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 requisito 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 texto.

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 autovalores reais e distintos de uma matriz companheira real.

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

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

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

Página

CAPÍTULO O

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

$$Ax = \lambda x \tag{1}$$

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

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

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

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

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

<u>Definição 4</u>: 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.

<u>Definição 5</u>: 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:

$$\left.\begin{array}{l}\text{positiva definida}\\ \text{negativa definida}\\ \text{positiva semi definida}\\ \text{negativa semi definida}\end{array}\right\}\text{de acordo com } x^tAx \left\{\begin{array}{l} > 0\\ < 0\\ \geq 0\\ \leq 0\end{array}\right. \text{, para todo } x \neq 0.$$

Definição 7: A matriz A de uma forma quadrática positiva - (negativa) definida x^tAx é chamada uma matriz positiva (negati - va) definida.

Definição 8: Uma forma quadrática x^tAx 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:

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

são chamadas respectivamente, triangular inferior e triangular superior.

<u>Definição 12</u>: A matriz:

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

é chamada matriz tridiagonal.

<u>Definição 13</u>: 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:

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

é chamada matriz diagonal.

Definição 15: As matrizes:

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

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 $x \in R^n$

CAPÍTULO 1

MÉTODO PROPOSTO:

1.1. Introdução:

Seja a equação matricial:

$$Cx = \lambda x \tag{3}$$

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,

$$Ax = \lambda Bx \tag{4}$$

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:

$$Ax = \lambda L L^t x \quad \text{ou}$$

$$L^{-1} Ax = \lambda L^t x \quad \text{ou ainda}$$

$$L^{-1}A(L^t)^{-1}L^t x = \lambda L^t x.$$

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

$$H = \begin{bmatrix} S_0 & S_1 & \dots & S_{n-1} \\ S_1 & S_2 & \dots & S_n \\ \vdots & \vdots & & \vdots \\ S_{n-1} & S_n & \dots & S_{2n-2} \end{bmatrix} \tag{5}$$

onde $S_i = tr(C^i)$ (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:

$$f(x) = c_{n+1}x^n - c_n x^{n-1} - c_{n-1}x^{n-2} \dots - c_2 x - c_1 , \tag{6}$$

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

e

$$g(x) = b_{m+1}x^m - b_m x^{m-1} - b_{m-1}x^{m-2} \dots - b_2 x - b_1 , \tag{7}$$

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:

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

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

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

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 equipararmos as potências de x em ambos os lados da equação, obteremos:

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

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

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

ou ainda:

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

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:

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

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

<u>Demonstração</u>: Seja:

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

Considere-se agora, a equação matricial:

$$CX = XC^t. \tag{11}$$

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

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

$$c_1x_1 + c_2x_2 + \ldots + c_nx_n = x_nC^t.$$

Eliminando-se $x_2, x_3, \ldots, x_n$, obteremos:

$$x_1 f(C) = [0].$$

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

cial (11) são tais que:

x_1 pode ser escolhida arbitrariamente

$x_2, \ldots, x_n$ satisfazem às relações recursivas

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

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

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

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

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

é fácil ver-se que:

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

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

$$s_0 = n = \sum_{i=1}^{n} \lambda_i^0$$

$$s_1 = c_n = \sum_{i=1}^{n} \lambda_i^1$$

$$s_2 = c_n^2 + 2\, c_{n-1} = \sum_{i=1}^{n} \lambda_i^2$$

$$\vdots$$

etc.

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:
$(\lambda_1^i, \lambda_2^i, \ldots, \lambda_n^i)$, $0 \le i \le n-1$.

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:

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

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:

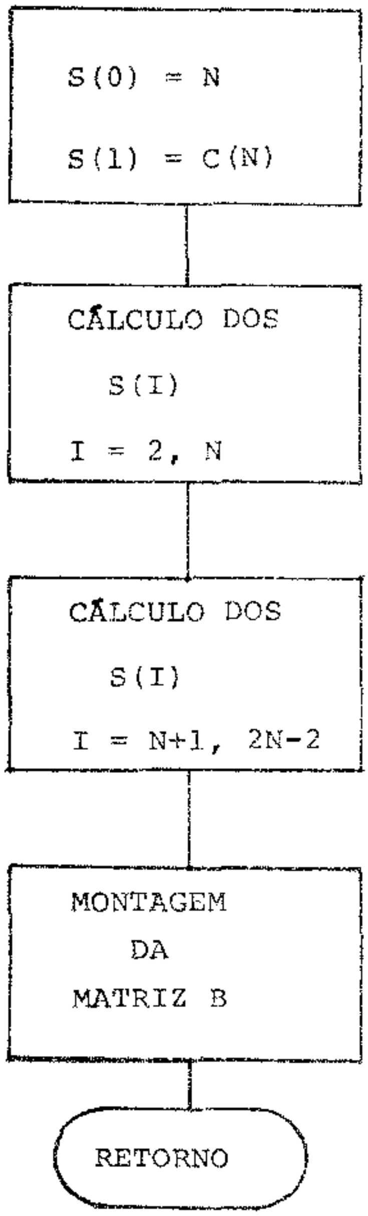
1.3.1. Construção da matriz Hankel:

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

Os s_i são calculados recursivamente por:

$$
\begin{aligned}
s_0 &= n \\
s_1 &= c_n \\
s_2 &= c_n s_1 + 2\, c_{n-1} \\
s_3 &= c_n\, s_2 + c_{n-1} s_1 + 3\, c_{n-2} \\
&\vdots \\
s_{n-1} &= c_n s_{n-2} + c_{n-1} s_{n-3} + \dots + c_3 s_1 + (n-1)\, c_2 \\
s_n &= c_n s_{n-1} + c_{n-1} s_{n-2} + \dots + c_2 s_1 + n\, c_1 \\
s_{n+1} &= c_n s_n + c_{n-1} s_{n-1} + \dots + c_2 s_2 + c_1 s_1 \\
&\vdots \\
s_{2n+2} &= c_n s_{2n-3} + c_{n-1} s_{2n-4} + \dots + c_2 s_{n-1} + c_1 s_{n-2}
\end{aligned}
$$

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.

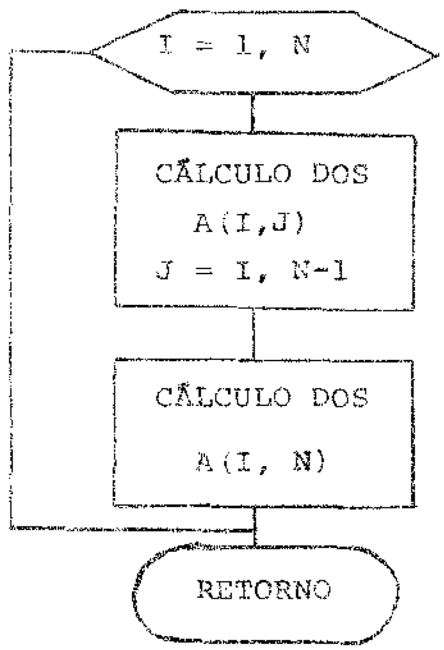
Os a_{ij} são calculados recursivamente por:

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

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

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

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 $k = 2, \ldots , n$

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

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

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

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

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

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 $i = 1, \ldots, n$

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

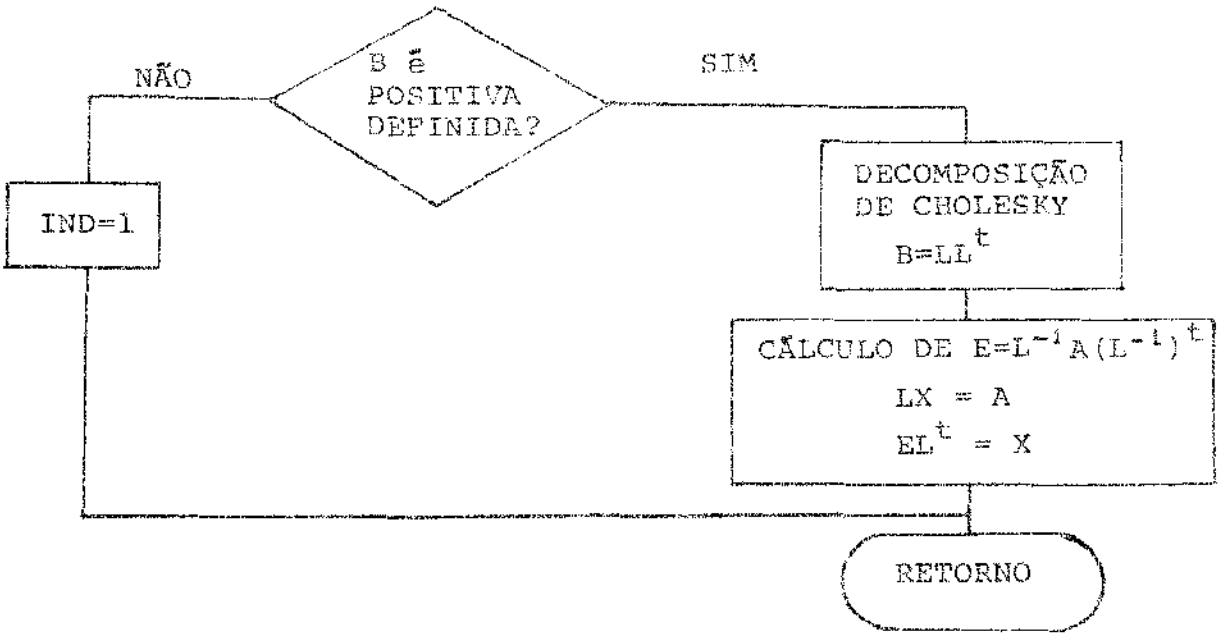
e as equações (13) são dadas por:

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

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

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

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,

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

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:

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

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

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

$$p_i = A_i \, u_i \, / \, H_i$$

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

$$q_i = p_i - K_i \, u_i \, .$$

Finalmente,

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

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

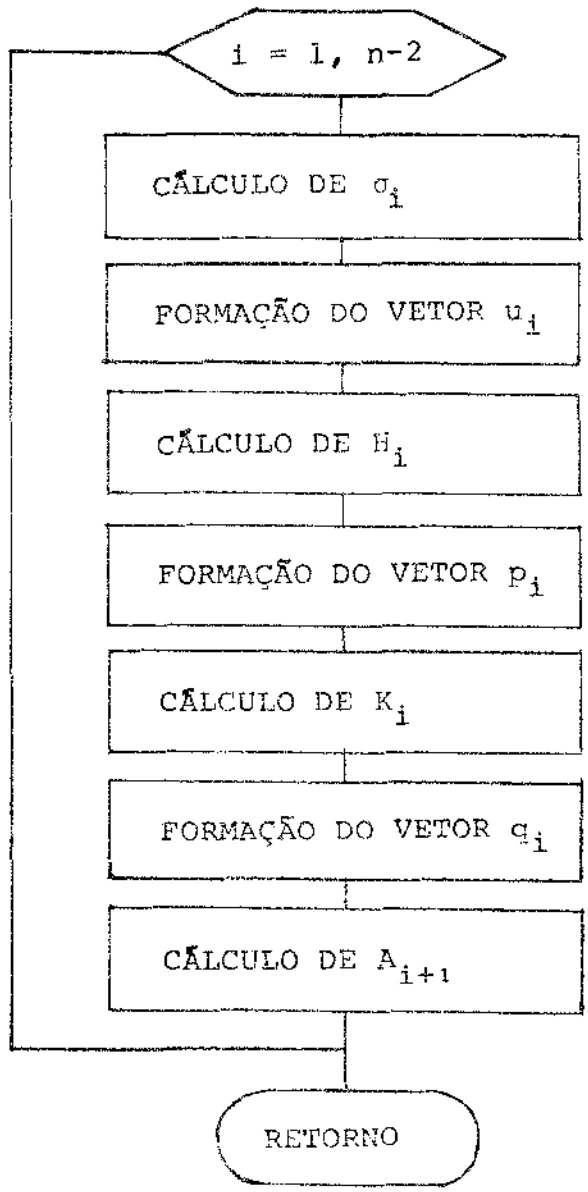
A matriz P_i da i-ésima transformação é da forma:

$$P_i = \left[\begin{array}{c|c} I - 2w_i w_i^t & 0 \\ \hline 0 & I \end{array} \right] \begin{array}{l} \} \; n-i \\ \\ \} \; i \end{array}$$

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

$$A_i = \left[\begin{array}{ccc|cccc} x & x & x & x & & & \\ x & x & x & x & & & \\ x & x & x & x & & & \\ \hline x & x & x & x & x & & \\ & & & x & x & x & \\ & & & & x & x & x \\ & & & & & x & x \end{array} \right] \begin{array}{l} \left.\begin{array}{c} \\ \\ \\ \end{array}\right\} n-i \\ \left.\begin{array}{c} \\ \\ \\ \\ \end{array}\right\} i \end{array}$$

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 aperfeiçoado, 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 $A_1 - \Sigma k_i I$ 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:

$$\left[\begin{array}{cccc|ccc}
x & x & & & & & \\
x & x & x & & & 0 & \\
 & x & x & x & & & \\
 & & x & x & \varepsilon & & \\
\hline
 & & & \varepsilon & x & x & \\
 & 0 & & & x & x & x \\
 & & & & & x & x
\end{array}\right] \begin{array}{l} \\ \\ \\ \\ \leftarrow \text{linha } i \\ \\ \\ \end{array}$$

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 x 2, que está mais em cima e, é próximo do primeiro elemento da diagonal dessa submatriz. Sempre que a submatriz principal 1 x 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:

$$Q_s = P_1^{(s)} \; P_2^{(s)} \; \ldots \; P_{n-1}^{(s)}$$

onde os $P_i^{(s)}$ são determinados na ordem $P_{n-1}^{(s)}, \ldots , P_1^{(s)}$; 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 $d_i^{(s)}$ $(i = 1, \ldots , n)$ os elementos da diagonal principal e $sd_i^{(s)}$ $(i = 1, \ldots , n-1)$ os elementos da subdiagonal superior.

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

$$\left.\begin{array}{l}
p_n = d_n^{(s)} - k_s \\
c_n = 1 \\
s_n = 0 \\
\text{para } i = n-1, \ldots, 1 \\
r_{i+1} = (p_{i+1}^2 + (sd_i^{(s)})^2)^{1/2} \\
g_{i+1} = c_{i+1}\, sd_i^{(s)} \\
h_{i+1} = c_{i+1}\, p_{i+1} \\
sd_{i+1}^{(s+1)} = s_{i+1}\, r_{i+1} \\
c_i = p_{i+1} / r_{i+1} \\
s_i = sd_i^{(s)} / r_{i+1} \\
p_i = c_i (d_i^{(s)} - k_s) - s_i\, g_{i+1} \\
d_{i+1}^{(s+1)} = h_{i+1} + s_i (c_i\, g_{i+1} + s_i (d_i^{(s)} - k_s)) \\
sd_1^{(s+1)} = s_1\, p_1 \\
d_1^{(s+1)} = c_1\, p_1
\end{array}\right\} \quad (14)$$

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

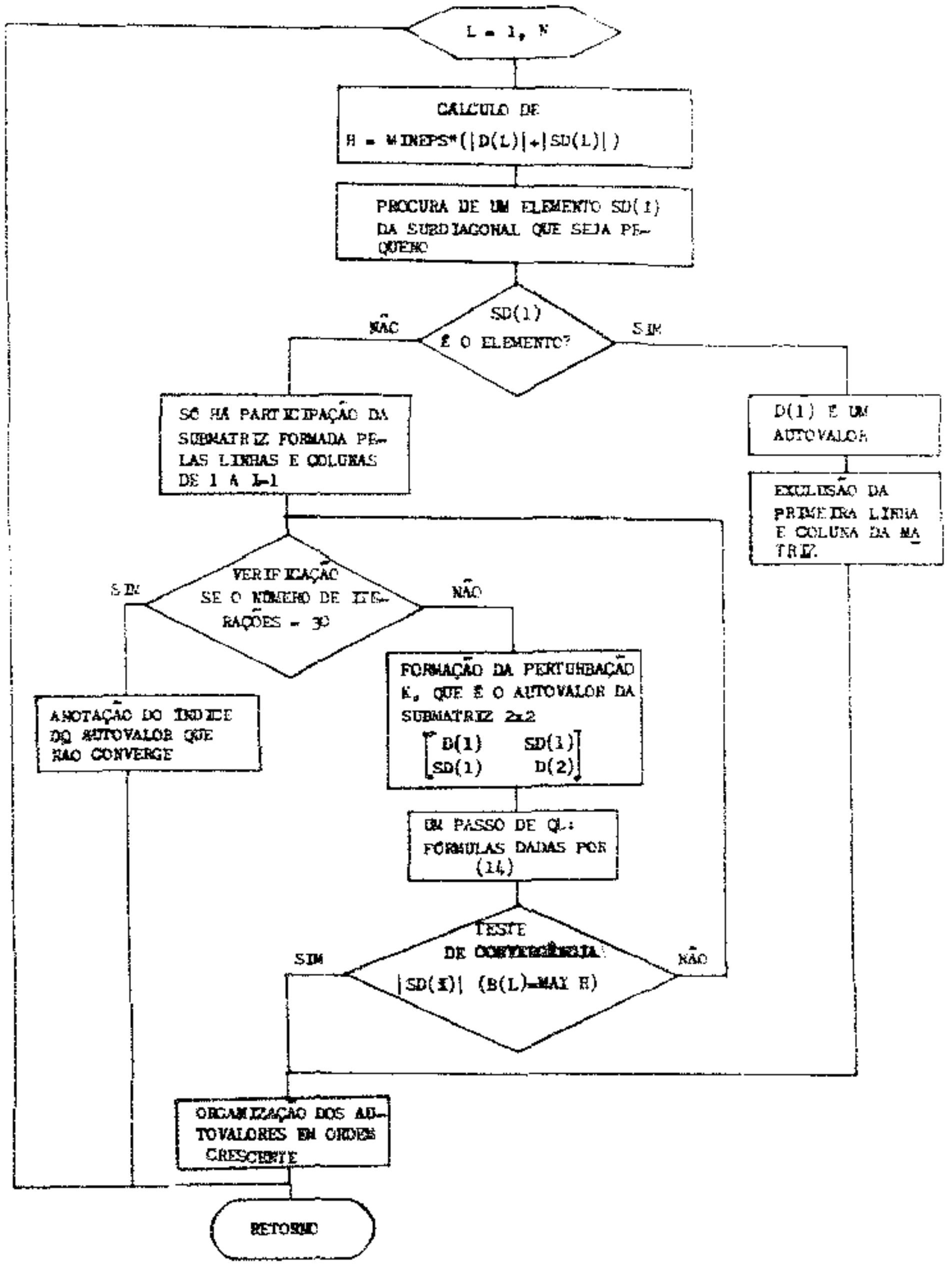
$$h_r = \text{mineps}\,(|\, d_r^{(s)} \,| + |\, sd_r^{(s)} \,|)$$

é computado (mineps é o menor ε tal que $1 + \varepsilon > 1$) e antes de começarmos a iteração, algum $|sd_i^{(s)}|$ menor que $b_r = \max\limits_{i=1,r} h_i$ é considerado como insignificante.

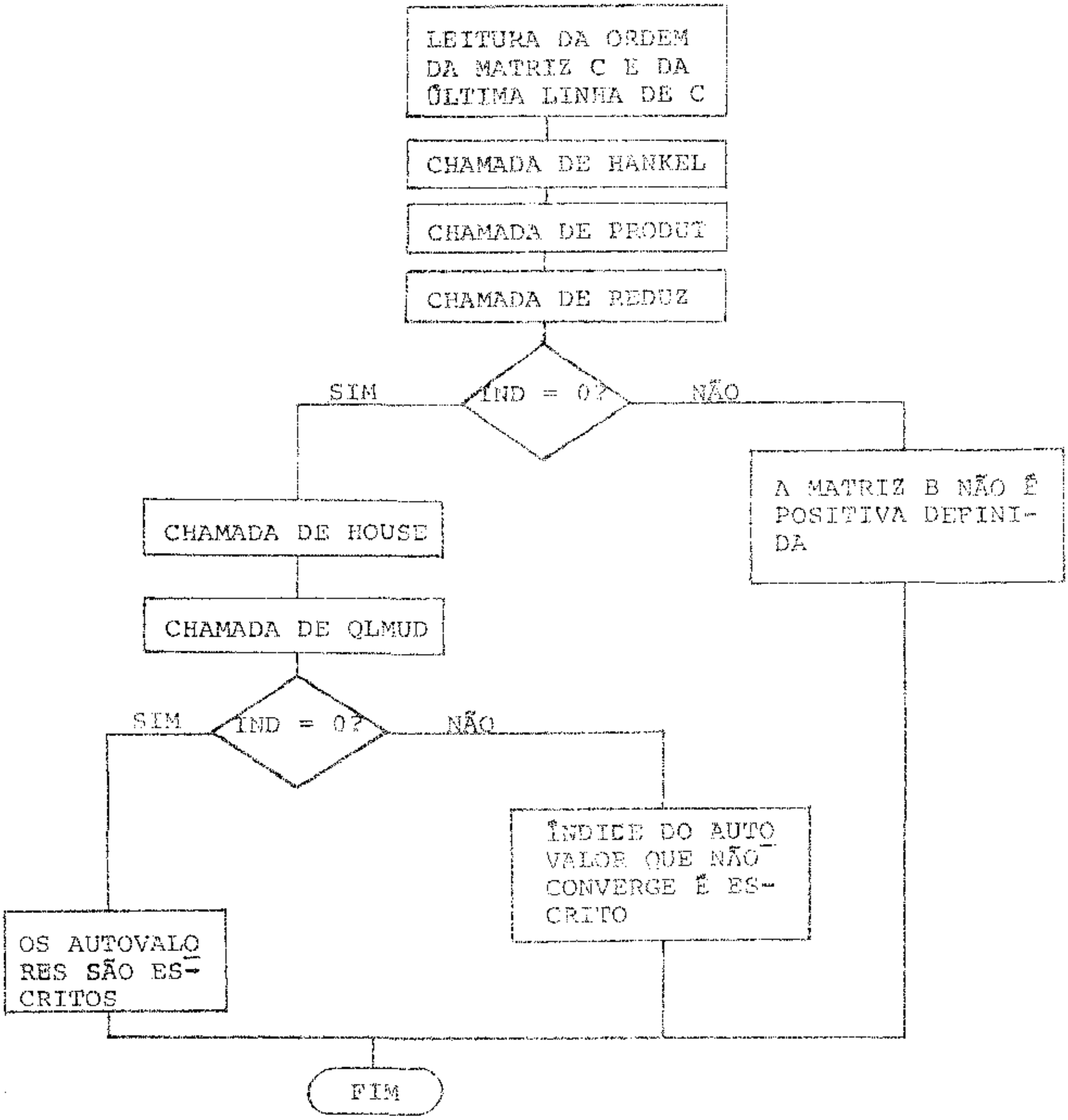
O número de iterações para cada autovalor é limitado - (arbitrariamente para 30).

O uso do QL é preferível ao convencional QR, pois as matrizes tridiagonais resultantes de Householder têm os elementos 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 $EL^t = X$:

$$E = \begin{bmatrix} -0.3333332 & & \\ 1.2472191 & 0.4761903 & \\ -9.6173*10^{-8} & 0.3499273 & -1.1428549 \end{bmatrix} = A_1$$

(iv) Método de Householder:

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

como $n = 3 \longrightarrow i = 1$

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

$u_1^t = (a_{31}; a_{32} \pm \sqrt{\sigma_1}\ ; 0) = (-9.6173*10^{-8}; 0.6998546; 0)$

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

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

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

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

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

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

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

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

(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: $2*10^{-24}$

L = 1

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

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) = $|4.6807*10^{-3}|$

$D(1) = -0.476077*10^{-3}$

IND = 4

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

K = -1.414214

PASSO DE QL

$SD(1) = |9.20869*10^{-10}|$

$D(1) = -0.1396683*10^{-10}$

$\boxed{\lambda_1 = -1.414214}$

L = 2

$H = 0.6228003*10^{-8}$

IND = 1

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

K = -1

PASSO DE QL

$SD(2) = |2.37402*10^{-7}|$

$D(2) = 0.2355201*10^{-7}$

$\boxed{\lambda_2 = -1}$

L = 3

$H = 0.3597459*10^{-7}$

I = 1 → submatriz formada pela linha e coluna 3

D(3) = 1.414214

$\boxed{\lambda_3 = 1.414214}$

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. Balanceamente de uma matriz:

Este algoritmo balanceia uma matriz real geral A e isola 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 $k = 1, 2, \ldots$ 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

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

define $f = f_k$ por $f = \beta^{\sigma}$.

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

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

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

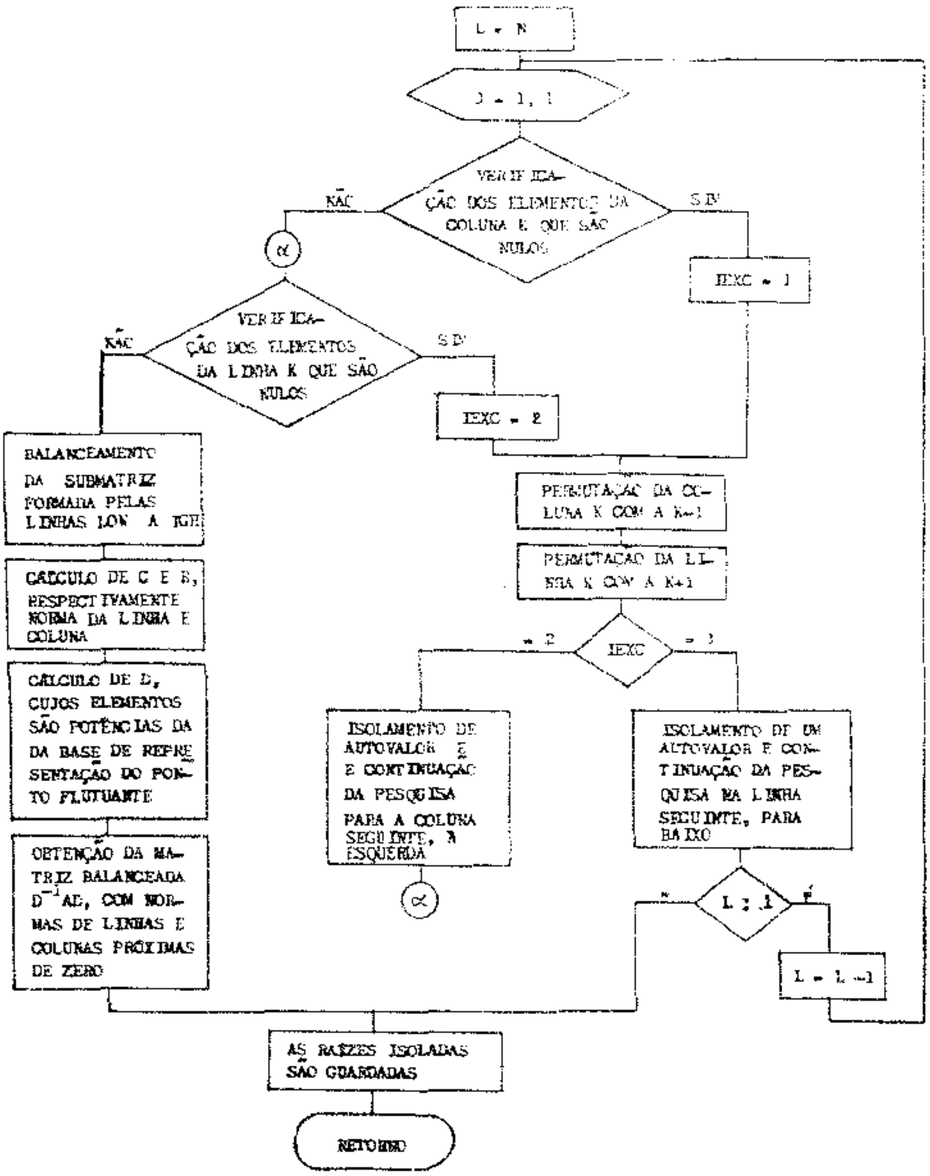
(c) Forma:

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

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

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

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

$$A = QR \quad e$$
$$B = RQ ,$$

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:

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

$$A_{s+1} = R_s Q_s^t + k_s I = Q_s A_s Q_s^t \quad ,$$

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

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

$$a_{i+1,i}^{(s)} \; a_{i+2,i+1}^{(s)} = 0 \qquad (i = 1, \ldots , n-2) \; ,$$

ou seja,

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

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

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

Teorema: O algoritmo básico QR aplicado a matrizes de Hessenberg irredutíveis 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 impar.

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 x 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:

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

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:

$$A_s = \left[\begin{array}{ccc|cccc} x & x & x & x & x & x & x \\ x & x & x & x & x & x & x \\ & x & x & x & x & x & x \\ \hline & & \varepsilon & x & x & x & x \\ & & & x & x & x & x \\ & & & & x & x & x \\ & & & & & x & x \end{array}\right] \begin{array}{l} \\ \\ \\ \leftarrow \text{linha } \ell \\ \\ \\ \\ \end{array}$$

Sempre que uma submatriz 1×1 ou 2×2 finalmente se separa do resto da matriz, os autovalores dessa submatriz serão 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 x 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 $P_1, P_2, \ldots, P_{n-1}$, 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 $Q = P_1\, P_2 \ldots P_{n-1}$.

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:

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

e

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

com $\sigma_s^2 = p_s^2 + q_s^2 + r_s^2$ 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:

$$A_s = \left[\begin{array}{cc|ccccc} x & x & x & x & x & x & x \\ x & x & x & x & x & x & x \\ & \varepsilon_1 & x & x & x & x & x \\ \hline & & \varepsilon_2 & x & x & x & x \\ & & & x & x & x & x \\ & & & & x & x & x \\ & & & & & x & x \end{array}\right] \leftarrow \text{linha } m$$

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 ele - mentos diferentes de zero, nas posições (m+1, m-1),(m+2, m-1) , quando aplicada à matriz cheia. Podemos ilustrar por:

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

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_1 para a matriz de Hessenberg A_s, onde

$$a^{(s)}_{i+1,i}\, a^{(s)}_{i+1,i+1} = 0 \qquad (\, i = 1, \ldots , n-2).$$

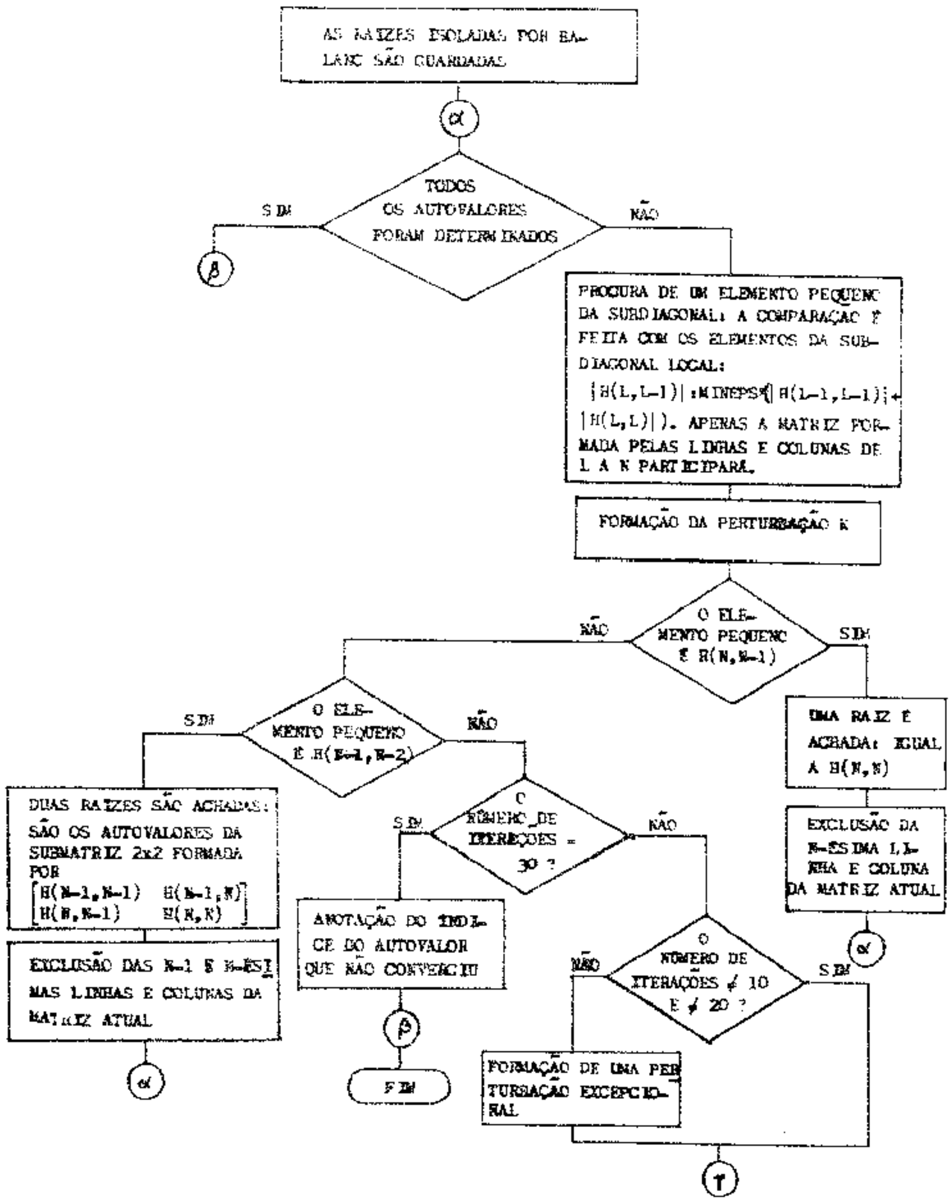
Se, depois de 10 iterações, nenhum autovalor for determinado, as perturbações k_1 e k_2 serão substituídas, na próxima iteração, por perturbações definidas por:

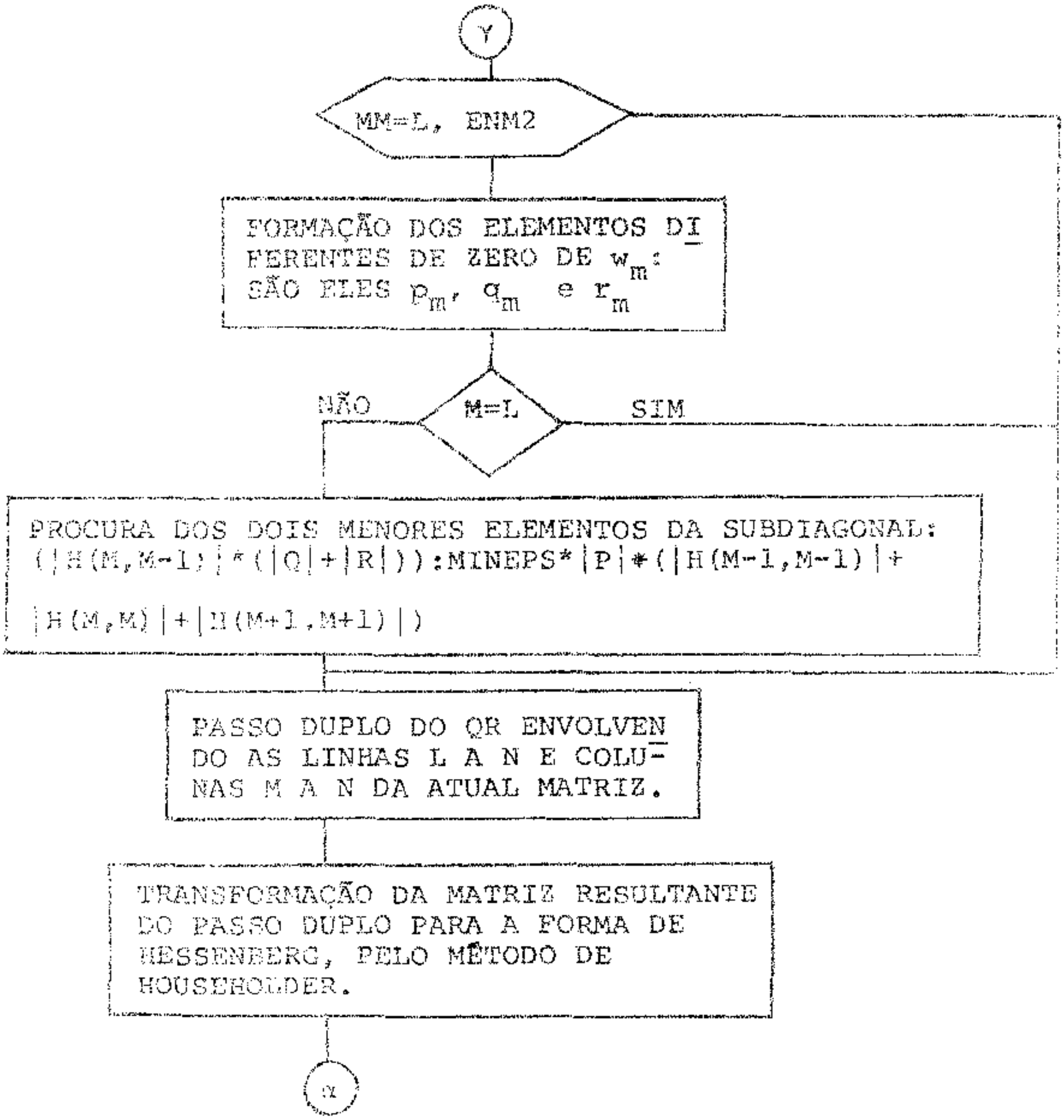
$$k_1 + k_2 = 1,5 * (|a_{n,n-1}| + |a_{n-1,n-2}|),$$

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

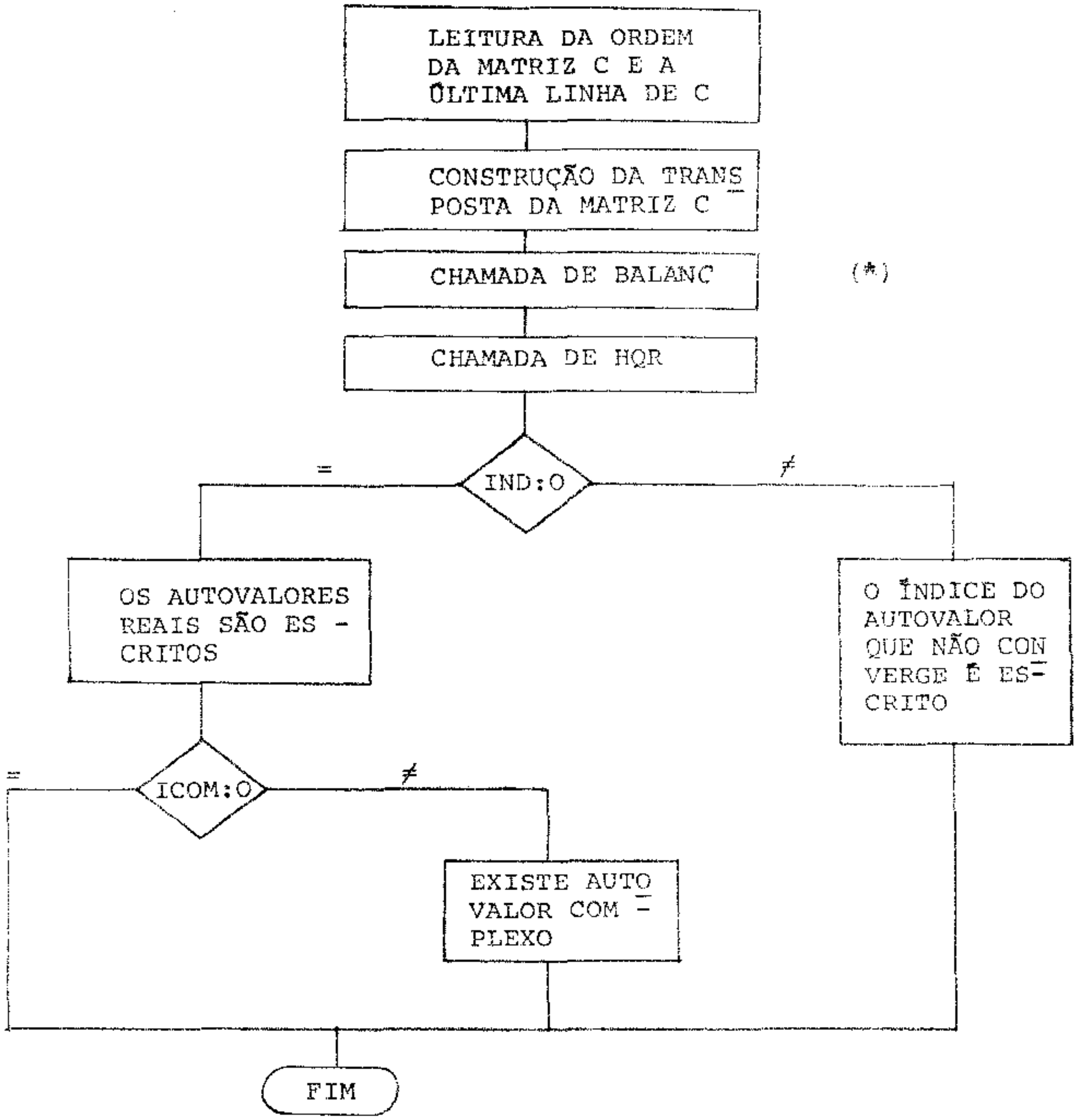
Essa estratégia é usada novamente depois de 20 iterações. Se 30 iterações são usadas, então, a indicação de falha de convergência é dada.

DIAGRAMA DE BLOCOS:




γ
MM=L, ENM2
FORMAÇÃO DOS ELEMENTOS DIFERENTES DE ZERO DE w_m: SÃO ELES p_m, q_m e r_m
NÃO
M=L
SIM
PROCURA DOS DOIS MENORES ELEMENTOS DA SUBDIAGONAL: (|H(M,M-1)|*(|Q|+|R|)):MINEPS*|P|*(|H(M-1,M-1)|+|H(M,M)|+|H(M+1,M+1)|)
PASSO DUPLO DO QR ENVOLVENDO AS LINHAS L A N E COLUNAS M A N DA ATUAL MATRIZ.
TRANSFORMAÇÃO DA MATRIZ RESULTANTE DO PASSO DUPLO PARA A FORMA DE HESSENBERG, PELO MÉTODO DE HOUSEHOLDER.
α

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:

$$\beta = 2$$

$$\gamma = 0,95$$

C_i = norma L_1 da coluna i

R_i = norma L_1 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 $\begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 2 \end{bmatrix}$

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

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

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

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

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

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

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

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

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

1a. linha * $\frac{1}{f}$ = 0.5 (0; 0; 2) = (0; 0; 1)

1a. coluna * f = 2 $\begin{bmatrix} 0 \\ 0.5 \\ 0 \end{bmatrix} = \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix}$

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

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

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

$c = 0 \rightarrow f = 1$

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

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

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

$c = 0 \rightarrow f = 1$

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

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

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

$c = 0 \rightarrow f = 1$

$C_1 * f + \dfrac{R_1}{f} = 2 > \gamma * (C_1 + R_1) = 1.9$

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

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

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

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

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

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

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

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

(ii) QR para matrizes reais de Hessenberg superior:

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

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

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

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

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

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

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

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

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:

$$c_1 = (\text{sinal de } p_1)\ \sqrt{p_1^2 + q_1^2 + r_1^2} = 0.6$$

$$u_1 = \left(\frac{p_1 + c}{c} , \frac{q_1}{c} ; \frac{r_1}{c} \right) = (1.666667; -0.3333333; -0.6666667)$$

$$v_1 = \left(1; \frac{q_1}{p_1 + c} ; \frac{r_1}{p_1 + c} \right) = (1; -0.2; -0.4)$$

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

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

$$p_2 = \frac{a_{21}}{|p| + |q| + |r|} = 0.3333333$$

$$q_2 = \frac{a_{31}}{|p| + |q| + |r|} = 0.6666667$$

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

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

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

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

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

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

$|a_{32}| = 1.2 > \text{MINEPS} * (|a_{33}| + |a_{22}|) = 2.244445*10^{-8}$

$|a_{21}| = 0.4969040 > \text{MINEPS} * (|a_{22}| + |a_{11}|) = 1.288889*10^{-8}$

$L = 1 \rightarrow$ 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:

$p_3 = 0.6771855$

$q_3 = 0.1431033$

$r_3 = 0.1797112$

PASSO DUPLO DO QR:

$\sigma_3 = 0.7159808$

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

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

$P_3 = I - u_3 v_3$

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

$p_4 = -0.4580274$

$q_4 = -0.5419726$

$r_4 = 0$

$\sigma_4 = -0.7095938$

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

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

$$P_4 = I - u_4 . v_4^t$$

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

$|a_{32}| = 1.164254 > \text{MINEPS} * (|a_{33}| + |a_{22}|) = 2.042201*10^{-8}$

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

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

$k_5 + k_5 = -0.1569206$

$k_5 * k_6 = -2.2465883$

<u>3a. ITERAÇÃO:</u>

$p_5 = -0.9004537$

$q_5 = 0.01797077$

$r_5 = 0.08157557$

PASSO DUPLO DO QR:

$\sigma_5 = -0.9043198$

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

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

$P_5 = I - u_5 v_5^t$

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

$p_6 = -0.3930739$

$q_6 = 0.6069261$

$r_6 = 0$

$\sigma_6 = -0.7230950$

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

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

$P_6 = I - u_6 v_6^t$

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

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

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

$L = 1 \rightarrow$ 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:

$\sigma_7 = -0.9735893$

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

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

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

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

$p_8 = -0.8686204$

$q_8 = -0.1313796$

$r_8 = 0$

$\sigma_8 = -0.8784998$

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

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

$$P_8 = I - u_8 \, v_8^t$$

$$A_8 = P_8 A_7 P_8 = \begin{bmatrix} -0.9953685 & -1.706220 & -0.2914317 \\ 0.001137156 & -1.418844 & 0.001628423 \\ 0 & 0.002424318 & 1.414212 \end{bmatrix}$$

$|a_{32}| = 0,002424318 > \text{MINEPS} * (|a_{33}| + |a_{22}|) = 2,833056*10^{-6}$

$|a_{21}| = 0,001137156 > \text{MINEPS} * (|a_{22}| + |a_{11}|) = 2,414212*10^{-6}$

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

$k_9 + k_{10} = -0,004632$

$k_9 * k_{10} = -2.00655016$

5a. ITERAÇÃO:

$p_9 = -0.9973243$

$q_9 = -0,002673017$

$r_9 = 0.2689366*10^{-5}$

PASSO DUPLO DO QR:

$\sigma_9 = -0.9973279$

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

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

$$P_9 = I - u_9 \, v_9^t$$

$$A_9 = P_9 A_8 P_9 = \begin{bmatrix} -0.9999407 & 1.707342 & 0.2914327 \\ -1.414271 & -0.1441242*10^{-4} & 0.002409517 \\ 0.002419716 & 0.1232914*10^{-7} & 1.414211 \end{bmatrix}$$

$p_{10} = -0.9991453$

$q_{10} = \;\; 0.8547211*10^{-3}$

$r_{10} = \;\; 0$

$\sigma_{10} = -0.9991456$

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

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

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

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

$|a_{32}| = 0.823347*10^{-7} > MINEPS*(|a_{33}|+|a_{22}|) = 2.828487*10^{-8}$

$|a_{21}| = 0.1441243*10^{-4} > MINEPS*(|a_{22}|+|a_{11}|) = 2.414214*10^{-8}$

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

6a. ITERAÇÃO:

$p_{11} = -0.9999652$

$q_{11} = -0.3478264*10^{-4}$

$r_{11} = \;\; 0.1186262*10^{-11}$

PASSO DUPLO DO QR:

$\sigma_{11} = -0.9999652$

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

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

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

$$A_{11} = P_{11} A_{10} P_{11} = \begin{bmatrix} -1 & 1.707107 & -0.2928932 \\ -0.2420165*10^{-8} & -1.414214 & -0.7099618*10^{-7} \\ -0.704189*10^{-16} & -0.8233271*10^{-7} & 1.414214 \end{bmatrix}$$

$p_{12} = -1$

$q_{12} = -0.2909674$

$r_{12} = 0$

$c_{12} = -1$

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

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

$$P_{12} = I - u_{12}\, v_{12}^{t}$$

$$A_{12} = P_{12} A_{11} P_{12} = \begin{bmatrix} -1 & -1.707107 & -0.2928932 \\ 0.2420165*10^{-8} & -1.414214 & -0.1130184*10^{-7} \\ 0 & 0.3469358*10^{-10} & 1.414214 \end{bmatrix}$$

$|a_{32}| = 0.3469358*10^{-10} < MINEPS*(|a_{33}|+|a_{22}|) = 2.828427*10^{-8}$

$\boxed{\lambda_1 = 1.414214}$

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

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

$\boxed{\lambda_2 = -1.414214}$

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(0)	0	0	0.13
(2; 1; -2)	DATTA	0	0	0.22
	QR(6)	0	0.1	0.23
(1.5; 4.75; 3.25; -1)	DATTA	10^{-6}	0.1	0.16
	QR(1)	10^{-7}	0.1	0.22
(-24; 50; -35; 10)	DATTA	10^{-5}	0	0.15
	QR(7)	10^{-6}	0	0.18
(-5.2625; -15.5; -7.125;	DATTA	10^{-1}	0	0.15
6.25; 2,5)	QR(3)	10^{-1}	0.2	0.23
(-720; 1764; -1624; 735;	DATTA	10^{-2}	0.1	0.17
-175; 21)	QR(1)	10^{-4}	0.2	0.32
(-1.44x10^{-5};3.6x10^{-5};1.96x10^{-2};	DATTA	10^{-6}	0.1	0.22
-4.9x10^{-3};-5.6x10^{-4};0.14; 0.4)	QR(1)	10^{-7}	0.2	0.36
(11.025x10^{-5};0;-13.02625x10^{-3};0;	DATTA	10^{-4}	0.2	0.32
21.0316x10^{-2};0;-1.0374;0;184;0)	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       ESSA SUBROTINA CONSTROI A MATRIZ DE HANKEL DAS SO-
C       MAS DE NEWTON.
C       N E UMA VARIAVEL INTEIRA DE ENTRADA.
C       C E O VETOR REAL DE ORDEM N QUE CONTEM OS ELEMEN-
C       TOS DA ULTIMA LINHA DA MATRIZ COMPANHEIRA.
C       B E A MATRIZ SIMETRICA REAL POSITIVA DEFINIDA DE
C       ORDEM N, CONSTRUIDA A PARTIR DAS SOMAS DE NEWTON.
C       APENAS O TRIANGULO SUPERIOR CHEIO DA MATRIZ B E
C       FORNECIDO.
C       S E O VETOR REAL DAS SOMAS DE NEWTON DE ORDEM
C       2N-2, CONSTRUIDO A PARTIR DE RELACOES RECURSIVAS.

        SUBROUTINE HANKEL(N)
        COMMON /BL1/A(100,100),B(100,100),C(100)
        DIMENSION S(0/99)
C       CONSTRUCAO DAS SOMAS DE NEWTON
        S(0)=N
        S(1)=C(N)
        DO 200 I=2,N
        CS=0.
        L=N+1
        DO 100 K=I-1,1,-1
        L=L-1
100     CS=CS+C(L)*S(K)
        S(I)=CS+I*C(N-I+1)
200     CONTINUE
        J=2*N-2
        DO 400 I=N+1,J
        S(I)=0.
        K=I
        DO 300 L=N,1,-1
        K=K-1
300     S(I)=S(I)+C(L)*S(K)
400     CONTINUE
C       MONTAGEM DA MATRIZ DE HANKEL
        DO 500 I=1,N
        DO 500 J=I,N
500     B(I,J)=S(I+J-2)
        RETURN
        END
```

```
C         4.2. PRODUT:

C         SUBROTINA PRODUT
C         ESSA SUBROTINA CONSTROI A MATRIZ SIMETRICA REAL A
C         A PARTIR DO PRODUTO DA MATRIZ DE HANKEL B PELA
C         TRANSPOSTA DA MATRIZ COMPANHEIRA REAL.
C         N E UMA VARIAVEL INTEIRA DE ENTRADA.
C         A E A MATRIZ SIMETRICA REAL DE ORDEM N.APENAS O
C         TRIANGULO SUPERIOR CHEIO DA MATRIZ E FORNECIDO.
C         B E A MATRIZ SIMETRICA REAL POSITIVA DEFINIDA DE
C         ORDEM N. APENAS O TRIANGULO SUPERIOR CHEIO DA MA-
C         TRIZ PRECISA SER FORNECIDO.
C         C E UM VETOR REAL DE ORDEM N QUE CONTEM OS ELEMEN-
C         TOS DA ULTIMA COLUNA DA MATRIZ COMPANHEIRA.

          SUBROUTINE PRODUT(N)
          COMMON /BLIZ/A(100,100),B(100,100),C(100)
C         CONSTRUCAO DA MATRIZ A IGUAL AO PRODUTO DA MATRIZ
C         B PELA TRANSPOSTA DA MATRIZ COMPANHEIRA.
          DO 400 I=1,N
          DO 100 J=1,N-1
100       A(I,J)=B(I,J+1)
          BC=0.
          DO 200 K=1,I
200       BC=BC+B(K,I)*C(K)
          AC=0.
          DO 300 L=I+1,N
300       AC=AC+B(I,L-1)*C(L)
          A(I,N)=BC+AC
400       CONTINUE
          RETURN
          END
```

```
C         4.3. REDUZ:

C         SUBROTINA REDUZ
C         ESSA SUBROTINA REDUZ O PROBLEMA SIMETRICO GENERALIZADO
C         DE AUTOVALORES AX=(LAMBDA)BX, ONDE B E POSITIVA DEFINI-
C         DA, PARA O PROBLEMA SIMETRICO PADRAO DE AUTOVALORES
C         PZ=(LAMBDA)Z, USANDO A FATORIZACAO DE CHOLESKY EM B.
C         N E UMA VARIAVEL INTEIRA DE ENTRADA.
C         A E A MATRIZ SIMETRICA REAL DE ORDEM N DO SISTEMA.
C         APENAS O TRIANGULO SUPERIOR CHEIO DA MATRIZ PRECISA
C         SER FORNECIDO. NA SAIDA, O TRIANGULO INFERIOR CHEIO
C         DE A CONTEM O TRIANGULO INFERIOR CHEIO DA MATRIZ
C         SIMETRICA OBTIDA DA REDUCAO PARA A FORMA PADRAO.
C         O TRIANGULO SUPERIOR DE A NAO E ALTERADO.
C         B E A MATRIZ SIMETRICA REAL POSITIVA DEFINIDA DE OR-
C         DEM N DO SISTEMA. APENAS O TRIANGULO SUPERIOR CHEIO
C         DA MATRIZ PRECISA SER FORNECIDO. NA SAIDA, O TRIAN-
C         GULO INFERIOR DE B CONTEM O TRIANGULO INFERIOR DO
C         FATOR CHOLESKY L. O TRIANGULO SUPERIOR CHEIO DE
C         [illegible] ALTERADO.
C         [illegible] DE ORDEM N [illegible]
C         [illegible] PRINCIPAL DO FATOR CHOLESKY [illegible].
C         [illegible] VARIAVEL INTEIRA DE SAIDA IGUAL A [illegible]
C         [illegible] POSITIVA DEFINIDA. NESSE CASO, A FATORIZACAO
C         DE CHOLESKY [illegible] NAO EXISTE. NORMALMENTE [illegible]
C         [illegible] ZERO.

          SUBROUTINE REDUZ(N)
          COMMON /AUX/A(100,100),B(100,100),DPL(100)
          COMMON /AUX2/D(100),DSO(100),[illegible]
C         CONSTRUCAO DE L NO TRIANGULO INFERIOR DE B E NO VETOR
C         DPL PELA FATORIZACAO DE CHOLESKY.
          [illegible]=0
          DO 40 I=1,N
          I1=I-1
          DO 40 J=I,N
          X=B(I,J)
          IF(I.EQ.1) GO TO 20
          DO 10 K=1,I1
10        X=X-B(I,K)*B(J,K)
20        IF(J.NE.I) GO TO 30
C         VERIFICA SE B E POSITIVA DEFINIDA
          IF(X.LE.0.) GO TO 130
          Y=SQRT(X)
          DPL(I)=Y
          GO TO 40
30        B(J,I)=X/Y
40        CONTINUE
C         CONSTRUCAO DA TRANSPOSTA DE INV(L)*A NO TRIANGULO
C         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
50    X=X-A(I,K)*A(J,K)
60    A(J,I)=X/Y
70    CONTINUE
C     PRE-MULTIPLICACAO POR INV(L)
      DO 120 J=1,N
      J1=J-1
      DO 120 I=J,N
      X=A(I,J)
      IF(I.EQ.J) GO TO 90
      I1=I-1
      DO 80 K=J,I1
80    X=X-A(K,J)*B(I,K)
90    IF(J.EQ.1) GO TO 110
      DO 100 K=1,J1
100   X=X-A(J,K)*B(I,K)
110   A(I,J)=X/DPL(I)
120   CONTINUE
      GO TO 140
C     INDICACAO DE QUANDO B NAO E POSITIVA DEFINIDA.
130   NPD=1
140   RETURN
      END
```

```
C         4.4.  HOUSE:

C         SUBROTINA HOUSE
C         ESSA SUBROTINA REDUZ UMA MATRIZ REAL E SIMETRICA A UMA
C         MATRIZ TRIDIAGONAL SIMETRICA T USANDO AS TRANSFORMACOES
C         ORTOGONAIS SIMILARES DO METODO DE HOUSEHOLDER.
C         N E UMA VARIAVEL INTEIRA DE ENTRADA.
C         A E NA ENTRADA UMA MATRIZ REAL E SIMETRICA DE ORDEM N
C         QUE SERA REDUZIDA PARA A FORMA TRIDIAGONAL. APENAS O
C         TRIANGULO INFERIOR CHEIO DA MATRIZ PRECISA SER FORNECIDO.
C         NA SAIDA, O TRIANGULO INFERIOR DE A CONTEM INFORMACOES
C         SOBRE AS TRANSFORMACOES ORTOGONAIS USADAS NA REDUCAO.
C         O TRIANGULO SUPERIOR CHEIO DA MATRIZ A FICA INALTERADO.
C         D E UM VETOR REAL DE DIMENSAO N QUE CONTEM OS ELEMEN-
C         TOS DA DIAGONAL PRINCIPAL DA MATRIZ TRIDIAGONAL T.
C         SD E UM VETOR REAL DE DIMENSAO N QUE CONTEM NAS SUAS
C         ULTIMAS N-1 POSICOES OS ELEMENTOS DA SUBDIAGONAL DA
C         MATRIZ TRIDIAGONAL T. O ELEMENTO SD(1) E ZERO.
C         QSD E UM VETOR REAL DE DIMENSAO N QUE CONTEM NAS SUAS
C         ULTIMAS N-1 POSICOES [illegible]
C         SUBDIAGONAL DA MATRIZ TRIDIAGONAL T. O ELEMENTO QSD(1)
C         E ZERO. [illegible]

          SUBROUTINE HOUSE(N)
          COMMON /HL1/A(100,100),B(100,100),SR(100)
          COMMON /HL2/D(100),QSD(100),[illegible]
C         GUARDA OS ELEMENTOS DA DIAGONAL PRINCIPAL DA MATRIZ
C         A NO VETOR D.
          DO 10 I=1,N
10        D(I)=A(I,I)
C         COMECA A ITERACAO DE N-2 PASSOS
          DO 140 I=N,1,-1
          L=I-1
          H=0.
          SCALE=0.
          IF(L.LT.1) GO TO 30
          DO 20 K=1,L
20        SCALE=SCALE+ABS(A(I,K))
          IF(SCALE.NE.0.) GO TO 40
30        SD(I)=0.
          QSD(I)=0.
          GO TO 130
40        DO 50 K=1,L
          A(I,K)=A(I,K)/SCALE
          H=H+A(I,K)*A(I,K)
50        CONTINUE
C         FORMACAO DO VETOR SIGMA
          QSD(I)=SCALE*SCALE*H
          F=A(I,L)
          G=-SIGN(SQRT(H),F)
          SD(I)=SCALE*G
```

```
      H=H-F*G
C     FORMACAO DO ELEMENTO DO VETOR U NA POSICAO (L,L-1)
      A(I,L)=F-G
      IF(L.EQ.1) GO TO 1[illegible]
      F=0.
      DO 90 J=1,L
      G=0.
C     CONSTRUCAO DE A*U
      DO 60 K=1,J
60    G=G+A(J,K)*A(I,K)
      JP1=J+1
      IF(L.LT.JP1) GO TO 80
      DO 70 K=JP1,L
70    G=G+A(K,J)*A(I,K)
C     CONSTRUCAO DE P
80    D(J)=G/H
      F=F+D(J)*A(I,J)
90    CONTINUE
C     CONSTRUCAO DE [illegible]
      HH=F/(H+H)
C     [illegible]
      DO 1[illegible] J=1,L
      F=A(I,J)
C     [illegible]
      G=D(J)-HH*F
      D(J)=G
      DO 1[illegible] K=1,J
      A(J,K)=A(J,K)-F*D(K)-G*A(I,K)
1[illegible] CONTINUE
1[illegible] DO 1[illegible] I=1,[illegible]
1[illegible] A(I,K)=[illegible]
C     RECUPERACAO DA DIAGONAL PRINCIPAL DA MATRIZ A ORIGINAL
1[illegible] H=D(I)
      D(I)=A(I,I)
      A(I,I)=H
1[illegible] CONTINUE
      RETURN
      END
```

```
C         4.5. QLRAT;

C         SUBROTINA QLRAT
C         ESSA SUBROTINA DETERMINA OS AUTOVALORES DE UMA MATRIZ
C         TRIDIAGONAL SIMETRICA T, USANDO UMA VARIACAO RACIO-
C         NAL DO METODO QL.
C         N E UMA VARIAVEL INTEIRA DE ENTRADA.
C         D E UM VETOR REAL DE DIMENSAO N QUE NA ENTRADA CON-
C         TEM OS ELEMENTOS DA DIAGONAL PRINCIPAL DA MATRIZ
C         TRIDIAGONAL T. NA SAIDA, ELE CONTEM OS AUTOVALORES
C         DESSA MATRIZ EM ORDEM CRESCENTE.
C         QSD E UM VETOR REAL DE DIMENSAO N QUE CONTEM NAS
C         SUAS ULTIMAS N-1 POSICOES O QUADRADO DOS ELEMEN-
C         TOS DA SUBDIAGONAL DA MATRIZ TRIDIAGONAL T. O ELE-
C         MENTO QSD(1) E ARBITRARIO. QLRAT DESTROI QSD.
C         ITER E UMA VARIAVEL INTEIRA DE SAIDA. SE MAIS DE
C         30 ITERACOES SAO REQUERIDAS PARA A DETERMINACAO
C         DE UM AUTOVALOR, QLRAT TERMINA COM ITER QUE INDI-
C         CA O INDICE DO AUTOVALOR PARA O QUAL A FALHA
C         OCORREU. SE TODOS OS AUTOVALORES SAO DETERMINA-
C         DOS COM MENOS DE 30 ITERACOES ITER E IGUAL A ZERO.
C         MINEPS E UMA CONSTANTE DEPENDENTE DO COMPUTADOR QUE
C         CORRESPONDE A PRECISAO RELATIVA DO PONTO FLUTUAN-
C         TE ARITMETICO, OU SEJA, MINEPS E O MENOR EPSILON
C         TAL QUE 1+EPSILON>1. PARA O PDP-10, O VALOR DE
C         MINEPS E IGUAL A 2.**-26.

          SUBROUTINE QLRAT(N)
          COMMON /MAT/ D(100),QSD(100),ITER
          REAL MINEPS
          MINEPS=2.**-26
          ITER=0
C         MUDANCA NO VETOR QSD. O ULTIMO ELEMENTO PASSA A SER
C         IGUAL A ZERO.
          DO 1 I=2,N
1         QSD(I-1)=QSD(I)
          F=0.
          B=0.
          QSD(N)=0.
C         L INDICA O AUTOVALOR QUE ESTA SENDO DETERMINADO.
          DO 120 L=1,N
          J=0
          H=MINEPS*(ABS(D(L))+SQRT(QSD(L)))
          IF(B.GT.H) GO TO 20
          B=H
          C=B*B
C         PROCURA UM ELEMENTO DO QUADRADO DA SUBDIAGONAL QUE SEJA
C         PEQUENO.
20        DO 30 M=L,N
          IF(QSD(M).LE.C) GO TO 40
30        CONTINUE
```

```
4       IF(M.EQ.L) GO TO 80
5       IF(J.EQ.30) GO TO 1[illegible]0
        J=J+1
C       FORMACAO DA PERTURBACAO K
        L1=L+1
        S=SQRT([illegible](L))
        G=D(L)
        P=(D(L1)-G)/(2.*S)
        R=SQRT(P*P+1.)
        D(L)=S/(P+SIGN(R,P))
        H=G-D(L)
C       MODIFICACAO DOS ELEMENTOS DA DIAGONAL.
        DO 6 I=L1,N
6       D(I)=D(I)-H
C       [illegible] A PERTURBACAO K QUE IRA NORMALIZAR OS
C       [illegible] FINAL DA SUBROTINA.
        F=F+H
C       [illegible]
        [illegible]
C       PARA EVITAR CICLAGEM INDEFINIDA, QUANDO G E IGUAL A
C       ZERO, SUBSTITUI-SE POR UM DELTA EQUIVALENTE A PRE-
C       [illegible]
        [illegible]
        [illegible]
        [illegible]
        DO 7[illegible] I=[illegible]-1,L,-1
        [illegible]
        [illegible]
        [illegible](I+1)=[illegible]
        [illegible](I)/[illegible]
        [illegible](I+1)=[illegible](I))
        [illegible](I)-[illegible](I)/[illegible]
C       [illegible] IGUAL A ZERO, SUBSTITUI-SE PELO DELTA
        IF([illegible]) G=B
        [illegible]
7       [illegible]
        [illegible](L)=[illegible]
        [illegible](L)=[illegible]
C       PROTECAO CONTRA DIVISAO POR ZERO NO TESTE DE CON-
C       VERGENCIA.
        IF([illegible]) GO TO 80
        IF(ABS([illegible](L)).LE.ABS(C/H)) GO TO 80
        [illegible](L)=[illegible](L)
        IF([illegible](L).[illegible]) GO TO 50
C       REGULARIZACAO DOS RESULTADOS
8       P=D(L)+F
C       ORGANIZACAO DOS AUTOVALORES EM ORDEM CRESCENTE
        IF(L.EQ.1) GO TO 100
        DO 90 I=L,2,-1
        IF(P.GE.D(I-1)) GO TO 110
        D(I)=D(I-1)
9       CONTINUE
```

```
100       I=1
110       D(I)=P
120       CONTINUE
          GO TO 140
C         INDICACAO DE QUANDO NAO HA CONVERGENCIA PARA ALGUM
C         AUTOVALOR
130       IERR=L
140       RETURN
          END
```

```
C         4.6. [illegible]

C         METODO [illegible]
C         ESSE METODO COMPUTA OS AUTOVALORES DE UMA MATRIZ
C         REAL COMPANHEIRA C.
C         NA SUBROTINA HANKEL, ATRAVES DE RELACOES RECUR-
C         SIVAS CONSTRUIMOS A MATRIZ SIMETRICA POSITIVA DE-
C         FINIDA B, E NA SUBROTINA PRODUT OBTEMOS A MATRIZ
C         SIMETRICA A=B*TRANSPOSTA(C), O QUE POSSIBILITA
C         REDUZIR O PROBLEMA SIMETRICO GENERALIZADO DE AU-
C         TOVALORES BX=([illegible])AX, PARA O PROBLEMA SIMETRI-
C         CO PADRAO [illegible]=([illegible])Z, ATRAVES DA SUBROTINA REDUZ.
C         [illegible] ESTAGIOS DO ME-
C         TODO DE HOUSEHOLDER [illegible] A MATRIZ SIMETRICA [illegible]
C         [illegible] MATRIZ SIMETRICA [illegible] NA SUBROTINA
C         HOUSE.
C         [illegible], COMPUTAMOS OS AUTO-
C         VALORES [illegible] MATRIZ C, ATRAVES
C         [illegible] RACIONAL DO METODO QL.

          [illegible]
          [illegible]
          READ([illegible])
1[illegible]      FORMAT([illegible])
          READ([illegible])
2[illegible]      FORMAT([illegible])
          WRITE(3,[illegible])
3[illegible]      FORMAT(/3X,'[illegible]')
          WRITE(3,[illegible])
4[illegible]      FORMAT(//3X,'[illegible] COMPANHEIRA C=',I)
          WRITE(3,5[illegible])
5[illegible]      FORMAT(///5X,'[illegible] MATRIZ C',//)
          WRITE(3,2[illegible])(C(I),I=1,[illegible])
          CALL HANKEL([illegible])
          CALL PRODUT([illegible])
          CALL REDUZ([illegible])
          IF(IND.EQ.0) GO TO 7[illegible]
          WRITE(3,6[illegible])
6[illegible]      FORMAT(///3X,'A MATRIZ B NAO E POSITIVA DEFINIDA',//)
          GO TO 11[illegible]
7[illegible]      CALL HOUSE([illegible])
          CALL QL[illegible]([illegible])
          IF(IND.EQ.0) GO TO 9000
          WRITE(3,8000)IND
8000      FORMAT(///3X,'AUTOVALOR QUE NAO CONVERGE=',I,//)
          GO TO 11000
9000      WRITE(3,10000)
10000     FORMAT(///5X,'AUTOVALORES DA MATRIZ C',//)
          WRITE(3,2000)(D(I),I=1,[illegible])
11000     STOP
          END
```

```
C        4.7. BALANC:

C        SUBROTINA BALANC
C        ESSA SUBROTINA BALANCEIA A MATRIZ REAL GERAL A,
C        ISTO E, IGUALA AS SOMAS DOS MODULOS DOS ELEMEN-
C        TOS DAS LINHAS E COLUNAS CORRESPONDENTES. TAM-
C        BEM ISOLA AUTOVALORES, SEMPRE QUE POSSIVEL.
C        N E UMA VARIAVEL DE ENTRADA.
C        A E NA ENTRADA, A MATRIZ DE ORDEM N QUE SERA
C        BALANCEADA. NA SAIDA, CONTEM A MATRIZ TRANSFORMADA.
C        LOW E IGH SAO VARIAVEIS INTEIRAS DE SAIDA QUE IN-
C        DICAM OS INDICES DE LIMITE DA MATRIZ BALANCEADA.
C        SCALE E UMA VARIAVEL REAL DE SAIDA, QUE CONTEM IN-
C        FORMACOES SOBRE AS TRANSFORMACOES DE SIMILARIDADE.
C        RADIX E UM PARAMETRO DEPENDENTE DO COMPUTADOR ES-
C        PECIFICANDO A BASE DA REPRESENTACAO DE PONTO FLU-
C        TUANTE. PARA O NOSSO CASO, RADIX= 2.

         SUBROUTINE BALANC(N)
         COMMON /MATRIZ/A([illegible],[illegible]),LOW,IGH
         DIMENSION SCALE([illegible])
         LOGICAL NOCONV
         RADIX=2.
         B2=RADIX*RADIX
         K=1
         L=N
         GO TO 100
C        PROCEDIMENTO EMBUTIDO PARA TROCA DE LINHA E COLUNA
20       SCALE(M)=J
         IF(J.EQ.M) GO TO 50
         DO 30 I=1,L
         F=A(I,J)
         A(I,J)=A(I,M)
         A(I,M)=F
30       CONTINUE
         DO 40 I=K,N
         F=A(J,I)
         A(J,I)=A(M,I)
         A(M,I)=F
40       CONTINUE
50       GO TO (80,130), IEXC
C        PESQUISA DE LINHAS: ISOLA UM AUTOVALOR E MUDA PARA
C        A LINHA SEGUINTE, PARA BAIXO.
80       IF(L.EQ.1) GO TO 280
         L=L-1
100      DO 120 JJ=1,L
         J=L+1-JJ
         DO 110 I=1,L
         IF(I.EQ.J) GO TO 110
         IF(A(J,I).NE.0.) GO TO 120
110      CONTINUE
```

```
      M=L
      IEXC=1
      GO TO 20
120   CONTINUE
      GO TO 130
C     PESQUISA DE COLUNAS: ISOLA UM AUTOVALOR E MUDA PARA
C     A COLUNA SEGUINTE, A ESQUERDA.
130   K=K+1
140   DO 170 J=K,L
      DO 150 I=K,L
      IF(I.EQ.J) GO TO 150
      IF(A(I,J).NE.0.) GO TO 170
150   CONTINUE
      M=K
      IEXC=2
      GO TO 20
170   CONTINUE
C     [illegible] SUBMATRIZ FORMADA PELAS LINHAS K A L
      DO 180 I=K,L
180   SCALE(I)=1.
C     [illegible] REDUCAO DA NORMA
190   NOCONV=.FALSE.
      DO 270 I=K,L
      C=0.
      R=0.
      DO 200 J=K,L
      IF(J.EQ.I) GO TO 200
      C=C+ABS(A(J,I))
      R=R+ABS(A(I,J))
200   CONTINUE
C     [illegible] C OU R IGUAIS A ZERO.
      IF(C.EQ.0..OR.R.EQ.0.) GO TO 270
      G=R/RADIX
      F=1.
      S=C+R
210   IF(C.GE.G) GO TO 220
      F=F*RADIX
      C=C*B2
      GO TO 210
220   G=R*RADIX
230   IF(C.LT.G) GO TO 240
      F=F/RADIX
      C=C/B2
      GO TO 230
C     BALANCEAMENTO
240   IF((C+R)/F.GE.0.95*S) GO TO 270
      G=1./F
      SCALE(I)=SCALE(I)*F
      NOCONV=.TRUE.
      DO 250 J=K,N
250   A(I,J)=A(I,J)*G
      DO 260 J=1,L
```

```
2[illegible]0      [illegible](J,I)=[illegible](J,I)*[illegible]
270      CONTINUE
         IF([illegible]) GO TO 190
2[illegible]0      [illegible]=[illegible]
         ISN=[illegible]
         RETURN
         END
```

```
C        4.8.  HQR:

C        SUBROTINA HQR2
C        ESSA SUBROTINA DETERMINA OS AUTOVALORES DE UMA
C        MATRIZ DE HESSENBERG SUPERIOR REAL H, USANDO O
C        METODO QR.
C        N E UMA VARIAVEL INTEIRA DE ENTRADA.
C        LOW E IGH SAO VARIAVEIS INTEIRAS DE ENTRADA QUE
C        INDICAM OS INDICES DO LIMITE DA MATRIZ BALANCEA-
C        DA. SE A MATRIZ NAO E BALANCEADA, LOW=1 E IGH=N.
C        H E DE ENTRADA A MATRIZ DE HESSENBERG SUPERIOR
C        DE ORDEM N. O ALGORITMO HQR DESTROI A MATRIZ H.
C        WR E DE SAIDA UM VETOR DE DIMENSAO N, CONTENDO
C        OS AUTOVALORES REAIS DA MATRIZ DE HESSENBERG
C        SUPERIOR. OS AUTOVALORES NAO SAO ORDENADOS.
C        IERR E UMA VARIAVEL INTEIRA DE SAIDA. SE MAIS
C        DE 30 ITERACOES SAO REQUERIDAS PARA A DETERMINA-
C        CAO DE UM AUTOVALOR, HQR TERMINA COM IERR QUE
C        INDICA O INDICE DO AUTOVALOR PARA O QUAL A FALHA
C        OCORREU. [illegible]
C        [illegible], IERR E IGUAL A ZERO.
C        MACHEP E UM PARAMETRO DEPENDENTE DO COMPUTADOR,
C        QUE ESPECIFICA A PRECISAO RELATIVA DO PONTO FLU-
C        TUANTE ARITMETICO, OU SEJA, MACHEP E O MENOR
C        NUMERO, TAL QUE 1+EPSILON>1. PARA O PDP-10
C        O VALOR DE MACHEP E IGUAL A 2.**-26.

C        FUNCAO QUE INDICA O MENOR ENTRE I E J
         FUNCTION MIN0(I,J)
         MIN0=I
         IF(J.LT.I) MIN0=J
         RETURN
         END
         SUBROUTINE HQR(N)
         COMMON /BL1/H(100,100),LOW,IGH
         COMMON /BL2/WR(100),IERR,ICOM
         REAL MACHEP,NORM
         INTEGER EN,ENM2
         LOGICAL NOTLAS
         MACHEP=2.**-26
         IERR=0
         NORM=0.
         K=1
C        GUARDA AS RAIZES ISOLADAS PELO BALANCEAMENTO
C        E COMPUTA A NORMA DA MATRIZ
         DO 50 I=1,N
         DO 40 J=K,N
40       NORM=NORM+ABS(H(I,J))
         K=I
         IF(I.GE.LOW.AND.I.LE.IGH) GO TO 50
         WR(I)=H(I,I)
```

```
50        CONTINUE
          EN=IGH
          T=0.
C         PESQUISA DOS PROXIMOS AUTOVALORES
60        IF(EN.LT.LOW) GO TO 1001
          ITS=0
          NA=EN-1
          ENM2=NA-1
C         PROCURA UM ELEMENTO PEQUENO DA SUBDIAGONAL.
70        DO 80 LL=LOW,EN
          L=EN+LOW-LL
          IF(L.EQ.LOW) GO TO 100
          S=ABS(H(L-1,L-1))+ABS(H(L,L))
          IF(S.EQ.0.) S=NORM
          IF(ABS(H(L,L-1)).LE.MINEPS*S) GO TO 100
80        CONTINUE
C         FORMACAO DA PERTURBACAO X
100       X=H(EN,EN)
          IF(L.EQ.EN) GO TO 270
          Y=H(NA,NA)
          W=H(EN,NA)*H(NA,EN)
          IF(L.EQ.NA) GO TO 280
          IF(ITS.EQ.30) GO TO 1000
          IF(ITS.NE.10.AND.ITS.NE.20) GO TO 130
C         FORMACAO DA PERTURBACAO EXCEPCIONAL
          T=T+X
          DO 120 I=LOW,EN
120       H(I,I)=H(I,I)-X
          S=ABS(H(EN,NA))+ABS(H(NA,ENM2))
          X=0.75*S
          Y=X
          W=-0.4375*S*S
130       ITS=ITS+1
C         PROCURA DOS DOIS MENORES ELEMENTOS CONSECU-
C         TIVOS DA SUBDIAGONAL
          DO 140 MM=L,ENM2
          M=ENM2+L-MM
          ZZ=H(M,M)
          R=X-ZZ
          S=Y-ZZ
          P=(R*S-W)/H(M+1,M)+H(M,M+1)
          Q=H(M+1,M+1)-ZZ-R-S
          R=H(M+2,M+1)
          S=ABS(P)+ABS(Q)+ABS(R)
          P=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)*
        X(ABS(H(M-1,M-1))+ABS(ZZ)+ABS(H(M+1,M+1))))GO TO 150
140       CONTINUE
150       MP2=M+2
          DO 160 I=MP2,EN
```

```
      H(I,I-2)=0.
      IF(I.EQ.MP2) GO TO 160
      H(I,I-3)=0.
160   CONTINUE
C     PASSO QR DUPLO ENVOLVENDO AS LINHAS L A
C     EN E COLUNAS M A EN
      DO 260 K=M,NA
      NOTLAS=K.NE.NA
      IF(K.EQ.M) GO TO 170
      P=H(K,K-1)
      Q=H(K+1,K-1)
      R=0.
      IF(NOTLAS) R=H(K+2,K-1)
      X=ABS(P)+ABS(Q)+ABS(R)
      IF(X.EQ.0.) GO TO 260
      P=P/X
      Q=Q/X
      R=R/X
170   S=SIGN(SQRT(P*P+Q*Q+R*R),P)
      IF(K.EQ.M) GO TO 180
      H(K,K-1)=-S*X
      GO TO 190
180   IF(L.NE.M) H(K,K-1)=-H(K,K-1)
190   P=P+S
      X=P/S
      Y=Q/S
      ZZ=R/S
      Q=Q/P
      R=R/P
C     MODIFICACAO DA LINHA
      DO 210 J=K,EN
      P=H(K,J)+Q*H(K+1,J)
      IF(.NOT.NOTLAS) GO TO 200
      P=P+R*H(K+2,J)
      H(K+2,J)=H(K+2,J)-P*ZZ
200   H(K+1,J)=H(K+1,J)-P*Y
      H(K,J)=H(K,J)-P*X
210   CONTINUE
      J=MIN0(EN,K+3)
C     MODIFICACAO DA COLUNA
      DO 230 I=L,J
      P=X*H(I,K)+Y*H(I,K+1)
      IF(.NOT.NOTLAS) GO TO 220
      P=P+ZZ*H(I,K+2)
      H(I,K+2)=H(I,K+2)-P*R
220   H(I,K+1)=H(I,K+1)-P*Q
      H(I,K)=H(I,K)-P
230   CONTINUE
260   CONTINUE
      GO TO 70
C     UMA RAIZ E ACHADA
270   WR(EN)=X+T
```

```
      EN=NA
      GO TO 60
C     DUAS RAIZES SAO ACHADAS
280   P=(Y-X)/2.
      Q=P*P+W
      ZZ=SQRT(ABS(Q))
      X=X+T
      IF(Q.LT.0.) GO TO 320
C     PAR REAL
      ZZ=P+SIGN(ZZ,P)
      WR(NA)=X+ZZ
      WR(EN)=WR(NA)
      IF(ZZ.NE.0.) WR(EN)=X-W/ZZ
      GO TO 330
320   ICH=1
330   EN=[illegible]
      GO TO 60
C     [illegible] NAO HA CONVERGENCIA
C     PARA [illegible] AUTOVALOR
1[illegible] [illegible]
1[illegible] RETURN
      [illegible]
```

```
C         4.4. QR:

C         II METODO QR
C         ESSE METODO COMPUTA OS AUTOVALORES REAIS DA TRANS-
C         POSTA DA MATRIZ COMPANHEIRA REAL C.
C         PRIMEIRAMENTE, E FEITA À MONTAGEM DA TRANSPOSTA
C         DA MATRIZ C, , ONDE A ULTIMA LINHA E FORNECIDA
C         COMO DADO DE ENTRADA.
C         NA SUBROTINA BALANC, A MATRIZ E BALANCEADA, DE MODO
C         QUE AS NORMAS DAS LINHAS E COLUNAS CORRESPONDENTES
C         SAO IGUALADAS, PODE OCORRER DE ALGUM AUTOVALOR SER
C         ISOLADO.
C         NA SUBROTINA HQR, OS AUTOVALORES REAIS SAO FINAL-
C         MENTE COMPUTADOS.

          COMMON /BL1/A(100,100),LOW,IGH
          COMMON /BL2/WR(100),ITER,ICOM
          DIMENSION C(100)
          WRITE(3,50)
 50       FORMAT(////5X,'PELO METODO QR')
          READ(20,100)N
 100      FORMAT(I)
          READ(20,300)(C(I),I=1,N)
 300      FORMAT(100G)
          WRITE(3,400)N
 400      FORMAT(//3X,'ORDEM DA MATRIZ COMPANHEIRA C=',I)
          WRITE(3,500)
 500      FORMAT(///5X,'ULTIMA LINHA DA MATRIZ COMPANHEIRA C',//)
          WRITE(3,300)(C(I),I=1,N)
C         CONSTRUCAO DA TRANSPOSTA DA MATRIZ C
          DO 650 I=1,N
          DO 630 J=1,N-1
          IF(J.EQ.I-1) GO TO 600
          A(I,J)=0.
          GO TO 630
 600      A(I,J)=1.
 630      CONTINUE
 650      A(I,N)=C(I)
          CALL BALANC(N)
          CALL HQR(N)
          IF(ITER.NE.0) GO TO 1300
          WRITE(3,1100)
 1100     FORMAT(///5X,'AUTOVALORES REAIS',//)
          WRITE(3,300)(WR(I),I=1,N)
          IF(ICOM.NE.0) WRITE(3,1200)
          GO TO 1500
 1200     FORMAT(//3X,'EXISTE AUTOVALOR COMPLEXO')
          GO TO 1500
 1300     WRITE(3,1400)ITER
 1400     FORMAT(///5X,'AUTOVALOR QUE NAO CONVERGE=',I,//)
 1500     STOP
          END
```

REFERÊNCIAS:

[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 Algebric 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.