t , e, ao invés de trabalharmos com a matriz companheira C em (3), trabalharemos com a sua transposta C^t , que tem os mesmos autovalores [3, p. 3] .

Mostraremos, agora, que a matriz Hankel das somas de Newton H (5) é positiva definida, se e somente se os autovalores de C são reais e distintos.

A demonstração é dada, seguindo o artigo de Datta -
[2] :

Sejam $\lambda_1, \lambda_2, \dots, \lambda_n$ as raízes de $f(x)$. Então, de
(5) temos:

$$\begin{aligned}s_0 &= n = \sum_{i=1}^n \lambda_i^0 \\s_1 &= c_n = \sum_{i=1}^n \lambda_i^1 \\s_2 &= c_n^2 + 2 c_{n-1} = \sum_{i=1}^n \lambda_i^2 \\&\vdots \\&\text{etc.}\end{aligned}$$

A matriz H_{nn} portanto, pode ser escrita da forma:

$$H_{nn} = VV^t$$

onde V é a matriz de Vandermonde e a linha i de V é:

$$(\lambda_1^i, \lambda_2^i, \dots, \lambda_n^i), \quad 0 \leq i \leq n-1.$$

Se os λ_i são distintos e reais, a matriz de Vandermonde V é não-singular; portanto, H_{nn} é positiva definida.

Suponhamos, agora, que H_{nn} é positiva definida. Então,
poderemos escrever a equação $CH_{nn} = H_{nn}C^t$ da forma:

$$H_{nn}^{-1/2} C H_{nn}^{1/2} = H_{nn}^{1/2} C^t H_{nn}^{-1/2}.$$

Se:

$$P = H_{nn}^{-1/2} C H_{nn}^{1/2}$$

então,

$$P^t = H_{nn}^{1/2} C^t H_{nn}^{-1/2} = H_{nn}^{-1/2} C H_{nn}^{1/2} = P.$$

Portanto, P é simétrica. Também:

$$C = H_{nn}^{1/2} P H_{nn}^{-1/2}$$

é similar a P .

Desde que C é similar à matriz simétrica P , os autovalores de C são todos reais. Novamente, a não-singularidade de H_{nn} implica que a matriz de Vandermonde V é não-singular e portanto, os λ_i são distintos.

Assim, com a obtenção da matriz Hankel positiva definida e simetrizadora de C , o método proposto fica completo.

Observação: É interessante observar que, além de usarmos a matriz Hankel das somas de Newton para computar os autovalores reais e distintos de C , como descrevemos acima, poderemos usar essa matriz para obter informações sobre a localização dos autovalores de C , sem os mesmos serem computados. Pelo teorema 3 de [1] temos que os autovalores de C são distintos e reais negativos (positivos) se e somente se $-H$ é positiva definida e CH é negativa (positiva) definida.

1.3. Algoritmos usados:

1.3.1. Construção da matriz Hankel:

Esse algoritmo constrói a matriz Hankel das somas de Newton (5).

Os s_i são calculados recursivamente por:

$$s_0 = n$$

$$s_1 = c_n$$

$$s_2 = c_n s_1 + 2 c_{n-1}$$

$$s_3 = c_0 s_2 + c_{n-1} s_1 + 3 c_{n-2}$$

⋮

$$s_{n-1} = c_n s_{n-2} + c_{n-1} s_{n-3} + \dots + c_3 s_1 + (n-1) c_2$$

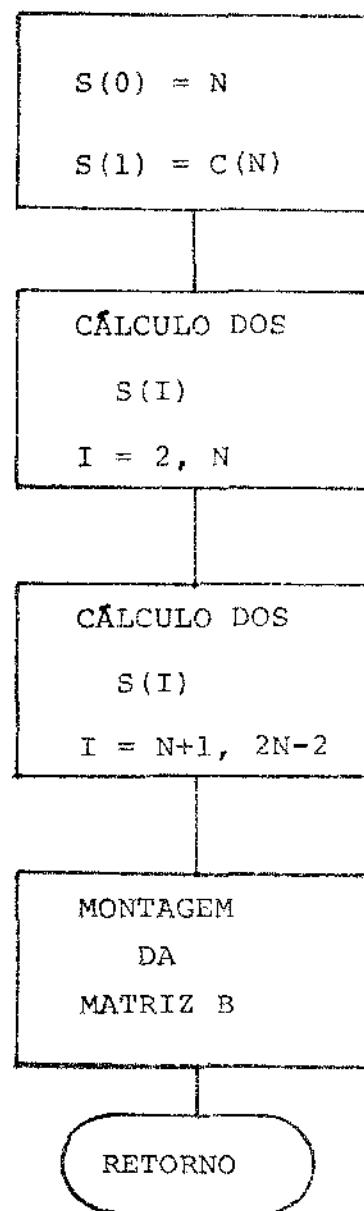
$$s_n = c_n s_{n-1} + c_{n-1} s_{n-2} + \dots + c_2 s_1 + n c_1$$

$$s_{n+1} = c_n s_n + c_{n-1} s_{n-1} + \dots + c_2 s_2 + c_1 s_1$$

⋮

$$s_{2n+2} = c_n s_{2n+3} + c_{n-1} s_{2n+4} + \dots + c_2 s_{n+1} + c_1 s_{n+2}$$

DIAGRAMA DE BLOCOS:



1.3.2. Construção da matriz A:

Esse algoritmo constrói a matriz A do problema simétrico generalizado $Ax = \lambda Bx$.

Essa matriz será real e simétrica; resultado do produto da matriz Hankel das somas de Newton H (5) pela transposta da matriz companheira C.

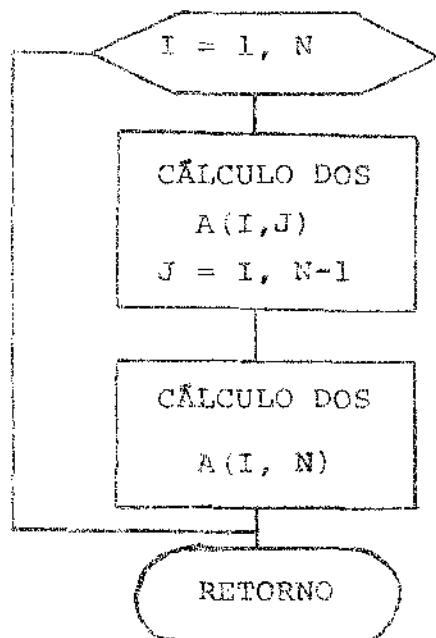
Os a_{ij} são calculados recursivamente por:

Para $i = 1, \dots, n$ temos:

$$(i) \quad a_{ij} = b_{i,j+1}, \quad j = i, \dots, n-1$$

$$(ii) \quad a_{in} = \sum_{k=1}^i b_{ki} c_k + \sum_{k=i+1}^n a_{i,k-1} c_k$$

DIAGRAMA DE BLOCOS:



1.3.3. Redução de $Ax = \lambda Bx$ para $Ez = \lambda x$:

Esse algoritmo reduz o problema de autovalores $Ax = \lambda Bx$ ao problema simétrico padrão $Ez = \lambda z$, conforme descrição no item 1.1.

A decomposição de Cholesky de B em LL^t , onde L é uma matriz triangular inferior é executada primeiro, do seguinte modo:

Como $\ell_{11} > 0$, temos:

(i) Para $k = 2, \dots, n$

$$\ell_{11} = \sqrt{a_{11}}$$

$$\ell_{k1} = \frac{b_{k1}}{\ell_{11}}$$

(ii) Para $i = 2, 3, \dots, n$

$$\ell_{ii} = \sqrt{b_{ii} - \sum_{j=1}^{i-1} \ell_{ij}^2}$$

$$\ell_{ki} = \frac{1}{\ell_{ii}} (b_{ki} - \sum_{j=1}^{i-1} \ell_{ij} \ell_{kj}), \quad k = i+1, \dots, n.$$

Depois, a matriz $E = L^{-1} A (L^{-1})^t$ é formada em dois passos:

$$LX = A \tag{12}$$

$$E L^t = X \tag{13}$$

onde as equações de (12) são dadas por:

Para $i = 1, \dots, n$

$$x_{ij} = \frac{a_{ji} - \sum_{k=1}^{i-1} x_{kj} l_{ik}}{l_{ii}}, \quad j = i, \dots, n,$$

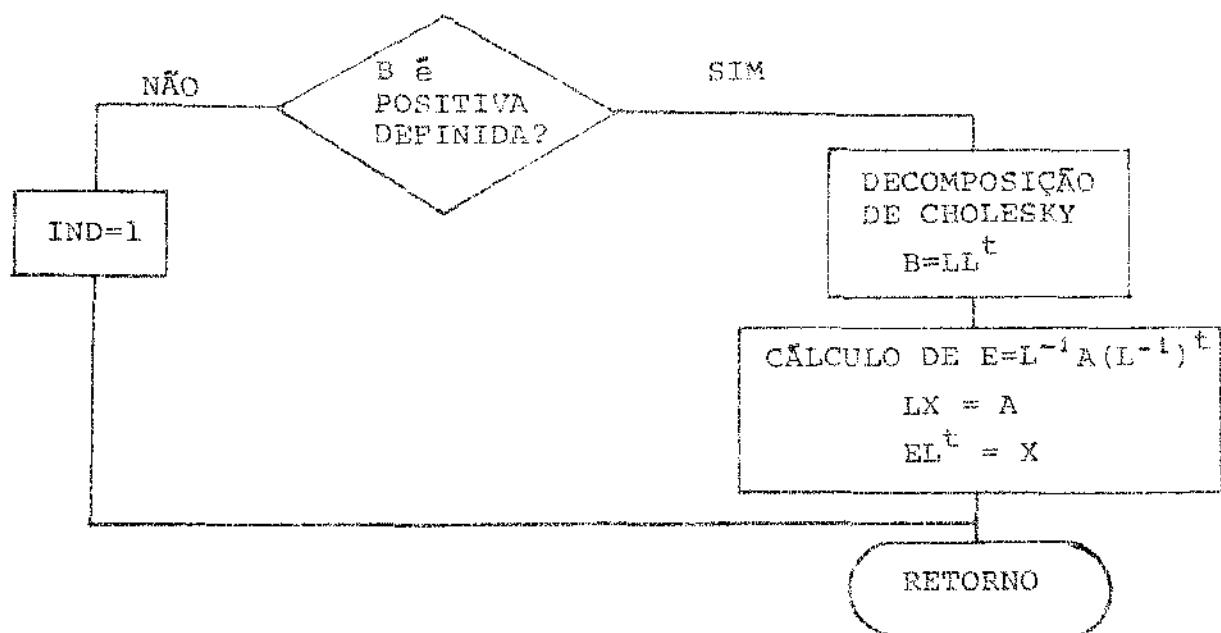
e as equações (13) são dadas por:

Para $i = 1, 2, \dots, n$

$$(i) e_{ij} = \frac{x_{ji} - \sum_{k=1}^{i-1} l_{ik} e_{jk}}{l_{ii}}, \quad j = i-1, i$$

$$(ii) e_{ij} = \frac{x_{ji} - \sum_{k=1}^{i-1} l_{ik} e_{kj}}{l_{ii}}, \quad j = 1, i-2.$$

DIAGRAMA DE BLOCOS:



1.3.4. Método de Householder:

Seja E uma matriz real e simétrica. Este método computa P tal que $P E P^t = T$, onde T é uma matriz tridiagonal e P é ortogonal.

Fazendo-se $E = A_1$, depois de $n-2$ transformações ortogonais, obteremos A_{n-1} tridiagonal.

Então,

$$A_{i+1} = P_i A_i P_i, \quad i = 1, 2, \dots, n-2$$

Seja w_i um vetor unitário ($\|w_i\|_2 = 1$ ou $w_i^t w_i = 1$) com n componentes, onde os i primeiros são zeros.

A matriz P_i é definida por

$$P_i = I - 2 w_i w_i^t = I - u_i u_i^t / H_i,$$

onde $H_i = \frac{1}{2} u_i^t u_i$ e u_i é um vetor definido por:

$$u_i^t = [a_{l,1}^{(i)}; a_{l,2}^{(i)}; \dots; a_{l,l-2}^{(i)}; a_{l,l-1}^{(i)} \pm \sigma_i^{1/2}; 0; \dots; 0]$$

com o sinal de σ_i sempre igual ao de $a_{l,l-1}^{(i)}$; $l = n-i+1$ e

$$\sigma_i = (a_{l,1}^{(i)})^2 + (a_{l,2}^{(i)})^2 + \dots + (a_{l,l-1}^{(i)})^2$$

$$p_i = A_i u_i / H_i$$

$$K_i = u_i^t p_i / 2 H_i$$

$$q_i = p_i - K_i u_i .$$

Finalmente,

$$A_{i+1} = (I - u_i u_i^t / H) A_i (I - u_i u_i^t / H)$$

$$A_{i+1} = A_i - u_i q_i^t = q_i u_i^t .$$

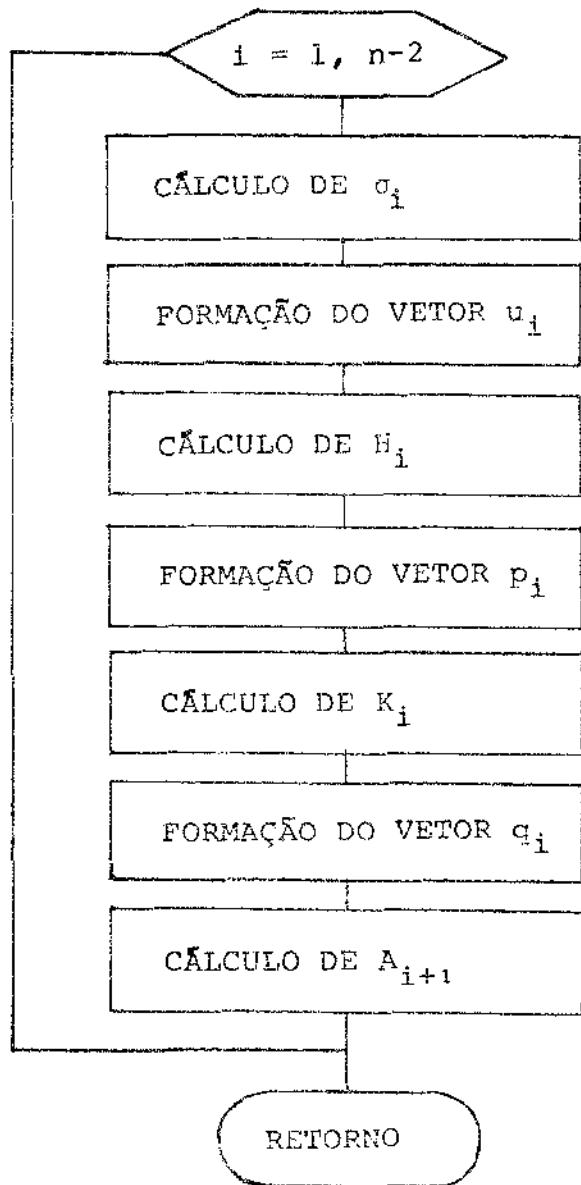
A matriz P_i da i -ésima transformação é da forma:

$$P_i = \begin{bmatrix} I - 2w_i w_i^t & | & 0 \\ \vdots & | & \vdots \\ 0 & | & I \end{bmatrix} \quad \begin{array}{l} \} n-i \\ \} i \end{array}$$

e a matriz A_i , para $n = 7$ e $i = 4$ tem a forma:

$$A_4 = \left[\begin{array}{cccc|cc} x & x & x & | & x & & \\ x & x & x & | & x & & \\ x & x & x & | & x & & \\ \hline - & - & - & | & - & - & - & - & - \\ x & x & x & | & x & x & & \\ & & & | & x & x & x & \\ & & & | & x & x & x & \\ & & & | & x & x & & \end{array} \right] \quad \begin{array}{l} \} n-i \\ \} i \end{array}$$

DIAGRAMA DE BLOCOS:



1.3.5. Variação racional do QL:

A essência desse método é um processo onde uma seqüência de matrizes tridiagonais reais e simétricas, unitariamente similares à matriz tridiagonal real e simétrica A_1 , é formada e converge para uma matriz diagonal, cujos elementos da diagonal principal são os autovalores de A_1 .

É definido pelas relações:

$$A_s = Q_s L_s$$

$$A_{s+1} = L_s Q_s = Q_s^T A_s Q_s$$

onde Q_s é ortogonal e L_s é triangular inferior.

O raio de convergência dessa seqüência é aperfeiçoadado, perturbando-se a matriz original a cada iteração. O processo torna-se:

$$A_s + k_s I = Q_s L_s ,$$

onde k_s é uma perturbação na s-ésima iteração.

$$A_{s+1} = L_s Q_s + k_s I$$

dando $A_{s+1} = Q_s^T A_s Q_s$

Ou ainda, de uma forma mais conveniente, para melhorar a convergência:

$$Q_s (A_s - k_s I) = L_s$$

$$A_{s+1} = L_s Q_s^t$$

(A_{s+1} é similar a $A_1 - \sum k_i I$ ao invés de A_1).

Antes das iterações, para cada autovalor, a matriz tridiagonal simétrica é checada para uma possível divisão em submatrizes, isto é, verificamos se existe um elemento ϵ na subdiagonal, de pequeno valor, para que a separação possa ser feita, como podemos ver ilustrada para $n = 7$ por:

$$\left[\begin{array}{ccccccc} x & x & & & & & \\ x & x & x & & & & \\ & x & x & x & & & \\ & & x & x & x & & \\ & & & x & x & \epsilon & \\ \hline & & & & \epsilon & x & x \\ & & & & & x & x \\ & & & & & & x & x \\ & & & & & & & x & x \end{array} \right] \quad + \text{linha } i$$

Se a divisão ocorre, apenas a submatriz que está mais em cima, à esquerda, participa na próxima iteração. Os autovalores vão sendo colocados em ordem crescente, conforme são achados. A perturbação na matriz original a cada iteração é um autovalor da atual submatriz 2×2 , que está mais em cima e, é próximo do primeiro elemento da diagonal dessa sub-

matriz. Sempre que a submatriz principal 1×1 , que está mais em cima, finalmente se separa do resto da matriz, seu elemento é um autovalor da matriz original e o algoritmo prossegue com a matriz restante.

Esse processo continua, até a matriz toda ter se dividido em submatrizes de ordem 1.

A matriz Q_s é obtida na forma fatorizada:

$$Q_s = p_1^{(s)} \ p_2^{(s)} \ \dots \ p_{n-1}^{(s)}$$

onde os $p_i^{(s)}$ são determinados na ordem $p_{n-1}^{(s)}, \dots, p_i^{(s)}$; os $p_i^{(s)}$ estão numa rotação no $(i, i+1)$ plano projetado, para exterminar o elemento na posição $(i, i+1)$.

Com uma pequena manipulação algébrica, mostraremos que uma simples iteração é definida da seguinte maneira:

Sejam $d_i^{(s)}$ ($i = 1, \dots, n$) os elementos da diagonal principal e $sd_i^{(s)}$ ($i = 1, \dots, n-1$) os elementos da subdiagonal superior.

Omitindo-se o sufixo s em tudo, menos em d e sd, temos:

$$p_n = d_n^{(s)} - k_s$$

$$c_n = 1$$

$$s_n = 0$$

para $i = n-1, \dots, 1$

$$r_{i+1} = (p_{i+1}^2 + (sd_i^{(s)})^2)^{1/2}$$

$$g_{i+1} = c_{i+1} sd_i^{(s)}$$

$$h_{i+1} = c_{i+1} p_{i+1}$$

$$sd_{i+1}^{(s+1)} = s_{i+1} r_{i+1}$$

(14)

$$c_i = p_{i+1} / r_{i+1}$$

$$s_i = sd_i^{(s)} / r_{i+1}$$

$$p_i = c_i (d_i^{(s)} - k_s) - s_i g_{i+1}$$

$$d_{i+1}^{(s+1)} = h_{i+1} + s_i (c_i g_{i+1} + s_i (d_i^{(s)} - k_s))$$

$$sd_1^{(s+1)} = s_1 p_1$$

$$d_1^{(s+1)} = c_1 p_1$$

É este o critério usado para decidir quando um elemento é insignificante: imediatamente antes de iterarmos para o r -ésimo autovalor,

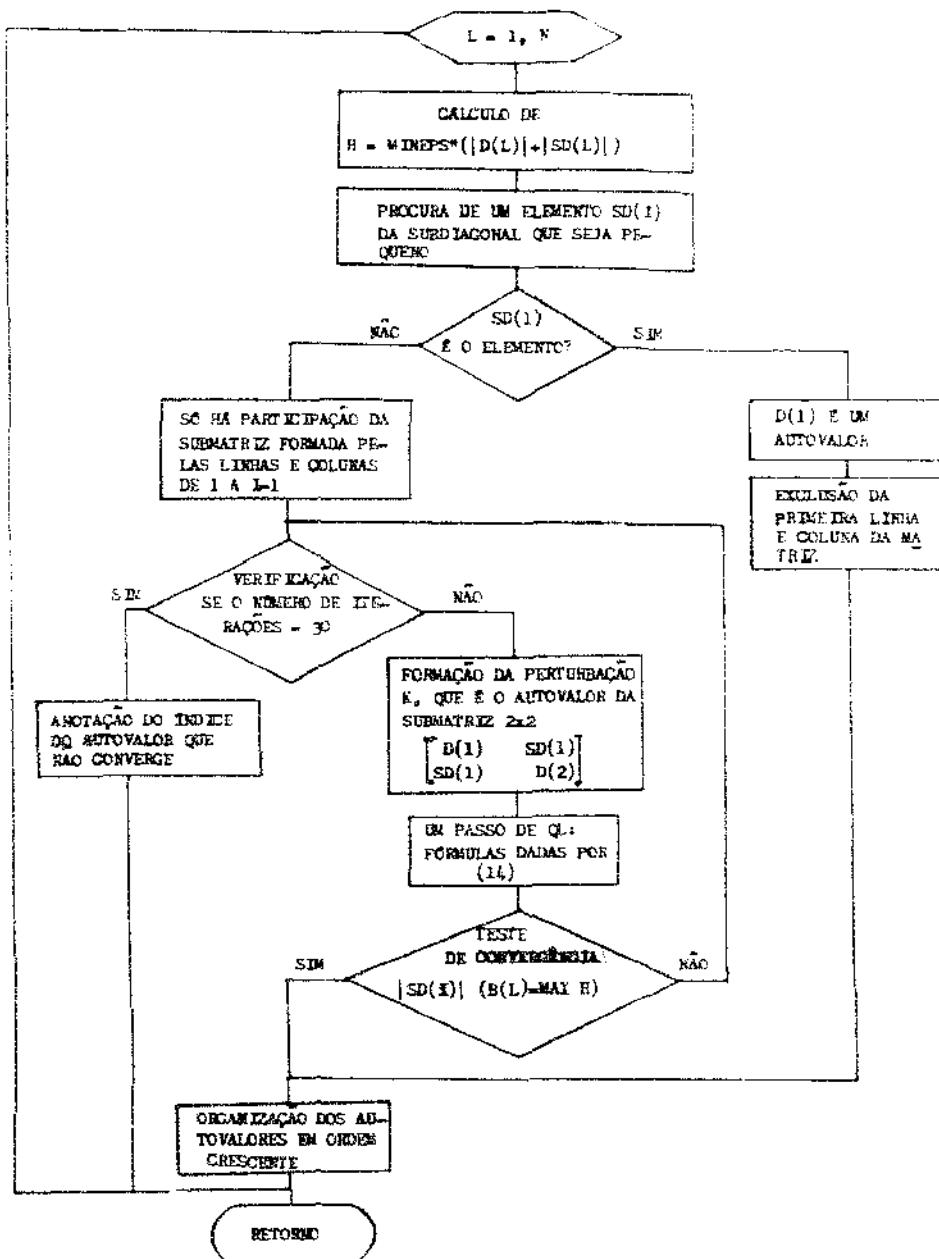
$$h_r = \text{mineps} (| d_r^{(s)} | + | sd_r^{(s)} |)$$

é computado (mineps é o menor ϵ tal que $1 + \epsilon > 1$) e antes de co-
meçarmos a iteração, algum $|sd_i^{(s)}|$ menor que $b_r = \max_{i=1,r} h_i$ é con-
siderado como insignificante.

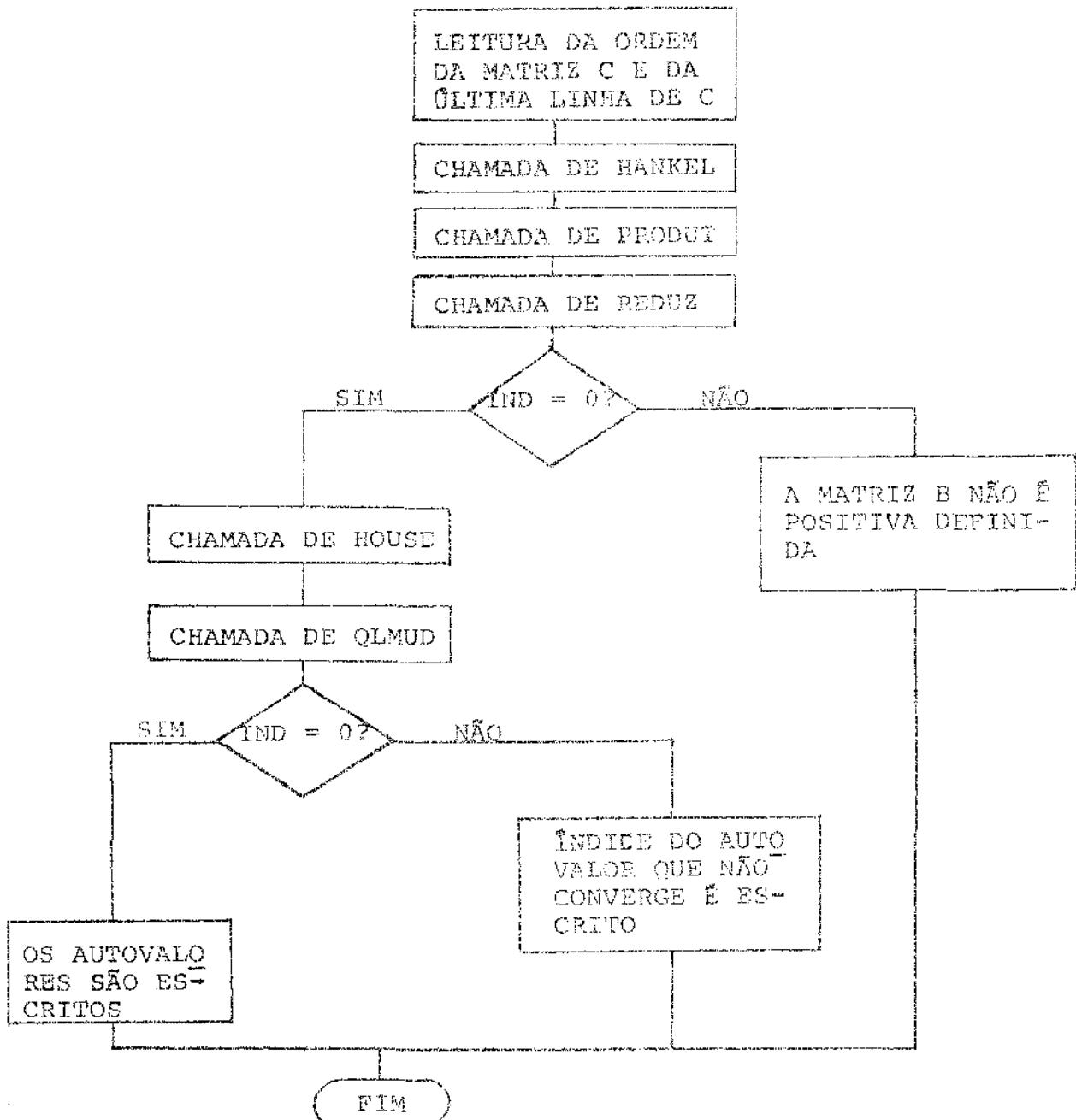
O número de iterações para cada autovalor é limitado ~
(arbitrariamente para 30).

O uso do QL é preferível ao convencional QR, pois as
matrizes tridiagonais resultantes de Householder têm os elemen-
tos maiores no canto inferior, à direita e isto faz com que se
torne necessário começar o algoritmo QR, trabalhando com a parte
de baixo para cima.

DIAGRAMA DE BLOCOS:



1.4. Diagrama de blocos:



1.5. Aplicação:

Computar os autovalores de $C^t x = \lambda x$ onde:

$$C^t = \begin{bmatrix} 0 & 0 & 2 \\ 1 & 0 & 2 \\ 0 & 1 & -1 \end{bmatrix}$$

Aplicando o método proposto:

(i) Construção da matriz Hankel:

$$H = \begin{bmatrix} s_0 & s_1 & s_2 \\ s_1 & s_2 & s_3 \\ s_2 & s_3 & s_4 \end{bmatrix} = \begin{bmatrix} 3 & -1 & 5 \\ -1 & 5 & -1 \\ 5 & -1 & 9 \end{bmatrix} = B$$

(ii) Construção da matriz $A = BC^t = HC^t$:

$$A = \begin{bmatrix} 3 & -1 & 5 \\ -1 & 5 & -1 \\ 5 & -1 & 9 \end{bmatrix} \cdot \begin{bmatrix} 0 & 0 & 2 \\ 1 & 0 & 2 \\ 0 & 1 & -1 \end{bmatrix} = \begin{bmatrix} -1 & 5 & -1 \\ 5 & -1 & 9 \\ -1 & 9 & -1 \end{bmatrix}$$

(iii) Redução de $Ax = \lambda Bx$ para $Ez = \lambda z$:

(a) Decomposição de $H = LL^t$ pelo método de Cholesky:

$$L = \begin{bmatrix} 1.7320508 & 0 & 0 \\ -0.5773502 & 2.1602469 & 0 \\ 2.8867513 & 0.3086065 & 0.7559292 \end{bmatrix}$$

(b) Cálculo de X de $LX = A$:

$$X = \begin{bmatrix} -0.5773502 & 2.8867513 & -0.5773502 \\ & 0.3086065 & 4.0118871 \\ & & -0.7559279 \end{bmatrix}$$

(c) Cálculo de E de $EL^t = X$:

$$E = \begin{bmatrix} -0.3333332 \\ 1.2472191 & 0.4761903 \\ -9.6173 \cdot 10^{-6} & 0.3499273 & -1.1426549 \end{bmatrix} = A_2$$

(iv) Método de Householder:

$$i = 1, \dots, n-2$$

$$\text{como } n = 3 \rightarrow i = 1$$

$$\sigma_1 = a_{31}^2 + a_{32}^2 = 0.1224491$$

$$u_1^t = (a_{31}; a_{32} \pm \sqrt{\sigma_1}; 0) = (-9.6173 \cdot 10^{-6}; 0.6996546; 0)$$

$$H_1 = \frac{1}{2} u_1^t u_1 = 0.2448982$$

$$p_1 = A_1 u_1 = \begin{bmatrix} 3.564224 \\ 1.360826 \\ 1.0000001 \end{bmatrix}$$

$$K_1 = u_1^T p_1 / 2 H_1 = 1.9444405$$

$$q_1 = p_1 - K_1 u_1 = \begin{bmatrix} 3.5642241 \\ 3.72 \cdot 10^{-7} \\ 1.0000001 \end{bmatrix}$$

$$u_1 q_1^T = \begin{bmatrix} -3.42782 \cdot 10^{-7} & -3.57763 \cdot 10^{-14} & -9.6173 \cdot 10^{-8} \\ 2.4944386 & 2.60345 \cdot 10^{-7} & 0.6998546 \\ 0 & 0 & 0 \end{bmatrix}$$

$$q_1 u_1^T = \begin{bmatrix} -3.42782 \cdot 10^{-7} & 2.4944386 & 0 \\ -3.57763 \cdot 10^{-14} & 2.60345 \cdot 10^{-7} & 0 \\ -9.6173 \cdot 10^{-8} & 0.6998546 & 0 \end{bmatrix}$$

$$A_2 = A_1 - u_1 q_1^T - q_1 u_1^T$$

$$A_2 = \begin{bmatrix} -0.3333325 & -1.2472195 & 0 \\ -1.2472195 & 0.4761897 & -0.3499273 \\ 0 & -0.3499273 & -1.1428549 \end{bmatrix} = T$$

(v) Variação racional do QL:

L: índice do autovalor que está sendo procurado;

IND: número de iterações para o autovalor de índice L;

I: índice do elemento pequeno da subdiagonal;

K: perturbação

MINEPS: $2 \cdot 10^{-24}$

L = 1

H = MINEPS * (|D(L)| + |SD(L)|) = $0.23550207 \cdot 10^{-7}$

IND = 1

I = 2 + submatriz atual formada pelas linhas e colunas de 1
a 3.

K = -1.239826

PASSO DE QL: fórmulas de (14)

SD(1) = |1.5817528|

D(1) = -0.02356389

IND = 2

I = 2 + submatriz atual formada pelas linhas e colunas de 1
a 3.

K = 1.319942

PASSO DE QL

SD(1) = |0.3084582|

D(1) = -0.07607795

IND = 3

I = 2 + submatriz atual formada pelas linhas e colunas de 1
a 3.

K = -1.413738

PASSO DE QL

SD(1) = |4.6807 $\cdot 10^{-3}$ |

D(1) = -0.476077*10⁻³

IND = 4

I = 2 → submatriz atual formada pelas linhas e colunas de I
a 3.

K = -1.414214

PASSO DE QL

SD(1) = |9.20869*10⁻¹⁰|

D(1) = -0.1396683*10⁻¹⁰

$\lambda_1 = -1.414214$

L = 2

H = 0.6228003*10⁻⁸

IND = 1

I = 2 → submatriz atual formada pelas linhas e colunas de 2
a 3.

K = -1

PASSO DE QL

SD(2) = |2.37402*10⁻⁷|

D(2) = 0.2355201*10⁻⁷

$\lambda_2 = -1$

L = 3

H = 0.3597459*10⁻⁷

I = 1 → submatriz formada pela linha e coluna 3

D(3) = 1.414214

$\lambda_3 = 1.414214$

CAPÍTULO 2

MÉTODO QR:

2.1. Introdução:

O algoritmo QR determina os autovalores de uma matriz qualquer. Uma seqüência de matrizes, unitariamente similar à matriz original, é formada. Tal seqüência converge para uma matriz triangular, cujos elementos na diagonal principal são os autovalores da matriz original.

Como estamos interessados nas matrizes companheiras reais, seguindo-se a indicação de "Eispack Guide" [5, 6], utilizaremos o seguinte caminho: primeiro, a matriz companheira será balanceada, isto é, as normas L_1 das linhas e colunas correspondentes da matriz original serão igualadas. Posteriormente, o método QR para matrizes de Hessenberg é aplicado.

2.2. Algoritmos usados:

2.2.1. Balanceamento de uma matriz:

Este algoritmo balanceia uma matriz real geral A e isolá autovalores, sempre que possível. As somas dos módulos dos elementos em linhas e colunas correspondentes serão quase iguais pelas transformações exatas de similaridade e, os autovalores serão isolados pelas transformações similares de permutação. O ba-

lanceamento reduz a norma 1 da matriz original, sempre que as somas dos módulos dos elementos em algumas linhas e correspondentes colunas, são marcadamente diferentes e, ao mesmo tempo, não altera os autovalores. Reduzindo a norma por esse caminho, podemos aperfeiçoar a exatidão dos autovalores computados. Adicionalmente, o tempo de execução do algoritmo é pequeno, comparado ao das rotinas que determinam os autovalores.

O método funciona do seguinte modo: primeiramente, determina-se o produto P da permutação das matrizes tal que:

$$P^t A P = \begin{bmatrix} T & X & Y \\ 0 & B & Z \\ 0 & 0 & R \end{bmatrix},$$

onde T e R são matrizes triangulares superiores e, B é uma matriz quadrada, formada pelas linhas e colunas LOW a IGH, com os elementos que estão fora da diagonal principal, diferentes de zero. X, Y e Z são matrizes retangulares, de dimensões apropriadas.

Os elementos da diagonal principal de T e R são os autovalores isolados de A.

Depois, a subrotina determina iterativamente a matriz diagonal D, não-singular, de ordem IGH - LOW + 1, tal que: $D^{-1}BD$ seja a matriz balanceada, com as somas dos módulos dos elementos das linhas e colunas correspondentes de $D^{-1}BD$, próximas de zero. Os elementos de D são potências exatas da base de representação do ponto aritmético flutuante do computador. Podemos represen-

tar a matriz final por:

$$\begin{bmatrix} T & XD & Y \\ 0 & D^{-1}BD & D^{-1}Z \\ 0 & 0 & R \end{bmatrix} .$$

O algoritmo pode ser descrito do seguinte modo: seja A_0 a matriz formada pelos elementos de A que não estão na diagonal principal. Para alguma matriz diagonal D não-singular,

$$D^{-1} A D = \text{diag}(A) + D^{-1} A_0 D,$$

e apenas A_0 é afetada. Assumimos que nem as linhas ou colunas de A_0 desaparecem.

De A_0 , uma seqüência $\{A_k\}$ é formada. O termo A_k difere de A_{k-1} apenas numa linha e correspondente coluna.

Seja $k = 1, 2, \dots$ e seja i o índice da linha e coluna modificada no passo de A_{k-1} a A_k . Então, se n é a ordem de A_0 , i é dado por:

$$i - 1 \equiv k - 1 \pmod{n}.$$

Então, as linhas são modificadas em ciclos, em suas ordens naturais. O k -ésimo passo é como se segue:

(a) Seja R_k e C_k as normas L_1 da linha i e coluna i de A_{k-1} . De acordo com a suposição acima, $R_k C_k \neq 0$. Portanto, se β é a base do computador, existe um único inteiro $\sigma = \sigma_k$ tal

que

$$\beta^{2(C-1)} < \frac{R_k}{C_k} \leq \beta^{2(C+1)} .$$

define $f = f_k$ por $f = \beta^0$.

(b) Para a constante dada $\gamma \leq 1$, tomamos:

$$\tilde{D}_k = \begin{cases} I + (f-1) e_i e_i^t & \text{se } (C_k * f) + \frac{R_k}{f} < \gamma * (C_k + R_k) \\ I & \text{e vice-versa} \end{cases}$$

onde $I = (e_1, \dots, e_n)$ é a matriz identidade.

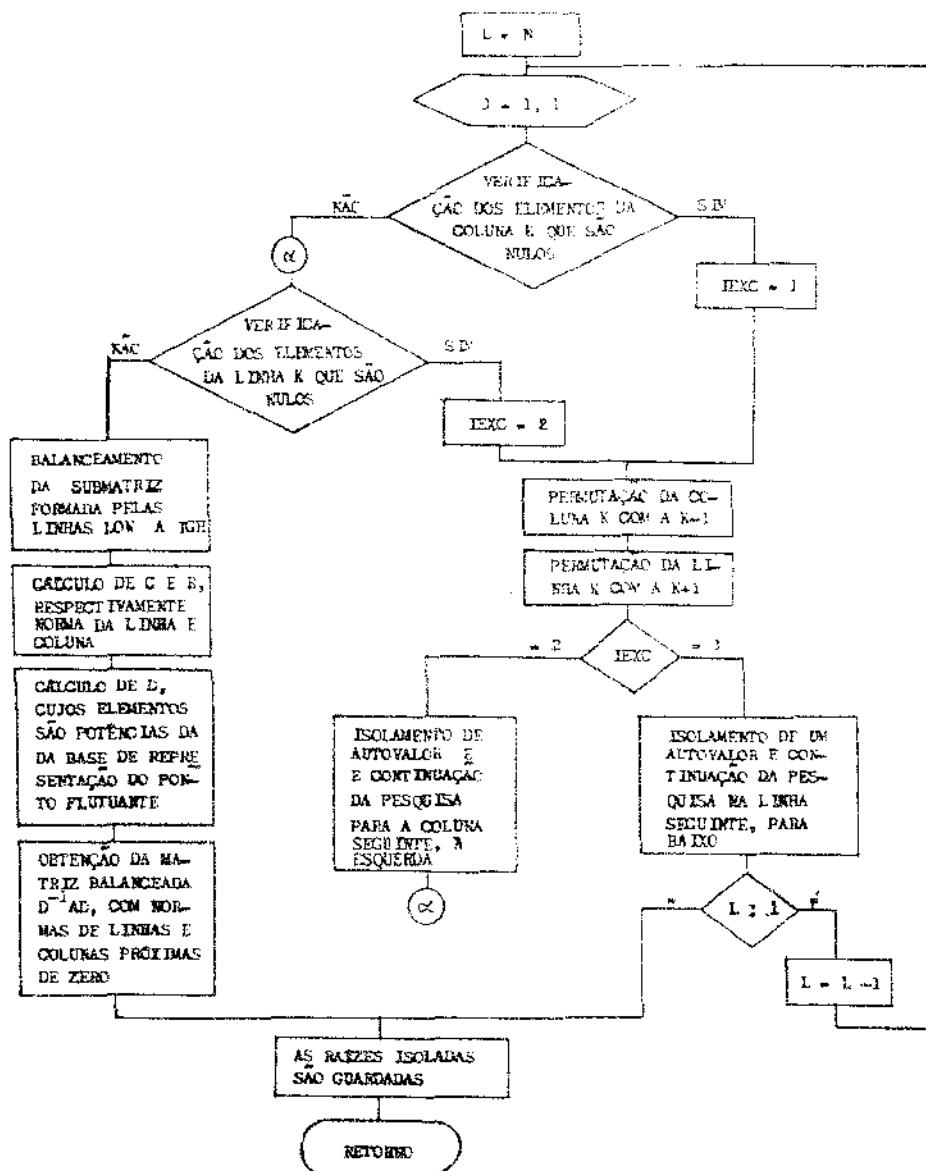
(c) Forma:

$$D_k = \tilde{D}_k D_{k-1} ; \quad D_0 = I ;$$

$$A_k = \tilde{D}_k^{-1} A_{k-1} \tilde{D}_k .$$

A iteração termina se, para um ciclo completo, $\tilde{D}_k \approx I$.

DIAGRAMA DE BLOCOS:



2.3. Algoritmo QR para matrizes reais de Hessenberg:

Seja A uma matriz real. O algoritmo QR é baseado na observação de que: se

$$A = QR \text{ e}$$

$$B = RQ,$$

onde Q é ortogonal e R triangular superior; então,

$$B = RQ = Q^T AQ,$$

isto é, B é unitariamente similar a A .

Por repetidas aplicações dos resultados acima, a seqüência de matrizes reais que é unitariamente similar à matriz dada A_1 , pode ser derivada das relações:

$$A_s = Q_s R_s,$$

$$A_{s+1} = R_s Q_s = Q_s^T A_s Q_s.$$

Quando a matriz inicial A_1 é da forma de Hessenberg superior, todo A_s também será dessa forma.

O volume de trabalho envolvido num passo QR é bem menor, se a matriz é da forma de Hessenberg.

O raio de convergência dessa seqüência é melhorado, pela perturbação na matriz original a cada iteração:

$$Q_s (A_s + k_s I) = R_s,$$

$$A_{s+} = R_s Q_s^T + k_s I = Q_s A_s Q_s^T ,$$

onde Q_s é ortogonal, R_s triangular superior e k_s é a perturbação na matriz original.

Pariett [18] mostrou que se todos k_s são zeros, então, em geral, A_s tende para um forma na qual:

$$a_{i+1,i}^{(s)} a_{i+2,i+1}^{(s)} = 0 \quad (i = 1, \dots, n-2) ,$$

ou seja,

Definição: Uma sequência $\{A^{(s)} = (a_{ij}^{(s)})\}$, $s = 1, 2, \dots$ de matrizes de Hessenberg $n \times n$, converge sempre que

$$a_{i+1,i}^{(s)} a_{i+2,i+1}^{(s)} = 0, \text{ para cada } j = 1, \dots, n-2.$$

Partindo-se desta definição, podemos enunciar o seguinte teorema:

Teorema: O algoritmo básico QR aplicado a matrizes Hessenberg irreductíveis A (isto é, $a_{ij} \neq 0$, $i = j+1$), produz uma seqüência de matrizes de Hessenberg, a qual converge se e somente se entre os conjuntos de autovalores de igual magnitude, existem no máximo dois de multiplicidade par e dois de multiplicidade ímpar.

Quando os autovalores são reais, a convergência é sempre garantida.

Portanto, todos os autovalores são isolados na diagonal principal ou são os autovalores da submatriz diagonal 2×2 . Para que haja uma convergência rápida, é essencial que as perturbações na matriz original sejam usadas e, que cada perturbação

k_s seja próxima de um autovalor de A_1 (e, portanto, de todo A_S).

Antes de cada iteração, a última forma de Hessenberg é verificada, para ver se existe algum elemento muito pequeno na subdiagonal, de modo que se torne possível a divisão da matriz em submatrizes.

O critério usado para se saber sobre a existência de elementos insignificantes é o seguinte:

$$|a_{l,l-1}| \leq \text{mineps}(|a_{l-1,l-1}| + |a_{l,l}|),$$

Isto é, compara o elemento da subdiagonal com os elementos da diagonal local.

Se a divisão ocorre, apenas a submatriz formada pelas linhas e colunas de l a n , que está mais em baixo, à direita, participará na próxima iteração, como podemos ver representada, quando $n = 7$ por:

$$A_S = \left[\begin{array}{ccc|cccc} x & x & x & | & x & x & x & x \\ x & x & x & | & x & x & x & x \\ x & x & | & x & x & x & x \\ \hline - & - & - & | & - & - & - & - \\ & \epsilon & | & x & x & x & x \\ & | & x & x & x & x \\ & | & x & x & x \\ & | & x & x \end{array} \right] + \text{linha } l$$

Sempre que uma submatriz 1×1 ou 2×2 finalmente se separa do resto da matriz, os autovalores dessa submatriz se

rão aqueles da matriz original e, o algoritmo prossegue, com o restante da matriz. Esse processo continua, até a matriz dividir-se completamente em submatrizes de ordem 1 ou 2.

A aritmética, no processo, é sempre mantida real, pela combinação de duas iterações numa única, aplicando-se duas perturbações reais na matriz original, ou um par de perturbações complexas conjugadas na matriz original.

Essas perturbações, a cada estágio, serão as duas raízes da matriz 2×2 , situadas no canto inferior, à direita, da atual A_s dadas por:

$$k_s + k_{s+1} = a_{n-1,n-1}^{(s)} + a_{nn}^{(s)}$$

$$k_s k_{s+1} = a_{n-1,n-1}^{(s)} - a_{n-1,n}^{(s)} a_{n,n-1}^{(s)}.$$

Householder [3, p. 290-293] mostrou que qualquer matriz real pode ser triangularizada por sucessivas pré-multiplicações de matrizes reais e simétricas elementares P_1, P_2, \dots, P_{n-1} , onde $P_s = I - 2 w_s w_s^t$ e, w_s é um vetor unitário, com elementos diferentes de zero, apenas nas posições $s, s+1, s+2$.

A matriz Q real pode, então, ser escrita da forma
$$Q = P_1 P_2 \dots P_{n-1}.$$

Podemos reduzir o número de operações, usando a expressão $2 w_s w_s^t$, na forma $u_s v_s^t$, onde u_s e v_s são ambos paralelos a w_s e seus elementos diferentes de zero são dados respectivamente, por:

$$u_s = (p_s \pm \sigma_s) / (\pm \epsilon_s), q_s / (r \epsilon_s), r_s / (\pm \sigma_s)$$

é

$$v_s = 1, q_s / (p_s \pm \sigma_s), r_s / (p_s \pm \sigma_s),$$

com $\omega_s^2 = p_s^2 + q_s^2 + r_s^2$ e os elementos p_s, q_s, r_s serão definidos mais adiante.

Depois de determinar ℓ , a submatriz formada pelas linhas e colunas de ℓ a n é examinada e, se existirem dois elementos consecutivos da subdiagonal, suficientemente pequenos, haverá possibilidade de trabalharmos com uma submatriz menor e equilibrada. Podemos representar para $n = 7$ por:

$$A_s = \left[\begin{array}{cccccc|cccc} x & x & | & x & x & x & x & x \\ x & x & | & x & x & x & x & x \\ \vdots & & | & & & & & \\ x & x & | & x & x & x & x & x \\ \vdots & & | & & & & & \\ \epsilon_1 & | & x & x & x & x & x \\ \hline \ell & | & x & x & x & x & x \\ \vdots & | & x & x & x & x & x \\ \epsilon_2 & | & x & x & x & x & x \\ \vdots & | & x & x & x & x & x \\ \vdots & | & x & x & x & x & x \\ \vdots & | & x & x & x & x & x \end{array} \right] * \text{linha } m$$

Começamos, então, pela linha m , calculando os elementos p_m, q_m, r_m , definidos por:

$$p_m = a_{mm}^2 - a_{mm}(k_1 + k_2) + k_1 k_2 + a_{m,m+1} a_{m+1,m}$$

$$q_m = a_{m+1,m} (a_{mm} + a_{m+1,m+1} - k_1 - k_2)$$

$$r_m = a_{m+2,m+1} - a_{m+1,m}$$

e deles, determinamos a apropriada matriz P_m , que irá gerar elementos diferentes de zero, nas posições $(m+1, m-1), (m+2, m-1) \dots$, quando aplicada à matriz cheia. Podemos ilustrar por:

$$\left[\begin{array}{ccccccccc} x & x & | & x & x & x & x & x & x \\ x & x & | & x & x & x & x & x & x \\ \hline x & x & | & x & x & x & x & x & x \\ n_1 & | & x & x & x & x & x & x \\ n_2 & | & x & x & x & x & x & x \\ n_3 & | & x & x & x & x & x & x \\ | & & & x & x & x & x & x \\ | & & & x & x & x & x & x \\ | & & & x & x & x & x & x \end{array} \right]$$

A condição requerida é que n_2 e n_3 sejam insignificantes, o que nos leva para o critério

$$|a_{m,m-1}| * (|q_m| + |r_m|) \leq \text{mineps} * |p_m| * \\ * (|a_{m+1,m+1}| + |a_{mm}| + |a_{m-1,m-1}|).$$

O algoritmo QR, finalmente, reduz a matriz de Hessenberg A_i para a matriz de Hessenberg A_s , onde

$$(s) \quad (s) \\ a_{i+1,i} a_{i+1,i+1} = 0 \quad (i = 1, \dots, n-2).$$

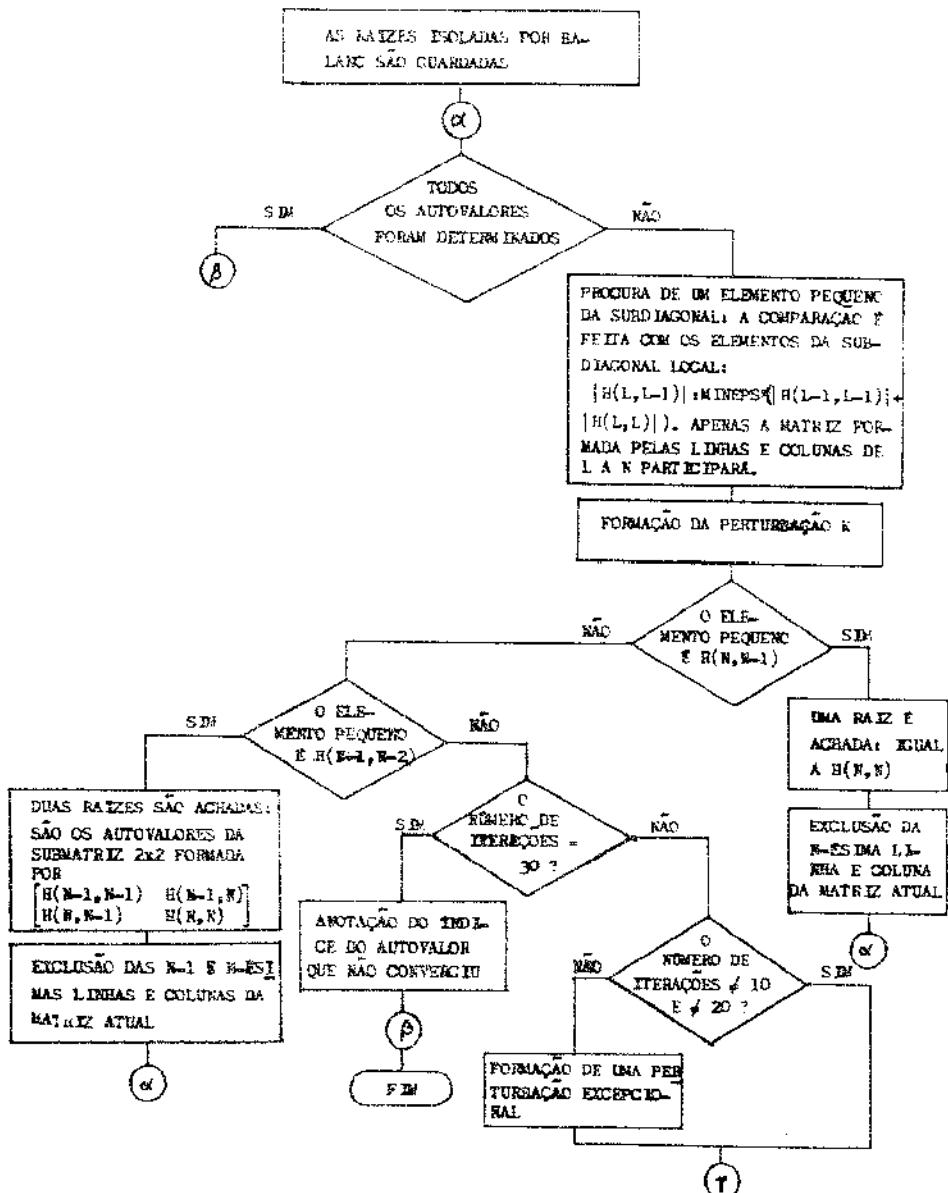
Se, depois de 10 iterações, nenhum autovalor for determinado, as perturbações k_1 e k_2 serão substituídas, na próxima i iteração, por perturbações definidas por:

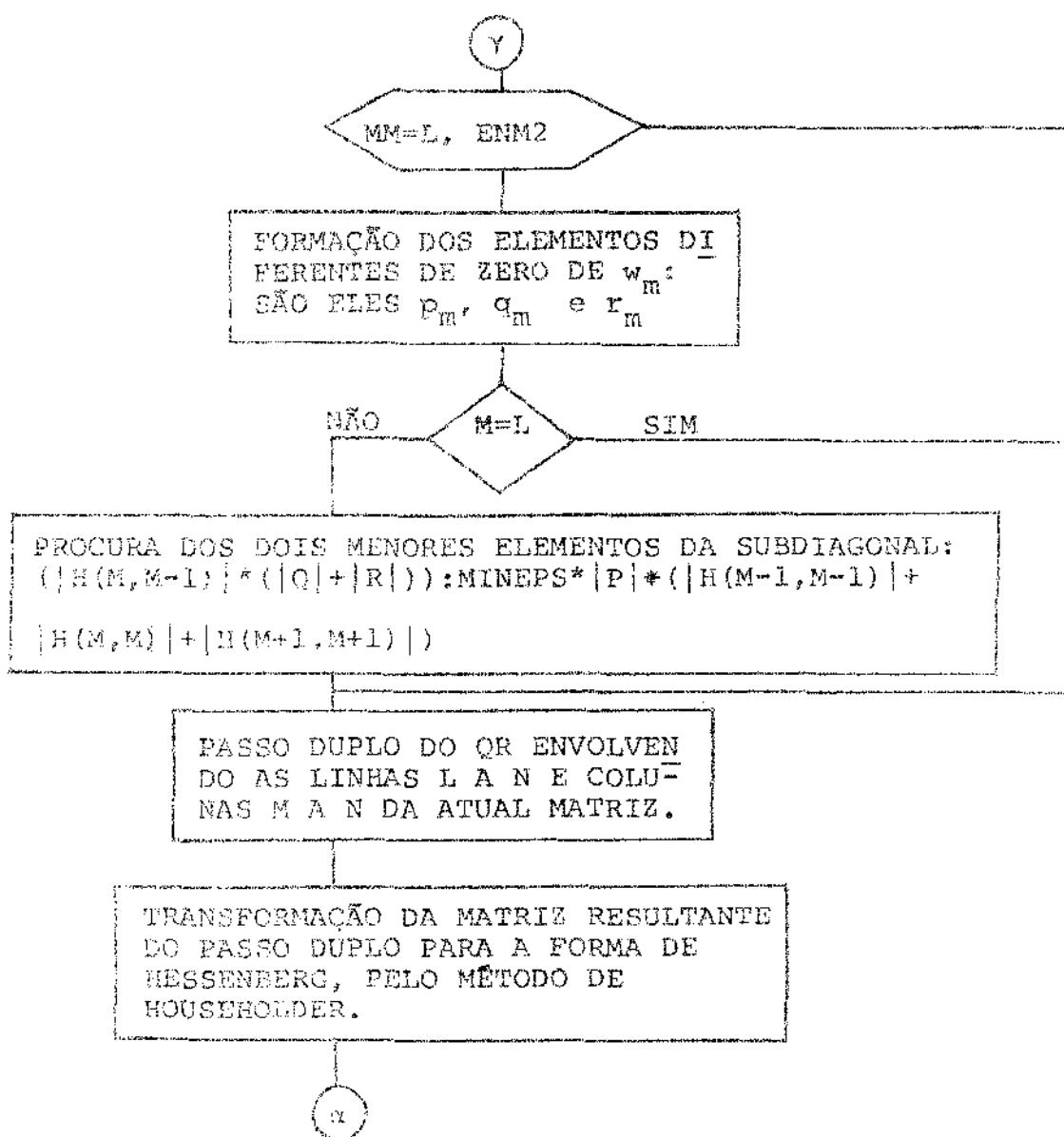
$$k_1 + k_2 = 1,5 * (|a_{n,n-1}| + |a_{n-1,n-2}|),$$

$$k_1 \cdot k_2 = (|a_{n,n-1}| + |a_{n-1,n-2}|)^2,$$

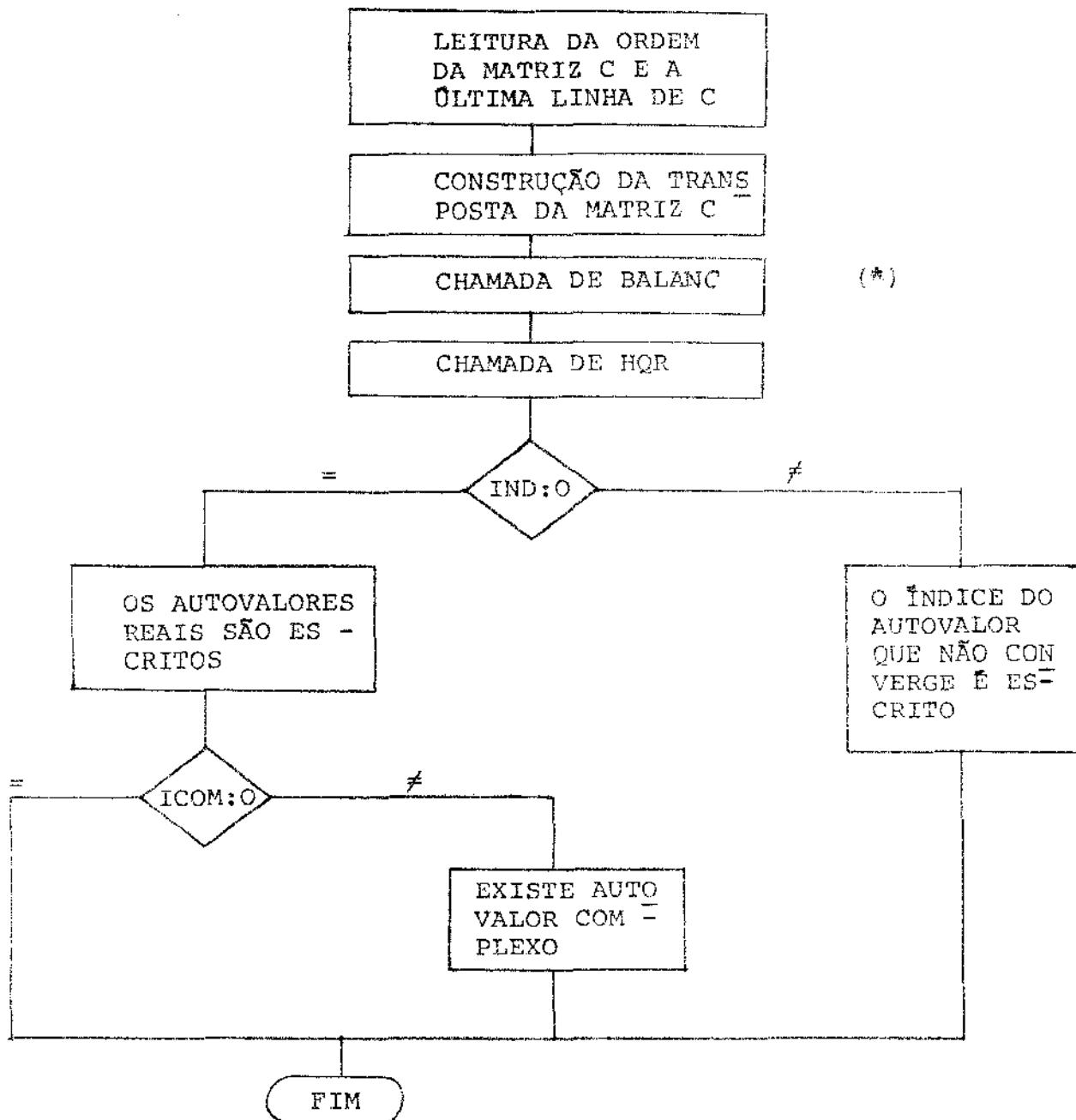
Essa estratégia é usada novamente depois de 20 iterações. Se 30 iterações são usadas, então, a indicação de falha de convergência é dada.

DIAGRAMA DE BLOCOS:





2.3. Diagrama de blocos:



(*) Podemos pular a subrotina BALANC, nomeando LOW=1 e IGH=N.

2.4. Aplicação:

Computar os autovalores de $C^T x = \lambda x$ onde

$$C^T = A_0 = \begin{bmatrix} 0 & 0 & 2 \\ 1 & 0 & 2 \\ 0 & 1 & -1 \end{bmatrix}.$$

Aplicando o método QR:

(i) Balanceamento da matriz C^T :

$$\ell = 2$$

$$\gamma = 0,95$$

$$C_i = \text{norma } L_1 \text{ da coluna } i$$

$$R_i = \text{norma } L_1 \text{ da linha } i$$

Toda linha e coluna tem um elemento não nulo. Portanto, não há permutação de linhas ou colunas.

A submatriz a ser balanceada é formada pelas linhas e colunas de 1 a 3.

$$C_1 = |A(2, 1)| + |A(3, 1)| = 1$$

$$R_1 = |A(1, 2)| + |A(1, 3)| = 2$$

$$\sigma = 0 \rightarrow f = 1$$

$$C_1 * f + \frac{R_1}{f} = 3 > \gamma * (C_1 + R_1) = 2,85$$

$$C_2 = |A(1, 2)| + |A(3, 2)| = 1$$

$$R_2 = |A(2, 1)| + |A(2, 3)| = 3$$

$$\sigma = 1 \rightarrow f = 2$$

$$C_2 * f + \frac{R_2}{f} = 3,5 < \gamma * (C_2 + R_2) = 3,8$$

2a. linha * $\frac{1}{f} = 0,5 (1; 0; 2) = (0,5; 0; 1)$

2a. coluna * f = 2

$$\begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 2 \end{bmatrix}$$

$$A_1 = \begin{bmatrix} 0 & 0 & 2 \\ 0.5 & 0 & 1 \\ 0 & 2 & -1 \end{bmatrix}$$

$$C_3 = |A(1, 3)| + |A(2, 3)| = 3$$

$$R_3 = |A(3, 1)| + |A(3, 2)| = 2$$

$$\sigma = 0 \rightarrow f = 1$$

$$C_3 * f + \frac{R_3}{f} = 5 < \gamma * (C_3 + R_3) = 4.75$$

$$C_1 = |A(2, 1)| + |A(3, 1)| = 0.5$$

$$R_1 = |A(1, 2)| + |A(1, 3)| = 2$$

$$\sigma = 1 \rightarrow f = 2$$

$$C_1 * f + \frac{R_1}{f} = 2 < \gamma * (C_1 + R_1) = 2.375$$

la. linha * $\frac{1}{2} = 0,5 \ (0; 0; 2) = (0; 0; 1)$

$$\text{la. coluna } * f = 2 \begin{bmatrix} 0 \\ 0,5 \\ 0 \end{bmatrix} = \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix}$$

$$A_2 = \begin{bmatrix} 0 & 0 & 1 \\ 1 & 0 & 1 \\ 0 & 2 & -1 \end{bmatrix}$$

$$C_2 = |A(1, 2)| + |A(3, 2)| = 2$$

$$R_2 = |A(2, 1)| + |A(2, 3)| = 2$$

$$c = 0 \rightarrow f = 1$$

$$C_2 * f + \frac{R_2}{f} = 4 > \gamma * (C_2 + R_2) = 3,8$$

$$C_3 = |A(1, 3)| + |A(2, 3)| = 2$$

$$R_3 = |A(3, 1)| + |A(3, 2)| = 2$$

$$c = 0 \rightarrow f = 1$$

$$C_3 * f + \frac{R_3}{f} = 4 > \gamma * (C_3 + R_3) = 3,8$$

$$C_1 = |A(1, 2)| + |A(1, 3)| = 1$$

$$R_1 = |A(2, 1)| + |A(3, 1)| = 1$$

$$c = 0 \rightarrow f = 1$$

$$C_1 * f + \frac{R_1}{f} = 2 > \gamma * (C_1 + R_1) = 1,9$$

$$C_2 = |A(1, 2)| + |A(3, 2)| = 2$$

$$R_2 = |A(2, 1)| + |A(2, 3)| = 2$$

$$\sigma = 0 + f = 1$$

$$C_2 * f + \frac{R_2}{f} = 4 > \gamma * (C_2 + R_2) = 3.8$$

$$C_3 = |A(1, 3)| + |A(2, 3)| = 2$$

$$R_3 = |A(3, 1)| + |A(3, 2)| = 2$$

$$C_3 * f + \frac{R_3}{f} = 4 > \gamma * (C_3 + R_3) = 3.8$$

$$C_{\text{BALANCEADA}}^t = \begin{bmatrix} 0 & 0 & 1 \\ 1 & 0 & 1 \\ 0 & 2 & -1 \end{bmatrix}$$

(ii) QR para matrizes reais de Hessenberg superior:

$$A_0 = \begin{bmatrix} 0 & 0 & 1 \\ 1 & 0 & 1 \\ 0 & 2 & -1 \end{bmatrix}$$

$$\text{MINEPS} = 10^{-8}$$

$$||A_0||_1 = 6$$

$$|a_{32}| = 2 > \text{MINEPS} * (|a_{33}| + |a_{22}|) = 10^{-8}$$

$$|a_{21}| = 1 > \text{MINEPS} * (|a_{22}| + |a_{11}|) = 0$$

$L = 1 \rightarrow$ a submatriz é formada pelas linhas e colunas de 1 a 3.

$$k_1 + k_2 = a_{22} + a_{33} = -1$$

$$k_1 \cdot k_2 = a_{22} \cdot a_{33} - a_{23} \cdot a_{32} = -2$$

1a. ITERAÇÃO:

$$p_1 = \frac{a_{11}^2 - a_{11}(k_1 + k_2) + k_1 k_2 + a_{12} a_{21}}{|p| + |q| + |r|} = -0,4$$

$$q_1 = \frac{a_{21}(a_{11} + a_{22} - k_1 - k_2)}{|p| + |q| + |r|} = 0,2$$

$$r_1 = \frac{a_{32} a_{21}}{|p| + |q| + |r|} = 0,4$$

PASSO DUPLO DO QR:

$$c_1 = (\text{sinal de } p_1) \sqrt{p_1^2 + q_1^2 + r_1^2} = 0,6$$

$$u_1 = \left(\frac{p_1 + c}{c}, \frac{q_1}{c}; \frac{r_1}{c} \right) = (1,666667; -0,333333; -0,666667)$$

$$v_1 = \left(1; \frac{q_1}{p_1 + c}; \frac{r_1}{p_1 + c} \right) = (1; -0,2; -0,4)$$

$$P_1 = I - u_1 v_1^t$$

$$A_1 = P_1 A_0 P_1 = \begin{bmatrix} -0.4444444 & 1.488889 & -0.6888889 \\ 0.2222222 & -0.1244445 & 1.684444 \\ 0.4444445 & 1.351111 & -0.4311111 \end{bmatrix}$$

$$p_2 = \frac{a_{2,1}}{|p| + |q| + |r|} = 0.3333333$$

$$q_2 = \frac{a_{3,1}}{|p| + |q| + |r|} = 0.6666667$$

$$r_2 = \frac{0}{|p| + |q| + |r|} = 0$$

$$\sigma_2 = (\text{sinal de } p_2) \sqrt{p_2^2 + q_2^2 + r_2^2} = 0.7453560$$

$$u_2 = \left(\frac{p_2 + \sigma}{\sigma}, \frac{q_2}{\sigma}, \frac{r_2}{\sigma} \right) = (1.447214; 0.8944272; 0)$$

$$v_2 = \left(1, \frac{q_2}{p_2 + \sigma}, \frac{r_2}{p_2 + \sigma} \right) = (1; 0.6180340; 0)$$

$$P_2 = I - u_2 v_2^t$$

$$A_2 = P_2 A_1 P_2 = \begin{bmatrix} -0.4444444 & -0.04969044 & -1.639783 \\ -0.4969040 & 0.8444445 & 0.8666666 \\ 0 & 1.2 & -1.4 \end{bmatrix}$$

$$|a_{32}| = 1.2 \times \text{MINEPS} * (|a_{33}| + |a_{22}|) = 2.244445 \times 10^{-9}$$

$$|a_{21}| = 0.4969040 \times \text{MINEPS} * (|a_{22}| + |a_{11}|) = 1.288869 \times 10^{-9}$$

L = 1 \rightarrow a submatriz é formada pelas linhas e colunas de 1 a 3.

Analogamente às fórmulas dadas acima, temos:

$$k_3 + k_4 = -0.5555555$$

$$k_3 * k_4 = -2.2222222$$

2a. ITERAÇÃO:

$$p_3 = 0.6771856$$

$$q_3 = 0.1431033$$

$$r_3 = 0.1797112$$

PASSO DUPLO DO QR:

$$\sigma_3 = 0.7150908$$

$$u_3 = (1.946992; 0.2001191; 0.2513123)$$

$$v_3 = (1; 0.1027837; 0.1290772)$$

$$P_3 = I - u_3 v_3^*$$

$$A_3 = P_3 A_2 P_3 = \begin{bmatrix} -0.8430799 & -0.5636437 & 1.555278 \\ -0.0895233 & 0.8377019 & 1.252939 \\ -0.1059307 & 1.128061 & -0.994622 \end{bmatrix}$$

$$p_4 = -0.4580274$$

$$q_4 = -0.5419726$$

$$r_4 = 0$$

$$\sigma_4 = -0.7095938$$

$$u_4 = (1.645478; 0.7637787; 0)$$

$$v_4 = (1; 0.4641682; 0)$$

$$P_4 = I + u_4 v_4^t$$

$$A_4 = P_4 A_3 P_4 = \begin{bmatrix} -0.8430799 & -0.8240687 & 1.434397 \\ 0.1386930 & 0.9426404 & 1.039376 \\ 0 & 1.164254 & -1.099561 \end{bmatrix}$$

$$|a_{32}| = 1.164254 > \text{MINEPS} * (|a_{33}| + |a_{22}|) = 2.042201 \cdot 10^{-6}$$

$$|a_{21}| = 0.1386930 > \text{MINEPS} * (|a_{22}| + |a_{11}|) = 1.785720 \cdot 10^{-8}$$

L = 1 + a submatriz é formada pelas linhas e colunas de 1 a 3.

$$k_s + k_s = -0.1569206$$

$$k_s * k_s = -2.2465883$$

3a. ITERAÇÃO:

$$p_5 = -0.9004537$$

$$q_5 = 0.01797077$$

$$r_5 = 0.08157557$$

PASSO DUPLO DO QR:

$$\sigma_5 = -0.9043198$$

$$u_5 = (1.995725; -0.01987214; -0.09020655)$$

$$v_5 = (1; -0.009957356; -0.04519990)$$

$$P_5 = I - u_5 v_5^t$$

$$A_5 = P_5 A_4 P_5 = \begin{bmatrix} -0.9557888 & 0.9622048 & -1.425528 \\ -0.00660777 & 0.9263119 & 1.074472 \\ 0.01020274 & 1.083464 & -0.9705231 \end{bmatrix}$$

$$p_5 = -0.3930739$$

$$q_6 = 0.6069261$$

$$r_6 = 0$$

$$\sigma_6 = -0.7230950$$

$$u_6 = (1.543599; -0.8393449; 0)$$

$$v_6 = (1; -0.5437583; 0)$$

$$P_6 = I - u_6 v_6^t$$

$$A_6 = P_6 A_5 P_6 = \begin{bmatrix} -0.9557888 & -1.719563 & 0.03270596 \\ 0.01215560 & -1.394604 & -0.4196699 \\ 0 & -0.4286622 & 1.350392 \end{bmatrix}$$

$$|a_{32}| = 0.4286622 > \text{MINEPS} * (|a_{33}| + |a_{22}|) = 2.744996 \cdot 10^{-8}$$

$$|a_{21}| = 0.0121556 > \text{MINEPS} * (|a_{22}| + |a_{11}|) = 2.350393 \cdot 10^{-8}$$

L = 1 → a submatriz é formada pelas linhas e colunas de 1 a 3.

$$k_7 + k_6 = -0.044212$$

$$k_7 * k_8 = -2.06315925$$

4a. ITERAÇÃO:

$$p_7 = -0.9733203$$

$$q_7 = -0.02249788$$

$$r_7 = -0.004181801$$

PASSO DUPLO DO QR:

$$\sigma_7 = -0.9735893$$

$$u_7 = (1.999724; 0.02310818; 0.004295241)$$

$$v_7 = (1; 0.01155569; 0.002147917)$$

$$P_7 = I - u_7 v_7^t$$

$$A_7 = P_7 A_6 P_7 = \begin{bmatrix} -0.9953685 & 1.730615 & -0.0329813 \\ -1.354883 & -0.001124368 & -0.4204566 \\ -0.4212525 & -0.0001700617 & 1.350251 \end{bmatrix}$$

$$p_8 = -0.8686204$$

$$q_8 = -0.1313796$$

$$r_8 = 0$$

$$\sigma_8 = -0.8784998$$

$$u_8 = (1.988754; 0.14955; 0)$$

$$v_8 = (1; 0.07519783; 0)$$

$$P_s = I - u_s v_s^t$$

$$A_s = P_s A P_s = \begin{bmatrix} -0.9953685 & -1.706220 & -0.2914317 \\ 0.001137156 & -1.414212 & 0.001628423 \\ 0 & 0.002424318 & 1.414212 \end{bmatrix}$$

$$|a_{32}| = 0.002424318 > \text{MINEPS} * (|a_{33}| + |a_{22}|) = 2.833056 \cdot 10^{-5}$$

$$|a_{21}| = 0.001137156 > \text{MINEPS} * (|a_{22}| + |a_{11}|) = 2.414212 \cdot 10^{-5}$$

$L = I - a$ submatriz é formada pelas linhas e colunas de 1 a 3.

$$k_9 + k_{10} = -0.004632$$

$$k_9 * k_{10} = -2.00655016$$

5a. ITERAÇÃO:

$$p_9 = -0.9973243$$

$$q_9 = -0.002673017$$

$$r_9 = -0.2689366 \cdot 10^{-5}$$

PASSO DUPLO DO QR:

$$c_9 \approx -0.9973279$$

$$u_9 = (1.999996; 0.002680179; -0.26966572 \cdot 10^{-5})$$

$$v_9 = (1; 0.001340092; -0.1348288 \cdot 10^{-5})$$

$$P_s = I - u_9 v_9^t$$

$$A_s = P_s A P_s = \begin{bmatrix} -0.9999407 & 1.707342 & 0.2914327 \\ -1.414271 & -0.1441242 \cdot 10^{-5} & 0.002409517 \\ 0.002419716 & 0.1232914 \cdot 10^{-7} & 1.414211 \end{bmatrix}$$

$$p_{10} = -0.9991453$$

$$q_{10} = 0.8547211 \cdot 10^{-3}$$

$$r_{10} = 0$$

$$c_{10} = -0.9991456$$

$$u_{10} = (2; -0.855452 \cdot 10^{-3}; 0)$$

$$v_{10} = (1; -0.4277261 \cdot 10^{-3}; 0)$$

$$P_{10} = I - u_{10} v_{10}^t$$

$$A_{10} = P_{10} A_9 P_{10} = \begin{bmatrix} -0.9999407 & -1.707093 & 0.2928932 \\ 0.1441243 \cdot 10^{-4} & -1.414273 & 0.1011696 \cdot 10^{-4} \\ 0 & -0.823347 \cdot 10^{-7} & 1.414214 \end{bmatrix}$$

$$|a_{32}| = 0.823347 \cdot 10^{-7} > \text{MINEPS} * (|a_{33}| + |a_{22}|) = 2.828487 \cdot 10^{-9}$$

$$|a_{21}| = 0.1441243 \cdot 10^{-4} > \text{MINEPS} * (|a_{22}| + |a_{11}|) = 2.414214 \cdot 10^{-9}$$

L = 1 → a submatriz é formada pelas linhas e colunas de 1 a 3.

6a. ITERAÇÃO:

$$p_{11} = -0.9999652$$

$$q_{11} = -0.3478264 \cdot 10^{-4}$$

$$r_{11} = 0.1186262 \cdot 10^{-11}$$

PASSO DUPLO DO QR:

$$c_{11} = -0.9999652$$

$$u_{11} = (2; 0.3478385 \cdot 10^{-4}; 0.1186303 \cdot 10^{-11})$$

$$v_{11} = (1; 0.1739193 \cdot 10^{-4}; 0.5931517 \cdot 10^{-12})$$

$$P_{11} = I = u_{11} v_{11}^t$$

$$A_{11} = P_{11} A_{10} P_{11} = \begin{bmatrix} -1 & 1.707107 & -0.2928932 \\ -0.2420165*10^{-6} & -1.414214 & -0.7099618*10^{-7} \\ -0.704189*10^{-16} & -0.8233271*10^{-7} & 1.414214 \end{bmatrix}$$

$$P_{12} = -1$$

$$q_{12} = -0.2909674$$

$$x_{12} = 0$$

$$c_{12} = -1$$

$$u_{12} = (2; 0.2909674*10^{-7}; 0)$$

$$v_{12} = (1; 0.1454837*10^{-7}; 0)$$

$$P_{12} = I = u_{12} v_{12}^t$$

$$A_{12} = P_{12} A_{11} P_{12} = \begin{bmatrix} -1 & -1.707107 & -0.2928932 \\ 0.2420165*10^{-6} & -1.414214 & -0.1130184*10^{-7} \\ 0 & 0.3469358*10^{-10} & 1.414214 \end{bmatrix}$$

$$|a_{32}| = 0.3469358*10^{-10} < \text{MINEPS} * (|a_{33}| + |a_{22}|) = 2.628427*10^{-8}$$

$$\boxed{\lambda_1 = 1.414214}$$

L = 2 → a submatriz é formada pelas linhas e colunas de 1 a 2.

$$|a_{21}| = 0.2420165*10^{-6} < \text{MINEPS} * (|a_{22}| + |a_{11}|) = 2.414214*10^{-8}$$

$$\boxed{\lambda_2 = -1.414214}$$

L = 1 → a submatriz é formada pela linha e coluna 1.

$$\boxed{\lambda_3 = -1}$$

CAPÍTULO 3

COMPARAÇÃO:

3.1. Experiência Computacional:

Os algoritmos foram programados em Fortran IV e, os testes foram feitos num computador PDP - 10.

As matrizes companheiras reais testadas, variam da ordem 3 a 10.

O tempo de execução do método proposto (DATTA) e do convencional QR com perturbação na matriz original, assim como o desvio médio dos autovalores, podem ser vistos na tabela 1, a seguir:

TABELA 1:

$C(N)$	MÉTODO	DESVIO MÉDIO	TEMPO DE EXECUÇÃO	CPU
$(2; 2; -1)$	DATTA	0	0	0.15
	QR(6)	0	0	0.19
$(6; -11; 6)$	DATTA	10^{-6}	0	0.12
	QR(5)	0	0	0.15
$(0; 2; -1)$	DATTA	10^{-6}	0	0.13
	QR(4)	0	0	0.13
$(2; 1; -3)$	DATTA	0	0	0.22
	QR(6)	0	0.1	0.25
$(1.5; 4.75; 3.25; -1)$	DATTA	10^{-6}	0.1	0.16
	QR(1)	10^{-7}	0.1	0.22
$(-24; 50; -33; 10)$	DATTA	10^{-5}	0	0.15
	QR(7)	10^{-6}	0	0.18
$(-5.2625; -35.6; -7.125;$ $6.25; 2.5)$	DATTA	10^{-1}	0	0.15
	QR(3)	10^{-1}	0.2	0.23
$(-720; 1764; -1624; 735;$ $-175; 21)$	DATTA	10^{-2}	0.1	0.17
	QR(1)	10^{-4}	0.2	0.32
$(-1.44 \times 10^{-5}; 3.6 \times 10^{-5}; 1.96 \times 10^{-5};$ $-4.9 \times 10^{-3}; -5.6 \times 10^{-3}; 0.14; 0.4)$	DATTA	10^{-6}	0.1	0.22
	QR(1)	10^{-7}	0.2	0.36
$(11.025 \times 10^{-5}; 0; -13.02625 \times 10^{-5}; 0;$ $21.0316 \times 10^{-2}; 0; -1.0374; 0; 184; 0)$	DATTA	10^{-4}	0.2	0.32
	QR(3)	10^{-7}	0.3	0.54

QR(I), I = número de iterações.

3.2. Conclusão:

A expectativa inicial com o método proposto do Capítulo 1, era a de se obter um algoritmo estável para a determinação dos autovalores reais e distintos de uma matriz companheira real. Porém, verificou-se, com surpresa, que esse método converge tão rapidamente quanto o convencional QR.

Em ambos os casos, os autovalores computados são próximos àqueles da matriz original; além disso, são os autovalores exatos da matriz próxima à matriz original. De modo que, podemos considerar os dois métodos numericamente estáveis.

A utilização do método proposto, portanto, é recomendável.

C CAPÍTULO 4:

C APÊNDICE:

C **4.1. HANKEL.**

C **SUBROTINA HANKEL.**

C ESSE SUBROTINA CONSTRUI A MATEZ DE HANKEL DAS SOMAS DE NEWTON.

C E UMA VARIABEL INTEIRA DE RETRADA.

C E O VETOR REAL DE ORDEN N QUE CONTEM OS ELEMENTOS DA ULTIMA LINHA DA MATEZ COMPARTILHA.

C E A MATEZ SIMETRICA REAL POSITIVA DEFINIDA DE GRADUA N, CONSTRUIDA A PARTIR DAS SOMAS DE NEWTON.

C APENAS O TRIANGULO SUPERIOR CHEIO DA MATEZ E FORNECIDO.

C E O VETOR REAL DAS SOMAS DE NEWTON DE ORDEN

C+1. CONSTRUINDO A PARTIR DAS LINHAS P CONSIVAS.

CALL HANKEL(N)

COMMON /BL1/A(100,100),B(100,100),C(100)

DIMENSION S(0/99)

CONSTRUCAO DAS SOMAS DE NEWTON.

S(0)=N

S(1)=C(0)

DO 200 I=2,N

CS=0.

L=N+1

DO 100 K=I-1,I,-1

L=L-1

CS=CS+C(L)*S(K)

S(I)=CS+I*C(N-I+1)

CONTINUE

J=2*N+2

DO 400 I=B+1,J

S(I)=0.

K=I

DO 300 L=N,1,-1

K=K-1

S(I)=S(I)+C(L)*S(K)

CONTINUE

MONTAGEM DA MATEZ DE HANKEL

DO 500 I=1,N

DO 500 J=I,N

B(I,J)=S(I+J-2)

RETURN

END

4.2. PROBLEMA:

SUBPROGRAMA PRINCIPAL

FEZ A SUBRUTINA CONSTRUIR A MATRIZ SIMETRICA REAL A
A PARTIR DO PRODUTO DE MATRIZ DE HANKEL E PELA
TRANSPOSTA DA MATRIZ COMPANHIERA REAL.

N P USA VARIÁVEL INTEIRA DE ENTRADA.

RECEBE MATRIZ SIMETRICA REAL DE ORDEM N APENAS O
TRIÂNGULO SUPERIOR DA MATRIZ E FORNECIDO.
É FEITA A MULTIPLICAÇÃO DA MATRIZ POSITIVA DEFINIDA DE
ORDEN N APENAS O TRIÂNGULO SUPERIOR DA MATRIZ
DEPÓSIO DA TRANSPOSTA.

O RÉTOR DEVE SER UMA MATRIZ QUE CONTÉM OS ELEMENTOS
DO TRIÂNGULO SUPERIOR DA MATRIZ COMPANHIERA.

SUBPROGRAMA SUBROUTINE

COM A IDENTIFICAÇÃO (1,1,1,0,0,0,0,0,0,0,0,0)

CONSTRUIU A MATRIZ SIMETRICA REAL A PARTIR DA MATRIZ
E PELA TRANSPOSTA DA MATRIZ COMPANHIERA.

MATRIZ A

1 1 1 1 1 1 1 1 1 1 1 1

2 1 1 1 1 1 1 1 1 1 1 1

3 1 1 1 1 1 1 1 1 1 1 1

4 1 1 1 1 1 1 1 1 1 1 1

5 1 1 1 1 1 1 1 1 1 1 1

6 1 1 1 1 1 1 1 1 1 1 1

7 1 1 1 1 1 1 1 1 1 1 1

8 1 1 1 1 1 1 1 1 1 1 1

9 1 1 1 1 1 1 1 1 1 1 1

10 1 1 1 1 1 1 1 1 1 1 1

11 1 1 1 1 1 1 1 1 1 1 1

12 1 1 1 1 1 1 1 1 1 1 1

13 1 1 1 1 1 1 1 1 1 1 1

14 1 1 1 1 1 1 1 1 1 1 1

15 1 1 1 1 1 1 1 1 1 1 1

16 1 1 1 1 1 1 1 1 1 1 1

17 1 1 1 1 1 1 1 1 1 1 1

18 1 1 1 1 1 1 1 1 1 1 1

19 1 1 1 1 1 1 1 1 1 1 1

20 1 1 1 1 1 1 1 1 1 1 1

21 1 1 1 1 1 1 1 1 1 1 1

22 1 1 1 1 1 1 1 1 1 1 1

23 1 1 1 1 1 1 1 1 1 1 1

24 1 1 1 1 1 1 1 1 1 1 1

25 1 1 1 1 1 1 1 1 1 1 1

26 1 1 1 1 1 1 1 1 1 1 1

27 1 1 1 1 1 1 1 1 1 1 1

28 1 1 1 1 1 1 1 1 1 1 1

29 1 1 1 1 1 1 1 1 1 1 1

30 1 1 1 1 1 1 1 1 1 1 1

31 1 1 1 1 1 1 1 1 1 1 1

32 1 1 1 1 1 1 1 1 1 1 1

33 1 1 1 1 1 1 1 1 1 1 1

34 1 1 1 1 1 1 1 1 1 1 1

35 1 1 1 1 1 1 1 1 1 1 1

36 1 1 1 1 1 1 1 1 1 1 1

37 1 1 1 1 1 1 1 1 1 1 1

38 1 1 1 1 1 1 1 1 1 1 1

39 1 1 1 1 1 1 1 1 1 1 1

40 1 1 1 1 1 1 1 1 1 1 1

41 1 1 1 1 1 1 1 1 1 1 1

42 1 1 1 1 1 1 1 1 1 1 1

C 4.3. PROGRAMA:

C SUBJETIVA 4000.
C FICA SUBJETIVA REDUZ O PROBLEMA SIMETRICO GENERALIZADO
C DE AUTOVALORES $AX = (LA + B)BX$, ONDE A E POSITIVA DEFINIDA,
C PRAO O PROBLEMA SIMETRICO PADRAO DE AUTOVALORES
C $EZ = (LA + B)B^{-1}Z$, USANDO A FATORIZACAO DE CHOLESKY EM B .
C SE Z FOR VARIAVEL INTEIRA DE ENTRADA.
C A E A MATRIZ SIMETRICA REAL DE ORDEM N DO SISTEMA.
C ALGUNS ALGORITMOS SOBRE O CHEIO DA MATRIZ PRECISA
C SER EXECUTADOS. NA SAIDA, O TRIANGULO INFERIOR DEPOIS
C DE CONSTRUCAO TRIANGULO INFERIOR CHEIO DA MATRIZ
C DE AUTOVALORES DA REDUCAO PARA A FORMA PADRAO.
C O PROGRAMA SUBJETIVA DEVE SER ESCRITO E INTERNO.
C B E A MATRIZ SIMETRICA, REAL, POSITIVA DEFINIDA DE TAMANHO
C $M \times M$ DA FORMA $B = D + E$, ONDE D E TRIANGULO SUPERIOR E CRES-
C CENDO, E E ETRIANGULO SIMETRICO. NA SAIDA, O TRIAN-
C GULO INFERIOR DE B CONSTRUIDO. O TRIANGULO INFERIOR DE
C B E A FORMA PADRAO EZ E. O TRIANGULO SUPERIOR CHEIO DE
C B E A FORMA PADRAO AX .

C SE Z FOR VARIAVEL INTEIRA DE ENTRADA, O PROGRAMA SUBJETIVA
C DEVE SER ESCRITO E INTERNO. FAZER OS TESTES DE VALIDADE
C DA CONSTRUCAO DA TRANSPOSTA DE INVOLTA NO TRIANGULO
C INFERIOR CHEIO DE A . NESTE CASO, A FATORIZACAO
C DE B DEVE SER FEITA POR ALGUMA ALTERNATIVA, INDEPENDENTEMENTE
C DA FORMA DE Z .

ALGORITMO SUBJETIVA(4)
SUBJETIVA(100,100), 5(100,100), BPL(1,1)
C(100,100)(100), BSL(100), BPD
CONSTRUCAO DE B NO TRIANGULO INFERIOR DE B E NO VETOR
BPD PELA FATORIZACAO DE CHOLESKY.
DO 100
DO 100 I=1,N
11=I+1
DO 100 J=1,I
B=H(I,J)
IF(I,J)=0 GO TO 20
DO 100 K=1,I
X=X+B(I,K)*B(J,K)
IF(J,N,E,TO GO TO 30
VERIFICA SE A E POSITIVA DEFINIDA
IF(A,LN,E,) GO TO 130
Y=SQR(TX)
DPL(I)=Y
GO TO 40
B(J,I)=X/Y
CONTINUE
CONSTRUCAO DA TRANSPOSTA DE INVOLTA NO TRIANGULO
INFERIOR CHEIO DE A .
DO 70 I=1,N

I1=I-1
Y=DPL(I)
DO 70 J=1,N
X=A(I,J)
IF(I.EQ.1) GO TO 60
DO 50 K=1,I1
X=X-B(I,K)*A(J,K)
B(J,I)=X/Y
CONTINUE
C
PTE=MULTIPLICACAO POR INV(L)
DO 120 J=1,N
J1=J-1
DO 120 I=J+1
X=A(I,J)
IF(I.EQ.1) GO TO 90
I1=I-1
DO 50 K=1,I1
X=X-A(K,I)*B(I,K)
B(J,K)=1.0
DO 100 I=1,J1
X=B(I,I)-X*A(I,K)
B(I,I)=X*B(I,I)
CONTINUE
GO TO 110
C
INDICACAO DE QUANDO B NAO E POSITIVA DEFINIDA.
I=0
NPO=1
IF(I.GT.0) GO TO 100

C. 4.5. 248018

RESUMO - O ESTUDO FIZO SUBSTÂNCIA DE PESQUISA OS AUTOVALORES DE UMA MATRIZ TRIDIAGÔNICA ELÉTRICA T, USANDO UMA VARIACAO RACIONAL DA MATRIZ.

DESENVOLVIMENTO DA MATEMÁTICA
DE UMA VETOR PODE SE DIVISAR A QUE SE ENTRADA COR-
RESPONDEM OS ELEMENTOS DA DIAGONAL PRINCIPAL DA MATRIZ
QUE, ASSIM, S. AS SAÍDAS, SÓ CONTEM OS AUTOVALORES
DESES VETORES, S. ORDEM CRESCENTE.

ESTA E A MELHOR TECNICA DE DISPOSAO E QUE CONTEM AS
MELHORES POSSIBILIDADES DE GUARDAR OS ELEMENTOS
DE FORMA QUE SE POSSA FAZER TRABALHOS DE CRIA-
CAO E MANUTENCAO MAIS FÁCIL. DEIXO ESTE POUCO USO.

ESTA ES UNA DE LAS TECNICAS DE GATOS. SE HABLA DE
QUE SE DEBEN COLOCAR ALGUNAS PIEDRAS PARA EL ENTRENAMIENTO
DE LOS GATOS, Y QUE TENDRA CON ELLOS UNA IN-
FLUENCIA POSITIVA EN SU ADAPTACION A NUEVA FAZ.

ARTIGO 1º Aprovam-se as alterações na data da abertura dos concursos para a realização das eleições de 1986, no sentido de que o resultado da votação seja divulgado no dia 20 de outubro de 1986.

ESTRATO DE VITELLO. O VITELLO PULIMENTO PASSA A SÉRICA E VITELLO.

$\frac{1}{n} \cdot \frac{1}{n} = \frac{1}{n^2}$

卷之三

• 100 •

卷之三

1. *Ensayo de compresión que está siendo determinado*

10. *Leucania* *luteola* (Hufnagel) *luteola* Hufnagel, 1808.

卷之三

卷之三

E (S, G, H)

卷之三

$$C = \delta * H$$

PKU

PEQUEN

DO DO M=L,N
DO DO M=L,N

1708D(1)

CONTINUE

```

4 IF(C.EQ.0.0) GO TO 50
5 IF(J.EQ.3+) GO TO 100
62 J+1
7 FOR I=1 TO N-1 DO 8
8 I=I+1
9 DO 10 I=1 TO N-1
10 G=U(I)
11 P=(D(I1)-G)/(2.*S)
12 R=SQR(P*(P+1.))
13 D(I)=S/(P+SIGN(R,P))
14 R=SQR(1.-P)
15 C=1.0-F*(C-G)/R
16 S=(1.-F)*R
17 U(I)=C*(1.-S)
18 C=C+(1.-S)*(1.-S)
19 S=S*(1.-S)
20 F=F+C*S
21 D(I)=D(I)+F
22 G=U(I)
23 IF(G.EQ.0.0) GO TO 24
24 D(I)=D(I)+F
25 C=C+(1.-S)*(1.-S)
26 S=S*(1.-S)
27 F=F+C*S
28 D(I)=D(I)+F
29 G=U(I)
30 IF(G.EQ.0.0) GO TO 31
31 D(I)=D(I)+F
32 C=C+(1.-S)*(1.-S)
33 S=S*(1.-S)
34 F=F+C*S
35 D(I)=D(I)+F
36 G=U(I)
37 IF(G.EQ.0.0) GO TO 38
38 D(I)=D(I)+F
39 C=C+(1.-S)*(1.-S)
40 S=S*(1.-S)
41 F=F+C*S
42 D(I)=D(I)+F
43 G=U(I)
44 IF(G.EQ.0.0) GO TO 45
45 D(I)=D(I)+F
46 C=C+(1.-S)*(1.-S)
47 S=S*(1.-S)
48 F=F+C*S
49 D(I)=D(I)+F
50 C=C+(1.-S)*(1.-S)
51 S=S*(1.-S)
52 F=F+C*S
53 D(I)=D(I)+F
54 G=U(I)
55 IF(G.EQ.0.0) GO TO 56
56 D(I)=D(I)+F
57 C=C+(1.-S)*(1.-S)
58 S=S*(1.-S)
59 F=F+C*S
60 D(I)=D(I)+F
61 G=U(I)
62 IF(G.EQ.0.0) GO TO 63
63 D(I)=D(I)+F
64 C=C+(1.-S)*(1.-S)
65 S=S*(1.-S)
66 F=F+C*S
67 D(I)=D(I)+F
68 G=U(I)
69 IF(G.EQ.0.0) GO TO 70
70 D(I)=D(I)+F
71 C=C+(1.-S)*(1.-S)
72 S=S*(1.-S)
73 F=F+C*S
74 D(I)=D(I)+F
75 G=U(I)
76 IF(G.EQ.0.0) GO TO 77
77 D(I)=D(I)+F
78 C=C+(1.-S)*(1.-S)
79 S=S*(1.-S)
80 F=F+C*S
81 D(I)=D(I)+F
82 G=U(I)
83 IF(G.EQ.0.0) GO TO 84
84 D(I)=D(I)+F
85 C=C+(1.-S)*(1.-S)
86 S=S*(1.-S)
87 F=F+C*S
88 D(I)=D(I)+F
89 G=U(I)
90 IF(G.EQ.0.0) GO TO 91
91 D(I)=D(I)+F
92 C=C+(1.-S)*(1.-S)
93 S=S*(1.-S)
94 F=F+C*S
95 D(I)=D(I)+F
96 G=U(I)
97 IF(G.EQ.0.0) GO TO 98
98 D(I)=D(I)+F
99 C=C+(1.-S)*(1.-S)
100 S=S*(1.-S)
101 F=F+C*S
102 D(I)=D(I)+F
103 G=U(I)
104 IF(G.EQ.0.0) GO TO 105
105 D(I)=D(I)+F
106 C=C+(1.-S)*(1.-S)
107 S=S*(1.-S)
108 F=F+C*S
109 D(I)=D(I)+F
110 G=U(I)
111 IF(G.EQ.0.0) GO TO 112
112 D(I)=D(I)+F
113 C=C+(1.-S)*(1.-S)
114 S=S*(1.-S)
115 F=F+C*S
116 D(I)=D(I)+F
117 G=U(I)
118 IF(G.EQ.0.0) GO TO 119
119 D(I)=D(I)+F
120 C=C+(1.-S)*(1.-S)
121 S=S*(1.-S)
122 F=F+C*S
123 D(I)=D(I)+F
124 G=U(I)
125 IF(G.EQ.0.0) GO TO 126
126 D(I)=D(I)+F
127 C=C+(1.-S)*(1.-S)
128 S=S*(1.-S)
129 F=F+C*S
130 D(I)=D(I)+F
131 G=U(I)
132 IF(G.EQ.0.0) GO TO 133
133 D(I)=D(I)+F
134 C=C+(1.-S)*(1.-S)
135 S=S*(1.-S)
136 F=F+C*S
137 D(I)=D(I)+F
138 G=U(I)
139 IF(G.EQ.0.0) GO TO 140
140 D(I)=D(I)+F
141 C=C+(1.-S)*(1.-S)
142 S=S*(1.-S)
143 F=F+C*S
144 D(I)=D(I)+F
145 G=U(I)
146 IF(G.EQ.0.0) GO TO 147
147 D(I)=D(I)+F
148 C=C+(1.-S)*(1.-S)
149 S=S*(1.-S)
150 F=F+C*S
151 D(I)=D(I)+F
152 G=U(I)
153 IF(G.EQ.0.0) GO TO 154
154 D(I)=D(I)+F
155 C=C+(1.-S)*(1.-S)
156 S=S*(1.-S)
157 F=F+C*S
158 D(I)=D(I)+F
159 G=U(I)
160 IF(G.EQ.0.0) GO TO 161
161 D(I)=D(I)+F
162 C=C+(1.-S)*(1.-S)
163 S=S*(1.-S)
164 F=F+C*S
165 D(I)=D(I)+F
166 G=U(I)
167 IF(G.EQ.0.0) GO TO 168
168 D(I)=D(I)+F
169 C=C+(1.-S)*(1.-S)
170 S=S*(1.-S)
171 F=F+C*S
172 D(I)=D(I)+F
173 G=U(I)
174 IF(G.EQ.0.0) GO TO 175
175 D(I)=D(I)+F
176 C=C+(1.-S)*(1.-S)
177 S=S*(1.-S)
178 F=F+C*S
179 D(I)=D(I)+F
180 G=U(I)
181 IF(G.EQ.0.0) GO TO 182
182 D(I)=D(I)+F
183 C=C+(1.-S)*(1.-S)
184 S=S*(1.-S)
185 F=F+C*S
186 D(I)=D(I)+F
187 G=U(I)
188 IF(G.EQ.0.0) GO TO 189
189 D(I)=D(I)+F
190 C=C+(1.-S)*(1.-S)
191 S=S*(1.-S)
192 F=F+C*S
193 D(I)=D(I)+F
194 G=U(I)
195 IF(G.EQ.0.0) GO TO 196
196 D(I)=D(I)+F
197 C=C+(1.-S)*(1.-S)
198 S=S*(1.-S)
199 F=F+C*S
200 D(I)=D(I)+F
201 G=U(I)
202 IF(G.EQ.0.0) GO TO 203
203 D(I)=D(I)+F
204 C=C+(1.-S)*(1.-S)
205 S=S*(1.-S)
206 F=F+C*S
207 D(I)=D(I)+F
208 G=U(I)
209 IF(G.EQ.0.0) GO TO 210
210 D(I)=D(I)+F
211 C=C+(1.-S)*(1.-S)
212 S=S*(1.-S)
213 F=F+C*S
214 D(I)=D(I)+F
215 G=U(I)
216 IF(G.EQ.0.0) GO TO 217
217 D(I)=D(I)+F
218 C=C+(1.-S)*(1.-S)
219 S=S*(1.-S)
220 F=F+C*S
221 D(I)=D(I)+F
222 G=U(I)
223 IF(G.EQ.0.0) GO TO 224
224 D(I)=D(I)+F
225 C=C+(1.-S)*(1.-S)
226 S=S*(1.-S)
227 F=F+C*S
228 D(I)=D(I)+F
229 G=U(I)
230 IF(G.EQ.0.0) GO TO 231
231 D(I)=D(I)+F
232 C=C+(1.-S)*(1.-S)
233 S=S*(1.-S)
234 F=F+C*S
235 D(I)=D(I)+F
236 G=U(I)
237 IF(G.EQ.0.0) GO TO 238
238 D(I)=D(I)+F
239 C=C+(1.-S)*(1.-S)
240 S=S*(1.-S)
241 F=F+C*S
242 D(I)=D(I)+F
243 G=U(I)
244 IF(G.EQ.0.0) GO TO 245
245 D(I)=D(I)+F
246 C=C+(1.-S)*(1.-S)
247 S=S*(1.-S)
248 F=F+C*S
249 D(I)=D(I)+F
250 G=U(I)
251 IF(G.EQ.0.0) GO TO 252
252 D(I)=D(I)+F
253 C=C+(1.-S)*(1.-S)
254 S=S*(1.-S)
255 F=F+C*S
256 D(I)=D(I)+F
257 G=U(I)
258 IF(G.EQ.0.0) GO TO 259
259 D(I)=D(I)+F
260 C=C+(1.-S)*(1.-S)
261 S=S*(1.-S)
262 F=F+C*S
263 D(I)=D(I)+F
264 G=U(I)
265 IF(G.EQ.0.0) GO TO 266
266 D(I)=D(I)+F
267 C=C+(1.-S)*(1.-S)
268 S=S*(1.-S)
269 F=F+C*S
270 D(I)=D(I)+F
271 G=U(I)
272 IF(G.EQ.0.0) GO TO 273
273 D(I)=D(I)+F
274 C=C+(1.-S)*(1.-S)
275 S=S*(1.-S)
276 F=F+C*S
277 D(I)=D(I)+F
278 G=U(I)
279 IF(G.EQ.0.0) GO TO 280
280 D(I)=D(I)+F
281 C=C+(1.-S)*(1.-S)
282 S=S*(1.-S)
283 F=F+C*S
284 D(I)=D(I)+F
285 G=U(I)
286 IF(G.EQ.0.0) GO TO 287
287 D(I)=D(I)+F
288 C=C+(1.-S)*(1.-S)
289 S=S*(1.-S)
290 F=F+C*S
291 D(I)=D(I)+F
292 G=U(I)
293 IF(G.EQ.0.0) GO TO 294
294 D(I)=D(I)+F
295 C=C+(1.-S)*(1.-S)
296 S=S*(1.-S)
297 F=F+C*S
298 D(I)=D(I)+F
299 G=U(I)
300 IF(G.EQ.0.0) GO TO 301
301 D(I)=D(I)+F
302 C=C+(1.-S)*(1.-S)
303 S=S*(1.-S)
304 F=F+C*S
305 D(I)=D(I)+F
306 G=U(I)
307 IF(G.EQ.0.0) GO TO 308
308 D(I)=D(I)+F
309 C=C+(1.-S)*(1.-S)
310 S=S*(1.-S)
311 F=F+C*S
312 D(I)=D(I)+F
313 G=U(I)
314 IF(G.EQ.0.0) GO TO 315
315 D(I)=D(I)+F
316 C=C+(1.-S)*(1.-S)
317 S=S*(1.-S)
318 F=F+C*S
319 D(I)=D(I)+F
320 G=U(I)
321 IF(G.EQ.0.0) GO TO 322
322 D(I)=D(I)+F
323 C=C+(1.-S)*(1.-S)
324 S=S*(1.-S)
325 F=F+C*S
326 D(I)=D(I)+F
327 G=U(I)
328 IF(G.EQ.0.0) GO TO 329
329 D(I)=D(I)+F
330 C=C+(1.-S)*(1.-S)
331 S=S*(1.-S)
332 F=F+C*S
333 D(I)=D(I)+F
334 G=U(I)
335 IF(G.EQ.0.0) GO TO 336
336 D(I)=D(I)+F
337 C=C+(1.-S)*(1.-S)
338 S=S*(1.-S)
339 F=F+C*S
340 D(I)=D(I)+F
341 G=U(I)
342 IF(G.EQ.0.0) GO TO 343
343 D(I)=D(I)+F
344 C=C+(1.-S)*(1.-S)
345 S=S*(1.-S)
346 F=F+C*S
347 D(I)=D(I)+F
348 G=U(I)
349 IF(G.EQ.0.0) GO TO 350
350 D(I)=D(I)+F
351 C=C+(1.-S)*(1.-S)
352 S=S*(1.-S)
353 F=F+C*S
354 D(I)=D(I)+F
355 G=U(I)
356 IF(G.EQ.0.0) GO TO 357
357 D(I)=D(I)+F
358 C=C+(1.-S)*(1.-S)
359 S=S*(1.-S)
360 F=F+C*S
361 D(I)=D(I)+F
362 G=U(I)
363 IF(G.EQ.0.0) GO TO 364
364 D(I)=D(I)+F
365 C=C+(1.-S)*(1.-S)
366 S=S*(1.-S)
367 F=F+C*S
368 D(I)=D(I)+F
369 G=U(I)
370 IF(G.EQ.0.0) GO TO 371
371 D(I)=D(I)+F
372 C=C+(1.-S)*(1.-S)
373 S=S*(1.-S)
374 F=F+C*S
375 D(I)=D(I)+F
376 G=U(I)
377 IF(G.EQ.0.0) GO TO 378
378 D(I)=D(I)+F
379 C=C+(1.-S)*(1.-S)
380 S=S*(1.-S)
381 F=F+C*S
382 D(I)=D(I)+F
383 G=U(I)
384 IF(G.EQ.0.0) GO TO 385
385 D(I)=D(I)+F
386 C=C+(1.-S)*(1.-S)
387 S=S*(1.-S)
388 F=F+C*S
389 D(I)=D(I)+F
390 G=U(I)
391 IF(G.EQ.0.0) GO TO 392
392 D(I)=D(I)+F
393 C=C+(1.-S)*(1.-S)
394 S=S*(1.-S)
395 F=F+C*S
396 D(I)=D(I)+F
397 G=U(I)
398 IF(G.EQ.0.0) GO TO 399
399 D(I)=D(I)+F
400 C=C+(1.-S)*(1.-S)
401 S=S*(1.-S)
402 F=F+C*S
403 D(I)=D(I)+F
404 G=U(I)
405 IF(G.EQ.0.0) GO TO 406
406 D(I)=D(I)+F
407 C=C+(1.-S)*(1.-S)
408 S=S*(1.-S)
409 F=F+C*S
410 D(I)=D(I)+F
411 G=U(I)
412 IF(G.EQ.0.0) GO TO 413
413 D(I)=D(I)+F
414 C=C+(1.-S)*(1.-S)
415 S=S*(1.-S)
416 F=F+C*S
417 D(I)=D(I)+F
418 G=U(I)
419 IF(G.EQ.0.0) GO TO 420
420 D(I)=D(I)+F
421 C=C+(1.-S)*(1.-S)
422 S=S*(1.-S)
423 F=F+C*S
424 D(I)=D(I)+F
425 G=U(I)
426 IF(G.EQ.0.0) GO TO 427
427 D(I)=D(I)+F
428 C=C+(1.-S)*(1.-S)
429 S=S*(1.-S)
430 F=F+C*S
431 D(I)=D(I)+F
432 G=U(I)
433 IF(G.EQ.0.0) GO TO 434
434 D(I)=D(I)+F
435 C=C+(1.-S)*(1.-S)
436 S=S*(1.-S)
437 F=F+C*S
438 D(I)=D(I)+F
439 G=U(I)
440 IF(G.EQ.0.0) GO TO 441
441 D(I)=D(I)+F
442 C=C+(1.-S)*(1.-S)
443 S=S*(1.-S)
444 F=F+C*S
445 D(I)=D(I)+F
446 G=U(I)
447 IF(G.EQ.0.0) GO TO 448
448 D(I)=D(I)+F
449 C=C+(1.-S)*(1.-S)
450 S=S*(1.-S)
451 F=F+C*S
452 D(I)=D(I)+F
453 G=U(I)
454 IF(G.EQ.0.0) GO TO 455
455 D(I)=D(I)+F
456 C=C+(1.-S)*(1.-S)
457 S=S*(1.-S)
458 F=F+C*S
459 D(I)=D(I)+F
460 G=U(I)
461 IF(G.EQ.0.0) GO TO 462
462 D(I)=D(I)+F
463 C=C+(1.-S)*(1.-S)
464 S=S*(1.-S)
465 F=F+C*S
466 D(I)=D(I)+F
467 G=U(I)
468 IF(G.EQ.0.0) GO TO 469
469 D(I)=D(I)+F
470 C=C+(1.-S)*(1.-S)
471 S=S*(1.-S)
472 F=F+C*S
473 D(I)=D(I)+F
474 G=U(I)
475 IF(G.EQ.0.0) GO TO 476
476 D(I)=D(I)+F
477 C=C+(1.-S)*(1.-S)
478 S=S*(1.-S)
479 F=F+C*S
480 D(I)=D(I)+F
481 G=U(I)
482 IF(G.EQ.0.0) GO TO 483
483 D(I)=D(I)+F
484 C=C+(1.-S)*(1.-S)
485 S=S*(1.-S)
486 F=F+C*S
487 D(I)=D(I)+F
488 G=U(I)
489 IF(G.EQ.0.0) GO TO 490
490 D(I)=D(I)+F
491 C=C+(1.-S)*(1.-S)
492 S=S*(1.-S)
493 F=F+C*S
494 D(I)=D(I)+F
495 G=U(I)
496 IF(G.EQ.0.0) GO TO 497
497 D(I)=D(I)+F
498 C=C+(1.-S)*(1.-S)
499 S=S*(1.-S)
500 F=F+C*S
501 D(I)=D(I)+F
502 G=U(I)
503 IF(G.EQ.0.0) GO TO 504
504 D(I)=D(I)+F
505 C=C+(1.-S)*(1.-S)
506 S=S*(1.-S)
507 F=F+C*S
508 D(I)=D(I)+F
509 G=U(I)
510 IF(G.EQ.0.0) GO TO 511
511 D(I)=D(I)+F
512 C=C+(1.-S)*(1.-S)
513 S=S*(1.-S)
514 F=F+C*S
515 D(I)=D(I)+F
516 G=U(I)
517 IF(G.EQ.0.0) GO TO 518
518 D(I)=D(I)+F
519 C=C+(1.-S)*(1.-S)
520 S=S*(1.-S)
521 F=F+C*S
522 D(I)=D(I)+F
523 G=U(I)
524 IF(G.EQ.0.0) GO TO 525
525 D(I)=D(I)+F
526 C=C+(1.-S)*(1.-S)
527 S=S*(1.-S)
528 F=F+C*S
529 D(I)=D(I)+F
530 G=U(I)
531 IF(G.EQ.0.0) GO TO 532
532 D(I)=D(I)+F
533 C=C+(1.-S)*(1.-S)
534 S=S*(1.-S)
535 F=F+C*S
536 D(I)=D(I)+F
537 G=U(I)
538 IF(G.EQ.0.0) GO TO 539
539 D(I)=D(I)+F
540 C=C+(1.-S)*(1.-S)
541 S=S*(1.-S)
542 F=F+C*S
543 D(I)=D(I)+F
544 G=U(I)
545 IF(G.EQ.0.0) GO TO 546
546 D(I)=D(I)+F
547 C=C+(1.-S)*(1.-S)
548 S=S*(1.-S)
549 F=F+C*S
550 D(I)=D(I)+F
551 G=U(I)
552 IF(G.EQ.0.0) GO TO 553
553 D(I)=D(I)+F
554 C=C+(1.-S)*(1.-S)
555 S=S*(1.-S)
556 F=F+C*S
557 D(I)=D(I)+F
558 G=U(I)
559 IF(G.EQ.0.0) GO TO 560
560 D(I)=D(I)+F
561 C=C+(1.-S)*(1.-S)
562 S=S*(1.-S)
563 F=F+C*S
564 D(I)=D(I)+F
565 G=U(I)
566 IF(G.EQ.0.0) GO TO 567
567 D(I)=D(I)+F
568 C=C+(1.-S)*(1.-S)
569 S=S*(1.-S)
570 F=F+C*S
571 D(I)=D(I)+F
572 G=U(I)
573 IF(G.EQ.0.0) GO TO 574
574 D(I)=D(I)+F
575 C=C+(1.-S)*(1.-S)
576 S=S*(1.-S)
577 F=F+C*S
578 D(I)=D(I)+F
579 G=U(I)
580 IF(G.EQ.0.0) GO TO 581
581 D(I)=D(I)+F
582 C=C+(1.-S)*(1.-S)
583 S=S*(1.-S)
584 F=F+C*S
585 D(I)=D(I)+F
586 G=U(I)
587 IF(G.EQ.0.0) GO TO 588
588 D(I)=D(I)+F
589 C=C+(1.-S)*(1.-S)
590 S=S*(1.-S)
591 F=F+C*S
592 D(I)=D(I)+F
593 G=U(I)
594 IF(G.EQ.0.0) GO TO 595
595 D(I)=D(I)+F
596 C=C+(1.-S)*(1.-S)
597 S=S*(1.-S)
598 F=F+C*S
599 D(I)=D(I)+F
600 G=U(I)
601 IF(G.EQ.0.0) GO TO 602
602 D(I)=D(I)+F
603 C=C+(1.-S)*(1.-S)
604 S=S*(1.-S)
605 F=F+C*S
606 D(I)=D(I)+F
607 G=U(I)
608 IF(G.EQ.0.0) GO TO 609
609 D(I)=D(I)+F
610 C=C+(1.-S)*(1.-S)
611 S=S*(1.-S)
612 F=F+C*S
613 D(I)=D(I)+F
614 G=U(I)
615 IF(G.EQ.0.0) GO TO 616
616 D(I)=D(I)+F
617 C=C+(1.-S)*(1.-S)
618 S=S*(1.-S)
619 F=F+C*S
620 D(I)=D(I)+F
621 G=U(I)
622 IF(G.EQ.0.0) GO TO 623
623 D(I)=D(I)+F
624 C=C+(1.-S)*(1.-S)
625 S=S*(1.-S)
626 F=F+C*S
627 D(I)=D(I)+F
628 G=U(I)
629 IF(G.EQ.0.0) GO TO 630
630 D(I)=D(I)+F
631 C=C+(1.-S)*(1.-S)
632 S=S*(1.-S)
633 F=F+C*S
634 D(I)=D(I)+F
635 G=U(I)
636 IF(G.EQ.0.0) GO TO 637
637 D(I)=D(I)+F
638 C=C+(1.-S)*(1.-S)
639 S=S*(1.-S)
640 F=F+C*S
641 D(I)=D(I)+F
642 G=U(I)
643 IF(G.EQ.0.0) GO TO 644
644 D(I)=D(I)+F
645 C=C+(1.-S)*(1.-S)
646 S=S*(1.-S)
647 F=F+C*S
648 D(I)=D(I)+F
649 G=U(I)
650 IF(G.EQ.0.0) GO TO 651
651 D(I)=D(I)+F
652 C=C+(1.-S)*(1.-S)
653 S=S*(1.-S)
654 F=F+C*S
655 D(I)=D(I)+F
656 G=U(I)
657 IF(G.EQ.0.0) GO TO 658
658 D(I)=D(I)+F
659 C=C+(1.-S)*(1.-S)
660 S=S*(1.-S)
661 F=F+C*S
662 D(I)=D(I)+F
663 G=U(I)
664 IF(G.EQ.0.0) GO TO 665
665 D(I)=D(I)+F
666 C=C+(1.-S)*(1.-S)
667 S=S*(1.-S)
668 F=F+C*S
669 D(I)=D(I)+F
670 G=U(I)
671 IF(G.EQ.0.0) GO TO 672
672 D(I)=D(I)+F
673 C=C+(1.-S)*(1.-S)
674 S=S*(1.-S)
675 F=F+C*S
676 D(I)=D(I)+F
677 G=U(I)
678 IF(G.EQ.0.0) GO TO 679
679 D(I)=D(I)+F
680 C=C+(1.-S)*(1.-S)
681 S=S*(1.-S)
682 F=F+C*S
683 D(I)=D(I)+F
684 G=U(I)
685 IF(G.EQ.0.0) GO TO 686
686 D(I)=D(I)+F
687 C=C+(1.-S)*(1.-S)
688 S=S*(1.-S)
689 F=F+C*S
690 D(I)=D(I)+F
691 G=U(I)
692 IF(G.EQ.0.0) GO TO 693
693 D(I)=D(I)+F
694 C=C+(1.-S)*(1.-S)
695 S=S*(1.-S)
696 F=F+C*S
697 D(I)=D(I)+F
698 G=U(I)
699 IF(G.EQ.0.0) GO TO 700
700 D(I)=D(I)+F
701 C=C+(1.-S)*(1.-S)
702 S=S*(1.-S)
703 F=F+C*S
704 D(I)=D(I)+F
705 G=U(I)
706 IF(G.EQ.0.0) GO TO 707
707 D(I)=D(I)+F
708 C=C+(1.-S)*(1.-S)
709 S=S*(1.-S)
710 F=F+C*S
711 D(I)=D(I)+F
712 G=U(I)
713 IF(G.EQ.0.0) GO TO 714
714 D(I)=D(I)+F
715 C=C+(1.-S)*(1.-S)
716 S=S*(1.-S)
717 F=F+C*S
718 D(I)=D(I)+F
719 G=U(I)
720 IF(G.EQ.0.0) GO TO 721
721 D(I)=D(I)+F
722 C=C+(1.-S)*(1.-S)
723 S=S*(1.-S)
724 F=F+C*S
725 D(I)=D(I)+F
726 G=U(I)
727 IF(G.EQ.0.0) GO TO 728
728 D(I)=D(I)+F
729 C=C+(1.-S)*(1.-S)
730 S=S*(1.-S)
731 F=F+C*S
732 D(I)=D(I)+F
733 G=U(I)
734 IF(G.EQ.0.0) GO TO 735
735 D(I)=D(I)+F
736 C=C+(1.-S)*(1.-S)
737 S=S*(1.-S)
738 F=F+C*S
739 D(I)=D(I)+F
740 G=U(I)
741 IF(G.EQ.0.0) GO TO 742
742 D(I)=D(I)+F
743 C=C+(1.-S)*(1.-S)
744 S=S*(1.-S)
745 F=F+C*S
746 D(I)=D(I)+F
747 G=U(I)
748 IF(G.EQ.0.0) GO TO 749
749 D(I)=D(I)+F
750 C=C+(1.-S)*(1.-S)
751 S=S*(1.-S)
752 F=F+C*S
753 D(I)=D(I)+F
754 G=U(I)
755 IF(G.EQ.0.0) GO TO 756
756 D(I)=D(I)+F
757 C=C+(1.-S)*(1.-S)
758 S=S*(1.-S)
759 F=F+C*S
760 D(I)=D(I)+F
761 G=U(I)
762 IF(G.EQ.0.0) GO TO 763
763 D(I)=D(I)+F
764 C=C+(1.-S)*(1.-S)
765 S=S*(1.-S)
766 F=F+C*S
767 D(I)=D(I)+F
768 G=U(I)
769 IF(G.EQ.0.0) GO TO 770
770 D(I)=D(I)+F
771 C=C+(1.-S)*(1.-S)
772 S=S*(1.-S)
773 F=F+C*S
774 D(I)=D(I)+F
775 G=U(I)
776 IF(G.EQ.0.0) GO TO 777
777 D(I)=D(I)+F
778 C=C+(1.-S)*(1.-S)
779 S=S*(1.-S)
780 F=F+C*S
781 D(I)=D(I)+F
782 G=U(I)
783 IF(G.EQ.0.0) GO TO 784
784 D(I)=D(I)+F
785 C=C+(1.-S)*(1.-S)
786 S=S*(1.-S)
787 F=F+C*S
788 D(I)=D(I)+F
789 G=U(I)
790 IF(G.EQ.0.0) GO TO 791
791 D(I)=D(I)+F
792 C=C+(1.-S)*(1.-S)
793 S=S*(1.-S)
794 F=F+C*S
795 D(I)=D(I)+F
796 G=U(I)
797 IF(G.EQ.0.0) GO TO 798
798 D(I)=D(I)+F
799 C=C+(1.-S)*(1.-S)
800 S=S*(1.-S)
801 F=F+C*S
802 D(I)=D(I)+F
803 G=U(I)
804 IF(G.EQ.0.0) GO TO 805
805 D(I)=D(I)+F
806 C=C+(1.-S)*(1.-S)
807 S=S*(1.-S)
808 F=F+C*S
809 D(I)=D(I)+F
810 G=U(I)
811 IF(G.EQ.0.0) GO TO 812
812 D(I)=D(I)+F
813 C=C+(1.-S)*(1.-S)
814 S=S*(1.-S)
815 F=F+C*S
816 D(I)=D(I)+F
817 G=U(I)
818 IF(G.EQ.0.0) GO TO 819
819 D(I)=D(I)+F
820 C=C+(1.-S)*(1.-S)
821 S=S*(1.-S)
822 F=F+C*S
823 D(I)=D(I)+F
824 G=U(I)
825 IF(G.EQ.0.0) GO TO 826
826 D(I)=D(I)+F
827 C=C+(1.-S)*(1.-S)
828 S=S*(1.-S)
829 F=F+C*S
830 D(I)=D(I)+F
831 G=U(I)
832 IF(G.EQ.0.0) GO TO 833
833 D(I)=D(I)+F
834 C=C+(1.-S)*(1.-S)
835 S=S*(1.-S)
836 F=F+C*S
837 D(I)=D(I)+F
838 G=U(I)
839 IF(G.EQ.0.0) GO TO 840
840 D(I)=D(I)+F
841 C=C+(1.-S)*(1.-S)
842 S=S*(1.-S)
843 F=F+C*S
844 D(I)=D(I)+F
845 G=U(I)
846 IF(G.EQ.0.0) GO TO 847
847 D(I)=D(I)+F
848 C=C+(1.-S)*(1.-S)
849 S=S*(1.-S)
850 F=F+C*S
851 D(I)=D(I)+F
852 G=U(I)
853 IF(G.EQ.0.0) GO TO 854
854 D(I)=D(I)+F
855 C=C+(1.-S)*(1.-S)
856 S=S*(1.-S)
857 F=F+C*S
858 D(I)=D(I)+F
859 G=U(I)
860 IF(G.EQ.0.0) GO TO 861
861 D(I)=D(I)+F
862 C=C+(1.-S)*(1.-S)
863 S=S*(1.-S)
864 F=F+C*S
865 D(I)=D(I)+F
866 G=U(I)
867 IF(G.EQ.0.0) GO TO 868
868 D(I)=D(I)+F
869 C=C+(1.-S)*(1.-S)
870 S=S*(1.-S)
871 F=F+C*S
872 D(I)=D(I)+F
873 G=U(I)
874 IF(G.EQ.0.0) GO TO 875
875 D(I)=D(I)+F
876 C=C+(1.-S)*(1.-S)
877 S=S*(1.-S)
878 F=F+C*S
879 D(I)=D(I)+F
880 G=U(I)
881 IF(G.EQ.0.0) GO TO 882
882 D(I)=D(I)+F
883 C=C+(1.-S)*(1.-S)
884 S=S*(1.-S)
885 F=F+C*S
886 D(I)=D(I)+F
887 G=U(I)
888 IF(G.EQ.0.0) GO TO 889
889 D(I)=D(I)+F
890 C=C+(1.-S)*(1.-S)
891 S=S*(1.-S)
892 F=F+C*S
893 D(I)=D(I)+F
894 G=U(I)
895 IF(G.EQ.0.0) GO TO 896
896 D(I)=D(I)+F
897 C=C+(1.-S)*(1.-S)
898 S=S*(1.-S)
899 F=F+C*S
900 D(I)=D(I)+F
901 G=U(I)
902 IF(G.EQ.0.0) GO TO 903
903 D(I)=D(I)+F
904 C=C+(1.-S)*(1.-S)
905 S=S*(1.-S)
906 F=F+C*S
907 D(I)=D(I)+F
908 G=U(I)
909 IF(G.EQ.0.0) GO TO 910
910 D(I)=D(I)+F
911 C=C+(1.-S)*(1.-S)
912 S=S*(1.-S)
913 F=F+C*S
914 D(I)=D(I)+F
915 G=U(I)
916 IF(G.EQ.0.0) GO TO 917
917 D(I)=D(I)+F
918 C=C+(1.-S)*(1.-S)
919 S=S*(1.-S)
920 F=F+C*S
921 D(I)=D(I)+F
922 G=U(I)
923 IF(G.EQ.0.0) GO TO 924
924 D(I)=D(I)+F
925 C=C+(1.-S)*(1.-S)
926 S=S*(1.-S)
927 F=F+C*S
928 D(I)=D(I)+F
929 G=U(I)
930 IF(G.EQ.0.0) GO TO 931
931 D(I)=D(I)+F
932 C=C+(1.-S)*(1.-S)
933 S=S*(1.-S)
934 F=F+C*S
935 D(I)=D(I)+F
936 G=U(I)
937 IF(G.EQ.0.0) GO TO 938
938 D(I)=D(I)+F
939 C=C+(1.-S)*(1.-S)
940 S=S*(1.-S)
941 F=F+C*S
942 D(I)=D(I)+F
943 G=U(I)
944 IF(G.EQ.0.0) GO TO 945
945 D(I)=D(I)+F
946 C=C+(1.-S)*(1.-S)
947 S=S*(1.-S)
948 F=F+C*S
949 D(I)=D(I)+F
950 G=U(I)
951 IF(G.EQ.0.0) GO TO 952
952 D(I)=D(I)+F
953 C=C+(1.-S)*(1.-S)
954 S=S*(1.-S)
955 F=F+C*S
956 D(I)=D(I)+F
957 G=U(I)
958 IF(G.EQ.0.0) GO TO 959
959 D(I)=D(I)+F
960 C=C+(1.-S)*(1.-S)
961 S=S*(1.-S)
962 F=F+C*S
963 D(I)=D(I)+F
964 G=U(I)
965 IF(G.EQ.0.0) GO TO 966
966 D(I)=D(I)+F
967 C=C+(1.-S)*(1.-S)
968 S=S*(1.-S)
969 F=F+C*S
970 D(I)=D(I)+F
971 G=U(I)
972 IF(G.EQ.0.0) GO TO 973
973 D(I)=D(I)+F
974 C=C+(1.-S)*(1.-S)
975 S=S*(1.-S)
976 F=F+C*S
977 D(I)=D(I)+F
978 G=U(I)
979 IF(G.EQ.0.0) GO TO 980
980 D(I)=D(I)+F
981 C=C+(1.-S)*(1.-S)
982 S=S*(1.-S)
983 F=F+C*S
984 D(I)=D(I)+F
985 G=U(I)
986 IF(G.EQ.0.0) GO TO 987
987 D(I)=D(I)+F
988 C=C+(1.-S)*(1.-S)
989 S=S*(1.-S)
990 F=F+C*S
991 D(I)=D(I)+F
992 G=U(I)
993 IF(G.EQ.0.0) GO TO 994
994 D(I)=D(I)+F
995 C=C+(1.-S)*(1.-S)
996 S=S*(1.-S)
997 F=F+C*S
998 D(I)=D(I)+F
999 G=U(I)
1000 IF(G.EQ.0.0) GO TO 1001
1001 D(I)=D(I)+F
1002 C=C+(1.-S)*(1.-S)
1003 S=S*(1.-S)
1004 F=F+C*S
1005 D(I)=D(I)+F
1006 G=U(I)
1007 IF(G.EQ.0.0) GO TO 1008
1008 D(I)=D(I)+F
1009 C=C+(1.-S)*(1.-S)
1010 S=S*(1.-S)
1011 F=F+C*S
1012 D(I)=D(I)+F
1013 G=U(I)
1014 IF(G.EQ.0.0) GO TO 1015
1015 D(I)=D(I)+F
1016 C=C+(1.-S)*(1.-S)
1017 S=S*(1.-S)
1018 F=F+C*S
1019 D(I)=D(I)+F
1020 G=U(I)
1021 IF(G.EQ.0.0) GO TO 1022
1022 D(I)=D(I)+F
1023 C=C+(1.-S)*(1.-S)
1024 S=S*(1.-S)
1025 F=F+C*S
1026 D(I)=D(I)+F
1027 G=U(I)
1028 IF(G.EQ.0.0) GO TO 1029
1029 D(I)=D(I)+F
1030 C=C+(1.-S)*(1.-S)
1031 S=S*(1.-S)
1032 F=F+C*S
1033 D(I)=D(I)+F
1034 G=U(I)
1035 IF(G.EQ.0.0) GO TO 1036
1036 D(I)=D(I)+F
1037 C=C+(1.-S)*(1.-S)
1038 S=S*(1.-S)
1039 F=F+C*S
1040 D(I)=D(I)+F
1041 G=U(I)
1042 IF(G.EQ.0.0) GO TO 1043
1043 D(I)=D(I)+F
1044 C=C+(1.-S)*(1.-S)
1045 S=S*(1.-S)
1046 F=F+C*S
1047 D(I)=D(I)+F
1048 G=U(I)
1049 IF(G.EQ.0.0) GO TO 1050
1050 D(I)=D(I)+F
1051 C=C+(1.-S)*(1.-S)
1052 S=S*(1.-S)
1053 F=F+C*S
1054 D(I)=D(I)+F
1055 G=U(I)
1056 IF(G.EQ.0.0) GO TO 1057
1057 D(I)=D(I)+F
1058 C=C+(1.-S)*(1.-S)
1059 S=S*(1.-S)
1060 F=F+C*S
1061 D(I)=D(I)+F
1062 G=U(I)
1063 IF(G.EQ.0.0) GO TO 1064
1064 D(I)=D(I)+F
1065 C=C+(1.-S)*(1.-S)
1066 S=S*(1.-S)
1067 F=F+C*S
1068 D(I)=D(I)+F
1069 G=U(I)
1070 IF(G.EQ.0.0) GO TO 1071
1071 D(I)=D(I)+F
1072 C=C+(1.-S)*(1
```

100 1=1
110 0(1)=P
120 CONTINUE
130 GO TO 140
C INICIAÇÃO DE SISTEMA DE CONVENIENCIA PARA ALGUM
C AUTORIZADOR
140 IF E=1
150 RETURN
END

C. 4. D. 10. T. 10.

M. F. SOUZA JATTI
 ESSA ATEÚDO CHAPADA OS AUTOVALORES DE UMA MATRIZ
 REAL COMPARENTE C.
 NA SODRÓTICA HANKEL, STRAVES DE RELAÇÕES RECUR-
 SIVAS CONSTRUIDOS A MATRIZ SIMÉTRICA POSITIVA DE-
 FINIDA E, COMO, DEDUZEMOS ENQUANTO OBTENDO A MATRIZ
 SIMÉTRICA ASSIM-TRANSPORTADA, E QUE POSSIBILITA
 REDUZIR O PROBLEMA SEMELHANTE GENERALIZADO DE ALU-
 MOVALORES DA $\#$ (CARTAS) X, PARA O PROBLEMA SIMÉTRI-
 CO ASSIM-TRANSPORTADO, APENAS SE SUBSTITUIR REDUZ-
 RÍAMOS PELAS RELAÇÕES DIFERENTES DE RELAÇÕES DO ME-
 TODO SEM ELABORAR NOVAMENTE A TABELA SIMÉTRICA E
 SEM MATRIZ SIMÉTRICA, MAS, APENAS TIRAR SIMPLIFICA-
 RÍAMOS.

A FOLHA DE COTAS, SISTEMA DE COTAS, CHAMAMOS AS AUTO-
 VALORES DA $\#$ (CARTAS) X, DEDUZEMOS ENQUANTO A MATRIZ C, ATRAVÉS
 DE ELA, FAZEMOS A RECUPERAÇÃO DE PONTOS.

C 4.7. BALANÇO:

C SUBRITINA BALANÇO.
C ESSA SUBRITINA BALANÇA A MATRIZ REAL GERAL A,
C Isto é, IGUALA AS SOMAS DOS MODULOS DOS ELEMENTOS DAS LINHAS E COLUMNAS CORRESPONDENTES. TAM-BEM ISOLA AUTOVALORES, SEMPRE QUE POSSIVEL.
C N E UMA VARIÁVEL DE ENTRADA.
C A E NA ENTRADA, A MATRIZ DE ORDEM N QUE SERÁ
C BALANÇADA. NA SAÍDA, CONTÉM A MATRIZ TRANSFORMADA.
C LOW E HIGH SÃO VARIÁVEIS INICIAIS DE SAÍDA QUE INDICAM OS LIMITES DA MATRIZ BALANÇADA.
C SCALS E UMA VARIÁVEL REAL DE SAÍDA, QUE CONTÉM INFRAÇÕES SUCINTAS DE TRANSFORMAÇÕES DE SIMILARIEDADE.
C RANK E O PARÂMETRO DEPENDENTE DO COMPUTADOR ESPECIFICADO NA LINHA DE INSTRUÇÕES DE FONTE FUDI-
C PÁGINA. PARA O IBM 360, RANK=2.

SUBRITINA BALANÇO(*n*)
C 1000 IF(*n*.EQ.1) GO TO 1000
C 1010 IF(*n*.LT.1) GO TO 1010
C 1020 IF(*n*.GT.1) GO TO 1020
C 1030 IF(*n*.NE.2) GO TO 1030
C 1040 IF(*n*.LT.2) GO TO 1040
C 1050 IF(*n*.GT.2) GO TO 1050
C 1060 IF(*n*.LT.3) GO TO 1060
C 1070 IF(*n*.GT.3) GO TO 1070
C 1080 IF(*n*.LT.4) GO TO 1080
C 1090 IF(*n*.GT.4) GO TO 1090
C 1100 IF(*n*.LT.5) GO TO 1100
C 1110 IF(*n*.GT.5) GO TO 1110
C 1120 IF(*n*.LT.6) GO TO 1120
C 1130 IF(*n*.GT.6) GO TO 1130
C 1140 IF(*n*.LT.7) GO TO 1140
C 1150 IF(*n*.GT.7) GO TO 1150
C 1160 IF(*n*.LT.8) GO TO 1160
C 1170 IF(*n*.GT.8) GO TO 1170
C 1180 IF(*n*.LT.9) GO TO 1180
C 1190 IF(*n*.GT.9) GO TO 1190
C 1200 IF(*n*.LT.10) GO TO 1200
C 1210 IF(*n*.GT.10) GO TO 1210
C 1220 IF(*n*.LT.11) GO TO 1220
C 1230 IF(*n*.GT.11) GO TO 1230
C 1240 IF(*n*.LT.12) GO TO 1240
C 1250 IF(*n*.GT.12) GO TO 1250
C 1260 IF(*n*.LT.13) GO TO 1260
C 1270 IF(*n*.GT.13) GO TO 1270
C 1280 IF(*n*.LT.14) GO TO 1280
C 1290 IF(*n*.GT.14) GO TO 1290
C 1300 IF(*n*.LT.15) GO TO 1300
C 1310 IF(*n*.GT.15) GO TO 1310
C 1320 IF(*n*.LT.16) GO TO 1320
C 1330 IF(*n*.GT.16) GO TO 1330
C 1340 IF(*n*.LT.17) GO TO 1340
C 1350 IF(*n*.GT.17) GO TO 1350
C 1360 IF(*n*.LT.18) GO TO 1360
C 1370 IF(*n*.GT.18) GO TO 1370
C 1380 IF(*n*.LT.19) GO TO 1380
C 1390 IF(*n*.GT.19) GO TO 1390
C 1400 IF(*n*.LT.20) GO TO 1400
C 1410 IF(*n*.GT.20) GO TO 1410
C 1420 IF(*n*.LT.21) GO TO 1420
C 1430 IF(*n*.GT.21) GO TO 1430
C 1440 IF(*n*.LT.22) GO TO 1440
C 1450 IF(*n*.GT.22) GO TO 1450
C 1460 IF(*n*.LT.23) GO TO 1460
C 1470 IF(*n*.GT.23) GO TO 1470
C 1480 IF(*n*.LT.24) GO TO 1480
C 1490 IF(*n*.GT.24) GO TO 1490
C 1500 IF(*n*.LT.25) GO TO 1500
C 1510 IF(*n*.GT.25) GO TO 1510
C 1520 IF(*n*.LT.26) GO TO 1520
C 1530 IF(*n*.GT.26) GO TO 1530
C 1540 IF(*n*.LT.27) GO TO 1540
C 1550 IF(*n*.GT.27) GO TO 1550
C 1560 IF(*n*.LT.28) GO TO 1560
C 1570 IF(*n*.GT.28) GO TO 1570
C 1580 IF(*n*.LT.29) GO TO 1580
C 1590 IF(*n*.GT.29) GO TO 1590
C 1600 IF(*n*.LT.30) GO TO 1600
C 1610 IF(*n*.GT.30) GO TO 1610
C 1620 IF(*n*.LT.31) GO TO 1620
C 1630 IF(*n*.GT.31) GO TO 1630
C 1640 IF(*n*.LT.32) GO TO 1640
C 1650 IF(*n*.GT.32) GO TO 1650
C 1660 IF(*n*.LT.33) GO TO 1660
C 1670 IF(*n*.GT.33) GO TO 1670
C 1680 IF(*n*.LT.34) GO TO 1680
C 1690 IF(*n*.GT.34) GO TO 1690
C 1700 IF(*n*.LT.35) GO TO 1700
C 1710 IF(*n*.GT.35) GO TO 1710
C 1720 IF(*n*.LT.36) GO TO 1720
C 1730 IF(*n*.GT.36) GO TO 1730
C 1740 IF(*n*.LT.37) GO TO 1740
C 1750 IF(*n*.GT.37) GO TO 1750
C 1760 IF(*n*.LT.38) GO TO 1760
C 1770 IF(*n*.GT.38) GO TO 1770
C 1780 IF(*n*.LT.39) GO TO 1780
C 1790 IF(*n*.GT.39) GO TO 1790
C 1800 IF(*n*.LT.40) GO TO 1800
C 1810 IF(*n*.GT.40) GO TO 1810
C 1820 IF(*n*.LT.41) GO TO 1820
C 1830 IF(*n*.GT.41) GO TO 1830
C 1840 IF(*n*.LT.42) GO TO 1840
C 1850 IF(*n*.GT.42) GO TO 1850
C 1860 IF(*n*.LT.43) GO TO 1860
C 1870 IF(*n*.GT.43) GO TO 1870
C 1880 IF(*n*.LT.44) GO TO 1880
C 1890 IF(*n*.GT.44) GO TO 1890
C 1900 IF(*n*.LT.45) GO TO 1900
C 1910 IF(*n*.GT.45) GO TO 1910
C 1920 IF(*n*.LT.46) GO TO 1920
C 1930 IF(*n*.GT.46) GO TO 1930
C 1940 IF(*n*.LT.47) GO TO 1940
C 1950 IF(*n*.GT.47) GO TO 1950
C 1960 IF(*n*.LT.48) GO TO 1960
C 1970 IF(*n*.GT.48) GO TO 1970
C 1980 IF(*n*.LT.49) GO TO 1980
C 1990 IF(*n*.GT.49) GO TO 1990
C 2000 IF(*n*.LT.50) GO TO 2000
C 2010 IF(*n*.GT.50) GO TO 2010
C 2020 IF(*n*.LT.51) GO TO 2020
C 2030 IF(*n*.GT.51) GO TO 2030
C 2040 IF(*n*.LT.52) GO TO 2040
C 2050 IF(*n*.GT.52) GO TO 2050
C 2060 IF(*n*.LT.53) GO TO 2060
C 2070 IF(*n*.GT.53) GO TO 2070
C 2080 IF(*n*.LT.54) GO TO 2080
C 2090 IF(*n*.GT.54) GO TO 2090
C 2100 IF(*n*.LT.55) GO TO 2100
C 2110 IF(*n*.GT.55) GO TO 2110
C 2120 IF(*n*.LT.56) GO TO 2120
C 2130 IF(*n*.GT.56) GO TO 2130
C 2140 IF(*n*.LT.57) GO TO 2140
C 2150 IF(*n*.GT.57) GO TO 2150
C 2160 IF(*n*.LT.58) GO TO 2160
C 2170 IF(*n*.GT.58) GO TO 2170
C 2180 IF(*n*.LT.59) GO TO 2180
C 2190 IF(*n*.GT.59) GO TO 2190
C 2200 IF(*n*.LT.60) GO TO 2200
C 2210 IF(*n*.GT.60) GO TO 2210
C 2220 IF(*n*.LT.61) GO TO 2220
C 2230 IF(*n*.GT.61) GO TO 2230
C 2240 IF(*n*.LT.62) GO TO 2240
C 2250 IF(*n*.GT.62) GO TO 2250
C 2260 IF(*n*.LT.63) GO TO 2260
C 2270 IF(*n*.GT.63) GO TO 2270
C 2280 IF(*n*.LT.64) GO TO 2280
C 2290 IF(*n*.GT.64) GO TO 2290
C 2300 IF(*n*.LT.65) GO TO 2300
C 2310 IF(*n*.GT.65) GO TO 2310
C 2320 IF(*n*.LT.66) GO TO 2320
C 2330 IF(*n*.GT.66) GO TO 2330
C 2340 IF(*n*.LT.67) GO TO 2340
C 2350 IF(*n*.GT.67) GO TO 2350
C 2360 IF(*n*.LT.68) GO TO 2360
C 2370 IF(*n*.GT.68) GO TO 2370
C 2380 IF(*n*.LT.69) GO TO 2380
C 2390 IF(*n*.GT.69) GO TO 2390
C 2400 IF(*n*.LT.70) GO TO 2400
C 2410 IF(*n*.GT.70) GO TO 2410
C 2420 IF(*n*.LT.71) GO TO 2420
C 2430 IF(*n*.GT.71) GO TO 2430
C 2440 IF(*n*.LT.72) GO TO 2440
C 2450 IF(*n*.GT.72) GO TO 2450
C 2460 IF(*n*.LT.73) GO TO 2460
C 2470 IF(*n*.GT.73) GO TO 2470
C 2480 IF(*n*.LT.74) GO TO 2480
C 2490 IF(*n*.GT.74) GO TO 2490
C 2500 IF(*n*.LT.75) GO TO 2500
C 2510 IF(*n*.GT.75) GO TO 2510
C 2520 IF(*n*.LT.76) GO TO 2520
C 2530 IF(*n*.GT.76) GO TO 2530
C 2540 IF(*n*.LT.77) GO TO 2540
C 2550 IF(*n*.GT.77) GO TO 2550
C 2560 IF(*n*.LT.78) GO TO 2560
C 2570 IF(*n*.GT.78) GO TO 2570
C 2580 IF(*n*.LT.79) GO TO 2580
C 2590 IF(*n*.GT.79) GO TO 2590
C 2600 IF(*n*.LT.80) GO TO 2600
C 2610 IF(*n*.GT.80) GO TO 2610
C 2620 IF(*n*.LT.81) GO TO 2620
C 2630 IF(*n*.GT.81) GO TO 2630
C 2640 IF(*n*.LT.82) GO TO 2640
C 2650 IF(*n*.GT.82) GO TO 2650
C 2660 IF(*n*.LT.83) GO TO 2660
C 2670 IF(*n*.GT.83) GO TO 2670
C 2680 IF(*n*.LT.84) GO TO 2680
C 2690 IF(*n*.GT.84) GO TO 2690
C 2700 IF(*n*.LT.85) GO TO 2700
C 2710 IF(*n*.GT.85) GO TO 2710
C 2720 IF(*n*.LT.86) GO TO 2720
C 2730 IF(*n*.GT.86) GO TO 2730
C 2740 IF(*n*.LT.87) GO TO 2740
C 2750 IF(*n*.GT.87) GO TO 2750
C 2760 IF(*n*.LT.88) GO TO 2760
C 2770 IF(*n*.GT.88) GO TO 2770
C 2780 IF(*n*.LT.89) GO TO 2780
C 2790 IF(*n*.GT.89) GO TO 2790
C 2800 IF(*n*.LT.90) GO TO 2800
C 2810 IF(*n*.GT.90) GO TO 2810
C 2820 IF(*n*.LT.91) GO TO 2820
C 2830 IF(*n*.GT.91) GO TO 2830
C 2840 IF(*n*.LT.92) GO TO 2840
C 2850 IF(*n*.GT.92) GO TO 2850
C 2860 IF(*n*.LT.93) GO TO 2860
C 2870 IF(*n*.GT.93) GO TO 2870
C 2880 IF(*n*.LT.94) GO TO 2880
C 2890 IF(*n*.GT.94) GO TO 2890
C 2900 IF(*n*.LT.95) GO TO 2900
C 2910 IF(*n*.GT.95) GO TO 2910
C 2920 IF(*n*.LT.96) GO TO 2920
C 2930 IF(*n*.GT.96) GO TO 2930
C 2940 IF(*n*.LT.97) GO TO 2940
C 2950 IF(*n*.GT.97) GO TO 2950
C 2960 IF(*n*.LT.98) GO TO 2960
C 2970 IF(*n*.GT.98) GO TO 2970
C 2980 IF(*n*.LT.99) GO TO 2980
C 2990 IF(*n*.GT.99) GO TO 2990
C 3000 IF(*n*.LT.100) GO TO 3000
C 3010 IF(*n*.GT.100) GO TO 3010
C 3020 IF(*n*.LT.101) GO TO 3020
C 3030 IF(*n*.GT.101) GO TO 3030
C 3040 IF(*n*.LT.102) GO TO 3040
C 3050 IF(*n*.GT.102) GO TO 3050
C 3060 IF(*n*.LT.103) GO TO 3060
C 3070 IF(*n*.GT.103) GO TO 3070
C 3080 IF(*n*.LT.104) GO TO 3080
C 3090 IF(*n*.GT.104) GO TO 3090
C 3100 IF(*n*.LT.105) GO TO 3100
C 3110 IF(*n*.GT.105) GO TO 3110
C 3120 IF(*n*.LT.106) GO TO 3120
C 3130 IF(*n*.GT.106) GO TO 3130
C 3140 IF(*n*.LT.107) GO TO 3140
C 3150 IF(*n*.GT.107) GO TO 3150
C 3160 IF(*n*.LT.108) GO TO 3160
C 3170 IF(*n*.GT.108) GO TO 3170
C 3180 IF(*n*.LT.109) GO TO 3180
C 3190 IF(*n*.GT.109) GO TO 3190
C 3200 IF(*n*.LT.110) GO TO 3200
C 3210 IF(*n*.GT.110) GO TO 3210
C 3220 IF(*n*.LT.111) GO TO 3220
C 3230 IF(*n*.GT.111) GO TO 3230
C 3240 IF(*n*.LT.112) GO TO 3240
C 3250 IF(*n*.GT.112) GO TO 3250
C 3260 IF(*n*.LT.113) GO TO 3260
C 3270 IF(*n*.GT.113) GO TO 3270
C 3280 IF(*n*.LT.114) GO TO 3280
C 3290 IF(*n*.GT.114) GO TO 3290
C 3300 IF(*n*.LT.115) GO TO 3300
C 3310 IF(*n*.GT.115) GO TO 3310
C 3320 IF(*n*.LT.116) GO TO 3320
C 3330 IF(*n*.GT.116) GO TO 3330
C 3340 IF(*n*.LT.117) GO TO 3340
C 3350 IF(*n*.GT.117) GO TO 3350
C 3360 IF(*n*.LT.118) GO TO 3360
C 3370 IF(*n*.GT.118) GO TO 3370
C 3380 IF(*n*.LT.119) GO TO 3380
C 3390 IF(*n*.GT.119) GO TO 3390
C 3400 IF(*n*.LT.120) GO TO 3400
C 3410 IF(*n*.GT.120) GO TO 3410
C 3420 IF(*n*.LT.121) GO TO 3420
C 3430 IF(*n*.GT.121) GO TO 3430
C 3440 IF(*n*.LT.122) GO TO 3440
C 3450 IF(*n*.GT.122) GO TO 3450
C 3460 IF(*n*.LT.123) GO TO 3460
C 3470 IF(*n*.GT.123) GO TO 3470
C 3480 IF(*n*.LT.124) GO TO 3480
C 3490 IF(*n*.GT.124) GO TO 3490
C 3500 IF(*n*.LT.125) GO TO 3500
C 3510 IF(*n*.GT.125) GO TO 3510
C 3520 IF(*n*.LT.126) GO TO 3520
C 3530 IF(*n*.GT.126) GO TO 3530
C 3540 IF(*n*.LT.127) GO TO 3540
C 3550 IF(*n*.GT.127) GO TO 3550
C 3560 IF(*n*.LT.128) GO TO 3560
C 3570 IF(*n*.GT.128) GO TO 3570
C 3580 IF(*n*.LT.129) GO TO 3580
C 3590 IF(*n*.GT.129) GO TO 3590
C 3600 IF(*n*.LT.130) GO TO 3600
C 3610 IF(*n*.GT.130) GO TO 3610
C 3620 IF(*n*.LT.131) GO TO 3620
C 3630 IF(*n*.GT.131) GO TO 3630
C 3640 IF(*n*.LT.132) GO TO 3640
C 3650 IF(*n*.GT.132) GO TO 3650
C 3660 IF(*n*.LT.133) GO TO 3660
C 3670 IF(*n*.GT.133) GO TO 3670
C 3680 IF(*n*.LT.134) GO TO 3680
C 3690 IF(*n*.GT.134) GO TO 3690
C 3700 IF(*n*.LT.135) GO TO 3700
C 3710 IF(*n*.GT.135) GO TO 3710
C 3720 IF(*n*.LT.136) GO TO 3720
C 3730 IF(*n*.GT.136) GO TO 3730
C 3740 IF(*n*.LT.137) GO TO 3740
C 3750 IF(*n*.GT.137) GO TO 3750
C 3760 IF(*n*.LT.138) GO TO 3760
C 3770 IF(*n*.GT.138) GO TO 3770
C 3780 IF(*n*.LT.139) GO TO 3780
C 3790 IF(*n*.GT.139) GO TO 3790
C 3800 IF(*n*.LT.140) GO TO 3800
C 3810 IF(*n*.GT.140) GO TO 3810
C 3820 IF(*n*.LT.141) GO TO 3820
C 3830 IF(*n*.GT.141) GO TO 3830
C 3840 IF(*n*.LT.142) GO TO 3840
C 3850 IF(*n*.GT.142) GO TO 3850
C 3860 IF(*n*.LT.143) GO TO 3860
C 3870 IF(*n*.GT.143) GO TO 3870
C 3880 IF(*n*.LT.144) GO TO 3880
C 3890 IF(*n*.GT.144) GO TO 3890
C 3900 IF(*n*.LT.145) GO TO 3900
C 3910 IF(*n*.GT.145) GO TO 3910
C 3920 IF(*n*.LT.146) GO TO 3920
C 3930 IF(*n*.GT.146) GO TO 3930
C 3940 IF(*n*.LT.147) GO TO 3940
C 3950 IF(*n*.GT.147) GO TO 3950
C 3960 IF(*n*.LT.148) GO TO 3960
C 3970 IF(*n*.GT.148) GO TO 3970
C 3980 IF(*n*.LT.149) GO TO 3980
C 3990 IF(*n*.GT.149) GO TO 3990
C 4000 IF(*n*.LT.150) GO TO 4000
C 4010 IF(*n*.GT.150) GO TO 4010
C 4020 IF(*n*.LT.151) GO TO 4020
C 4030 IF(*n*.GT.151) GO TO 4030
C 4040 IF(*n*.LT.152) GO TO 4040
C 4050 IF(*n*.GT.152) GO TO 4050
C 4060 IF(*n*.LT.153) GO TO 4060
C 4070 IF(*n*.GT.153) GO TO 4070
C 4080 IF(*n*.LT.154) GO TO 4080
C 4090 IF(*n*.GT.154) GO TO 4090
C 4100 IF(*n*.LT.155) GO TO 4100
C 4110 IF(*n*.GT.155) GO TO 4110
C 4120 IF(*n*.LT.156) GO TO 4120
C 4130 IF(*n*.GT.156) GO TO 4130
C 4140 IF(*n*.LT.157) GO TO 4140
C 4150 IF(*n*.GT.157) GO TO 4150
C 4160 IF(*n*.LT.158) GO TO 4160
C 4170 IF(*n*.GT.158) GO TO 4170
C 4180 IF(*n*.LT.159) GO TO 4180
C 4190 IF(*n*.GT.159) GO TO 4190
C 4200 IF(*n*.LT.160) GO TO 4200
C 4210 IF(*n*.GT.160) GO TO 4210
C 4220 IF(*n*.LT.161) GO TO 4220
C 4230 IF(*n*.GT.161) GO TO 4230
C 4240 IF(*n*.LT.162) GO TO 4240
C 4250 IF(*n*.GT.162) GO TO 4250
C 4260 IF(*n*.LT.163) GO TO 4260
C 4270 IF(*n*.GT.163) GO TO 4270
C 4280 IF(*n*.LT.164) GO TO 4280
C 4290 IF(*n*.GT.164) GO TO 4290
C 4300 IF(*n*.LT.165) GO TO 4300
C 4310 IF(*n*.GT.165) GO TO 4310
C 4320 IF(*n*.LT.166) GO TO 4320
C 4330 IF(*n*.GT.166) GO TO 4330
C 4340 IF(*n*.LT.167) GO TO 4340
C 4350 IF(*n*.GT.167) GO TO 4350
C 4360 IF(*n*.LT.168) GO TO 4360
C 4370 IF(*n*.GT.168) GO TO 4370
C 4380 IF(*n*.LT.169) GO TO 4380
C 4390 IF(*n*.GT.169) GO TO 4390
C 4400 IF(*n*.LT.170) GO TO 4400
C 4410 IF(*n*.GT.170) GO TO 4410
C 4420 IF(*n*.LT.171) GO TO 4420
C 4430 IF(*n*.GT.171) GO TO 4430
C 4440 IF(*n*.LT.172) GO TO 4440
C 4450 IF(*n*.GT.172) GO TO 4450
C 4460 IF(*n*.LT.173) GO TO 4460
C 4470 IF(*n*.GT.173) GO TO 4470
C 4480 IF(*n*.LT.174) GO TO 4480
C 4490 IF(*n*.GT.174) GO TO 4490
C 4500 IF(*n*.LT.175) GO TO 4500
C 4510 IF(*n*.GT.175) GO TO 4510
C 4520 IF(*n*.LT.176) GO TO 4520
C 4530 IF(*n*.GT.176) GO TO 4530
C 4540 IF(*n*.LT.177) GO TO 4540
C 4550 IF(*n*.GT.177) GO TO 4550
C 4560 IF(*n*.LT.178) GO TO 4560
C 4570 IF(*n*.GT.178) GO TO 4570
C 4580 IF(*n*.LT.179) GO TO 4580
C 4590 IF(*n*.GT.179) GO TO 4590
C 4600 IF(*n*.LT.180) GO TO 4600
C 4610 IF(*n*.GT.180) GO TO 4610
C 4620 IF(*n*.LT.181) GO TO 4620
C 4630 IF(*n*.GT.181) GO TO 4630
C 4640 IF(*n*.LT.182) GO TO 4640
C 4650 IF(*n*.GT.182) GO TO 4650
C 4660 IF(*n*.LT.183) GO TO 4660
C 4670 IF(*n*.GT.183) GO TO 4670
C 4680 IF(*n*.LT.184) GO TO 4680
C 4690 IF(*n*.GT.184) GO TO 4690
C 4700 IF(*n*.LT.185) GO TO 4700
C 4710 IF(*n*.GT.185) GO TO 4710
C 4720 IF(*n*.LT.186) GO TO 4720
C 4730 IF(*n*.GT.186) GO TO 4730
C 4740 IF(*n*.LT.187) GO TO 4740
C 4750 IF(*n*.GT.187) GO TO 4750
C 4760 IF(*n*.LT.188) GO TO 4760
C 4770 IF(*n*.GT.188) GO TO 4770
C 4780 IF(*n*.LT.189) GO TO 4780
C 4790 IF(*n*.GT.189) GO TO 4790
C 4800 IF(*n*.LT.190) GO TO 4800
C 4810 IF(*n*.GT.190) GO TO 4810
C 4820 IF(*n*.LT.191) GO TO 4820
C 4830 IF(*n*.GT.191) GO TO 4830
C 4840 IF(*n*.LT.192) GO TO 4840
C 4850 IF(*n*.GT.192) GO TO 4850
C 4860 IF(*n*.LT.193) GO TO 4860
C 4870 IF(*n*.GT.193) GO TO 4870
C 4880 IF(*n*.LT.194) GO TO 4880
C 4890 IF(*n*.GT.194) GO TO 4890
C 4900 IF(*n*.LT.195) GO TO 4900
C 4910 IF(*n*.GT.195) GO TO 4910
C 4920

```

100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
260
261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
279
280
281
282
283
284
285
286
287
288
289
289
290
291
292
293
294
295
296
297
298
299
299
300
301
302
303
304
305
306
307
308
309
309
310
311
312
313
314
315
316
317
318
319
319
320
321
322
323
324
325
326
327
328
329
329
330
331
332
333
334
335
336
337
338
339
339
340
341
342
343
344
345
346
347
348
349
349
350
351
352
353
354
355
356
357
358
359
359
360
361
362
363
364
365
366
367
368
369
369
370
371
372
373
374
375
376
377
378
379
379
380
381
382
383
384
385
386
387
388
389
389
390
391
392
393
394
395
396
397
398
399
399
400
401
402
403
404
405
406
407
408
409
409
410
411
412
413
414
415
416
417
418
419
419
420
421
422
423
424
425
426
427
428
429
429
430
431
432
433
434
435
436
437
438
439
439
440
441
442
443
444
445
446
447
448
449
449
450
451
452
453
454
455
456
457
458
459
459
460
461
462
463
464
465
466
467
468
469
469
470
471
472
473
474
475
476
477
478
479
479
480
481
482
483
484
485
486
487
488
489
489
490
491
492
493
494
495
496
497
498
499
499
500
501
502
503
504
505
506
507
508
509
509
510
511
512
513
514
515
516
517
518
519
519
520
521
522
523
524
525
526
527
528
529
529
530
531
532
533
534
535
536
537
538
539
539
540
541
542
543
544
545
546
547
548
549
549
550
551
552
553
554
555
556
557
558
559
559
560
561
562
563
564
565
566
567
568
569
569
570
571
572
573
574
575
576
577
578
579
579
580
581
582
583
584
585
586
587
588
589
589
590
591
592
593
594
595
596
597
598
599
599
600
601
602
603
604
605
606
607
608
609
609
610
611
612
613
614
615
616
617
618
619
619
620
621
622
623
624
625
626
627
628
629
629
630
631
632
633
634
635
636
637
638
639
639
640
641
642
643
644
645
646
647
648
649
649
650
651
652
653
654
655
656
657
658
659
659
660
661
662
663
664
665
666
667
668
669
669
670
671
672
673
674
675
676
677
678
679
679
680
681
682
683
684
685
686
687
688
689
689
690
691
692
693
694
695
696
697
698
698
699
700
701
702
703
704
705
706
707
708
709
709
710
711
712
713
714
715
716
717
718
719
719
720
721
722
723
724
725
726
727
728
729
729
730
731
732
733
734
735
736
737
738
739
739
740
741
742
743
744
745
746
747
748
749
749
750
751
752
753
754
755
756
757
758
759
759
760
761
762
763
764
765
766
767
768
769
769
770
771
772
773
774
775
776
777
778
779
779
780
781
782
783
784
785
786
787
788
789
789
790
791
792
793
794
795
796
797
798
798
799
800
801
802
803
804
805
806
807
808
809
809
810
811
812
813
814
815
816
817
818
819
819
820
821
822
823
824
825
826
827
828
829
829
830
831
832
833
834
835
836
837
838
839
839
840
841
842
843
844
845
846
847
848
849
849
850
851
852
853
854
855
856
857
858
859
859
860
861
862
863
864
865
866
867
868
869
869
870
871
872
873
874
875
876
877
878
879
879
880
881
882
883
884
885
886
887
888
889
889
890
891
892
893
894
895
896
897
898
898
899
900
901
902
903
904
905
906
907
908
909
909
910
911
912
913
914
915
916
917
918
919
919
920
921
922
923
924
925
926
927
928
929
929
930
931
932
933
934
935
936
937
938
939
939
940
941
942
943
944
945
946
947
948
949
949
950
951
952
953
954
955
956
957
958
959
959
960
961
962
963
964
965
966
967
968
969
969
970
971
972
973
974
975
976
977
978
979
979
980
981
982
983
984
985
986
987
988
989
989
990
991
992
993
994
995
996
997
998
999
999
1000

```

21.0 $\beta \cup \beta_1, \beta_2 = \beta \cup (\beta_1, \beta_2) * S^1$
27.0 $\mathbb{C} \otimes \mathbb{C}[1] \otimes \mathbb{C}$
 $\text{Tr}(\text{Tr}(\text{Tr}(G, V)) \otimes G) = \text{Tr}(G) \otimes G$
21.0 $\text{Tr}_G \otimes \mathbb{C}[1]$
 $\text{Tr}_G \otimes \mathbb{C}[1]$
 $\mathbb{C}[1] \otimes \mathbb{C}[1]$

C 4.8. BLANC

C SUBSTITUI RAZ^Z
C ESSA SUBRUTINA DETERMINA OS AUTOVALORES DE UMA
C MATRIZ DE HESSEBERG SUPERIOR REAL H, USANDO O
C MÉTODO QR.
C N É UMA VARIÁVEL INTEIRA DE ENTRADA.
C LOW E IGH SÃO VARIÁVEIS INTEIRAS DE ENTRADA QUE
C INDICAM OS ÍNDICES DO LIMITE DA MATRIZ BALANCEA-
C DA. SE A MATRIZ NÃO É BALANÇADA, LOW=1 E IGH=N.
C H É A ENTRADA À MATRIZ DE HESSEBERG SUPERIOR
C DE TAL MODO QUE ALGORITMO QUE DESTROI A MATRIZ H.
C LOW E IGH SÃO OS ÍNDICES DE DIAGONAL, CONTENDO
C OS AUTOVALORES REais DA MATRIZ DE HESSEBERG
C DE P. ESSA SUBRUTINA NÃO SÓ COMPUTA OS AUTOVALORES,
C MAS TAMBÉM COMPUTA AS RAIZES ISOLADAS DA MATRIZ INA-
C BALADA, QUE SÃO AS RAIZES DA EQUAÇÃO CHAR-
C ACTIVA. A TÉCNICA EMPREGADA PARA COMPUTAR A FALHA
C DE HESSEBERG É A SUCCESSIONE DE SUBSTITUIÇÃO DA
C LINHA DE BALANÇAMENTO, ATÉ QUE SEJA A SEGUINTE.
C QUANDO SE CHEGA A UMA LINHA DE BALANÇAMENTO COM
C RAIZ ISOLADA, A SUBRUTINA FAZ UMA CHAMADA PARA
C SUBRUTINA SUBR, QUE COMPUTA A RAIZ ISOLADA.
C VAI SEUS VALORES NA LINHA DE BALANÇAMENTO, E FAZ
C A SUBSTITUIÇÃO DA LINHA DE BALANÇAMENTO. SE NÃO HOU-
CVER RAIZ ISOLADA, SE SLEVA, E SE PSS > 0, PODE
C HACER, CASO NAO, PSS > 1. PARA PSS = 10
C VAI SEUS VALORES DA LINHA DE BALANÇAMENTO.

C SUBRUTINA PARA COMPUTAR RAIZES ISOLADAS DA LINHA DE BALANÇAMENTO H(I,J)
C IF(J.LT.1) GO TO 10
C SUBR(H,I,J)
C 10 IF(J.EQ.I) GO TO 12
C SUBR(H,I,J)
C 12 SUBR(H,I,J)
C GUARDA AS RAIZES ISOLADAS PENO BALANÇAMENTO
C E COMPUTA A NORMA DA MATRIZ
C DO 50 I=1,N
C DO 40 J=K,N
C NORM=NORM+ABS(H(I,J))
C K=I
C IF(I.GE.LOW.AND.I.LE.IGH) GO TO 50
C WR(I)=H(I,I)

5 CONTINUE
EN=1GN
T=0.
C PESQUISA DOS PRÓXIMOS AUTOVALORES
6 IF(LN,LF,LND) GO TO 100
LND=0
MA=MN=1
ENM2=NA=1
C PROCURA UM ELEMANTO PEQUENO DA SUBDIAGONAL.
7 DO 80 LL=LDA,EN
L=EN+LDS+LL
IF(LL,LE,LND) GO TO 100
S=ABS(H(L-1,M-1))+4*ABS(H(L,L))
IF(S,LE,0.) S=100
IF(ABS(H(L,L-1)),LT,MINEPS*S) GO TO 100
8 CONTINUE
C FAZIAÇÃO DA PARTIÇÃO-VALOR X
9 X=A(I,I,M)
IF(I,I,M,0) GOTO 120
Y=A(I,I,M-1)
Z=H(M,M,A)+H(M-1,M)
IF(C,LE,1) Z=H(M,M)
IF(C,LE,0) Z=H(M-1,M)
IF(X,I,M,0) GOTO 120
IF(X,I,M,1) GOTO 120
C FAZIAÇÃO DA PARTIÇÃO-VALOR EXCEPCIONAL
10 L=L+X
R=I+X
P=(I,I)=A(I,I)=X
Q=4*ABS(H(I,I,M))+4*ABS(H(I,I,M-1))
K=I+7)*S
T=X
X=0.4375*S
110 IFS=IFS+1
C PROCURA DUS OUTROS MELHORES ELEMENTOS CONSCU-
120 TILOS DA SUBDIAGONAL
P=I+9*I+I, P=I+I
M=M+2+L-1
Z2=A(I,M+1)
R=X-Z2
S=Y-Z2
P=(X+Z2)/H(I+1,M)+H(I,M+1)
Q=H(M+1,M+1)-Z2-R-S
K=H(M+2,M+1)
S=ABS(P)+ABS(Q)+ABS(K)
R=P/S
Q=Q/S
R=R/S
IF(M,EQ,L) GO TO 150
IF(ABS(H(M,M+1))*(ABS(Q)+ABS(R)),LE,MINEPS*ABS(P)*
130 X*(ABS(H(M-1,M+1))+ABS(Z2)+ABS(H(M+1,M+1))))GO TO 150
140 CONTINUE
150 MP2=M+2
DO 160 I=MP2,EN

H(I,J-2)=1.
IF(I.EQ.MP2) GO TO 160
H(I,I-3)=0.
CONTINUE
PRINT OUT AND PRINT A MELTHERED AS LINHAS DA A
ELETRONICA DE ALTA
DO 160 K=1,NA
NUTLAS=N+6.1A
IF(K.EQ.M) GO TO 170
P=H(K,K+1)
Q=H(S+1,K+1)
R=0.
IF(NUTLAS .EQ. 0) H(K+2,K+1)=H(K+2,K+1)
X=H(K,K)+H(K+1,K+1)+H(K+2,K+2)
IF(X.GT.0.1) H(K,K)=H(K,K)-X
P=P/X
Q=Q/X
R=R/X
H(K,K)=H(K,K)+P*(H(K+1,K+1))
IF(H(K,K).LT.0.1) H(K,K)=0.
H(K,K+1)=H(K,K+1)-P
H(K,K+2)=H(K,K+2)-Q
H(K,K+3)=H(K,K+3)-R
CONTINUE
IF(I.EQ.MP2) GO TO 170
P=H(K,K+1)+H(K+1,K+1)
IF(I.EQ.MP1) GO TO 170
P=P+H(K+2,K+2)
H(I,K+2)=H(I,K+2)-P*Z
H(I,K+1)=H(I,K+1)-P*Y
H(I,K)=H(I,K)-P
CONTINUE
CONTINUE
GO TO 70
C UMA RAIIZ E ACHADA
270 WR(EN)=X+T

EN=RA
GO TO 60
C DUAS RAIZES SAO ACHADAS
280 P=(Y-X)/2.
Q=P*P+d
ZZ=SQRT(ABS(Q))
X=X+T
IF(Q.LT.0.) GO TO 320
C PAR REAL
ZZ=P+SIGN(ZZ,P)
RH(RA)=X+ZZ
RH(EN)=X-w/ZZ
GO TO 330
310 L(G)=1
320 H(G)=L(G)*2
G2=1/(1-L(G))
C L(G)=C1*(1-G2)+L(G1)*(1-G2)+L(G2)*(1-G1)
C RH(RA)=L(G)*H(G)+RH(EN)
1-3 L(G)=RH(RA)
1-4 H(G)=RH(EN)

C 4.4.1878

11-127000 Q2

ESSE MÉTODO COMPUTA OS AUTOVALORES REAIS DA TRANSPOSTA DA MATRIZ COMPANHEIRA REAL C.
 PRIMEIRAMENTE, É FEITA A MONTAGEM DA TRANSPOSTA DA MATRIZ C, , ONDE A ÚLTIMA LINHA É FORNECIDA COMO DADO DE ENTRADA.
 NA SUBROTINA BALANC, A MATRIZ É BALANÇADA, DE MODO QUE AS SÍNTAXES DAS LINHAS E COLUNAS SEjam IGUAIS. SE NÃO FOR SÓLIDAS, PODE OCORRER DE ALGUM AUTOVALOR SER PERDIDO.
 NA SUBROTINA HOR, OS AUTOVALORES REAIS SÃO FINALMENTE COMPUTADOS.

REFERÉNCIAS:

- [1] DATTA, B. N. - "Application of Hankel Matrix to the root-location problem", IEEE - Trans. on Auto. Control., Agosto/1976, p. 610-612.
- [2] DATTA, B. N. - "Application of Hankel matrices of Markov parameters to the solution of the Routh-Hurwitz and the Schur-Cohn problems", J. Math. Anal. Appl., v. 68, nº 1, 1979, p. 276-290.
- [3] WILKINSON, J. H. - "The Algebraic Eigenvalue Problem", Clarendon Press, Oxford, 1965.
- [4] WILKINSON, J. H. and REINSCH, C. - "Handbook for Automatic Computation, Volume II, Linear Algebra", Springer - Verlag, New York - Heidelberg - Berlin, 1971.
- [5] SMITH, B. T. and others - "Matrix Eigensystem Routines, Eispack Guide", Lectures Notes in Computer Science, V. 6, Springer-Verlag, 1976.
- [6] GARBOW, B. S. and others - "Matrix Eigensystem Routines, Eispack Guide Extension", Lect. Notes in Comp. Science, V 51, Springer-Verlag, 1977.
- [7] BLUM, K. - "Numerical Analysis and Computation, Theory and Practice", Addison-Wesley, Reading, Mass., 1972, p. 230-264.

- [8] KRISHNAMURTHY, E. V. and SEN, S. K. - "Computer Based Numerical Algorithms", EWP (Affiliated East-West Press PVT Ltd.), New Delhi-Madras, 1976 - p. 248 - 249.
- [9] GRAD, J. and BREBNER, M. A. - "Eigenvalues and Eigenvectors of a Real General Matrix", Comm. of the ACM , Vol. 11, nº 12, 1968, p. 820-825.
- [10] BARROS, I. Q. - "Métodos Numéricos, I, Álgebra Linear", Brasil, Março/70, p. 46-52, 122-135, 143-158.
- [11] GREGORY and KARNEY - "A Collection of Matrices for testing computational algorithms", Wiley, New York, 1969, p. 55-113, 134-142.
- [12] DATTA, B. N. - Notas de aula do curso de Análise Numérica I, UNICAMP, Março/77.
- [13] TAUSSKY, O. and ZASSENHAUS, H. - "On the similarity transformation between a matrix and its transpose", Pacific J. Math., vol. 9, 1959, p. 893-896.
- [14] BARNET, S. - "Matrices in Control Theory", Van Nostrand-Reinhold, London, 1971, p. 39.
- [15] REINSCH, C. H. - "A stable, rational QR algorithm for the computation of the eigenvalues of an Hermitian, Tridiagonal matrix", Math. of Comput., v. 25, nº 115, 1971, p. 591-597.

- [16] DORN, W. S. e MC CRACKEN, D. - "Cálculo Numérico com
estudos de casos em fortran IV", ED. Campus (USP),
Rio de Janeiro, 1978, p. 95-138.
- [17] GANTMACHER, F. R. - "The Theory of Matrices", v. 2 ,
Chelsea Publ. Company, New York, 1971, p. 214.
- [18] PARLETT, B. N. - "Global Convergence of the basic QR
algorithm on Hessenberg Matrices", Math. Comp.
22, 1968, p. 803-817.
- [19] WILKINSON, J. H. - "Convergence of the LR, QR and
related algorithms", Comput. J., v. 8, 1965, p.
77-84.