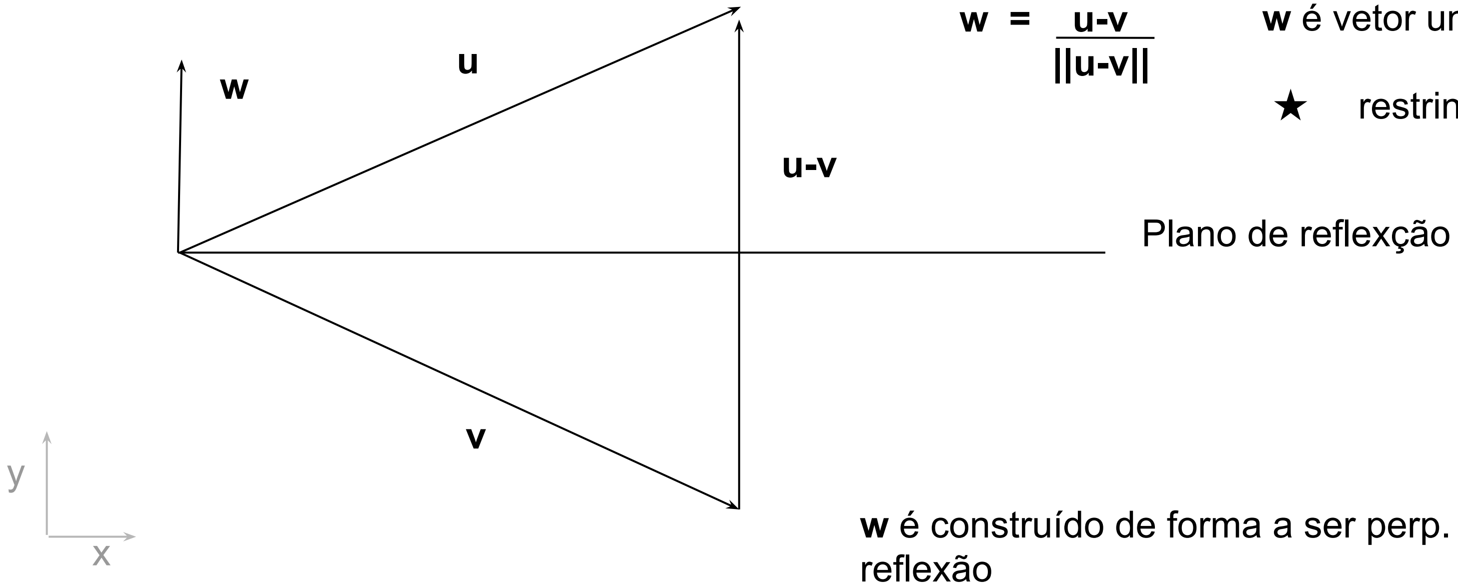
Decomposição QR

Bruno Nunes Cabral Tenório
Jorge Junio Moreira Antunes
Algebra Linear Computacional
Mauro Rincon
PPGI - UFRJ

Método de Holseholder & matriz de reflexão

Seja **u** e **v** $\in R^k$



$$\mathbf{w} = \frac{\mathbf{u-v}}{||\mathbf{u-v}||}$$

w é vetor unitário

★ restringimos $||\mathbf{v}||=||\mathbf{u}||$

w é construído de forma a ser perp. ao plano de reflexão

Matriz de reflexão de Householder

Matriz de reflexão:

definimos:

$H_k = I_k - 2\mathbf{w}\mathbf{w}^t$

onde

H_k é simétrica e ortogonal ($\mathbf{w}^t\mathbf{w} = 1$ porque $\mathbf{w}$ é unitário)

$H_k H_k = I_k$

Matriz de reflexão de Householder

Considere que H_k atua em dois vetores $\mathbf{x}$ e $\mathbf{y} \in R^k$ tais que

i) $\mathbf{x}$ perpendicular a $\mathbf{w}$

ii) $\mathbf{y}$ paralelo a $\mathbf{w}$

i) $H_k\mathbf{x}=(I_k - 2\mathbf{w}\mathbf{w}^\mathbf{t})\mathbf{x} = \mathbf{x} - 2\cancel{\mathbf{w}\mathbf{w}^\mathbf{t}\mathbf{x}}= \mathbf{x}$

mas se $\mathbf{y} \backslash\backslash \mathbf{w}$: $\mathbf{y}=c.\mathbf{w}$

ii) $H_k\mathbf{y}=(I_k - 2\mathbf{w}\mathbf{w}^\mathbf{t})\mathbf{y} = \mathbf{y} - 2\mathbf{w}\mathbf{w}^\mathbf{t}\mathbf{y} = \mathbf{y} - 2c.\mathbf{w}\mathbf{w}^\mathbf{t}\mathbf{w} = \mathbf{y} - 2c.\mathbf{w} = \mathbf{y}-2\mathbf{y}=-\mathbf{y}$

i) e ii) nos diz como acontece a reflexão!

Matriz de reflexão de Householder

Seja $\mathbf{z}$ um vetor de R^k, podemos escrever $\mathbf{z} = \mathbf{z}_{perp|\mathbf{w}} + \mathbf{z}_{//\mathbf{w}}$

$H_k\mathbf{z} = H_k(\mathbf{z}_{perp|\mathbf{w}} + \mathbf{z}_{//\mathbf{w}}) = \mathbf{z}_{perp|\mathbf{w}} - \mathbf{z}_{//\mathbf{w}}$

comparando com os vetores **u** e **v** da figura,

$H_k\mathbf{u} = \mathbf{v}$

$H_k\mathbf{v} = \mathbf{u}$

Como usamos isso para reduzir uma matriz simétrica A para a forma Tridiagonal??

Método de Holseholder

Consideramos a matriz simétrica A

$$A = \begin{bmatrix} A_{11} & A_{12} & A_{13} & \dots & A_{1n} \\ A_{21} & A_{22} & A_{23} & \dots & A_{2n} \\ A_{31} & A_{32} & A_{33} & \dots & A_{3n} \\ A_{41} & A_{42} & A_{43} & \dots & A_{4n} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ A_{n1} & A_{n2} & A_{n3} & \dots & A_{nn} \end{bmatrix}_{n \times n}$$

$\mathbf{u} \in R^{n-1}$

Quero projetar esse vetor **u** tal que **v** // e_{n-1}

$\mathbf{v} = (||\mathbf{u}||,0,0,\dots)$

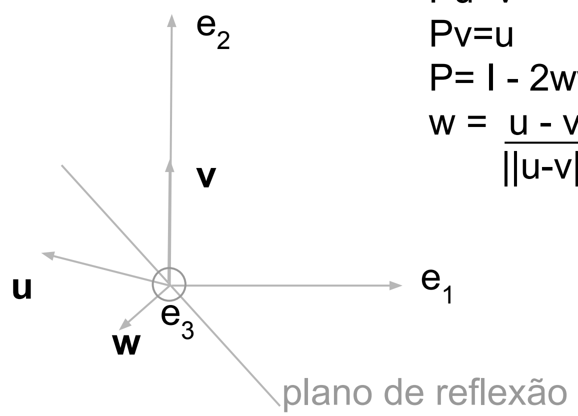
Exemplo no R^3

$Pu=v$

$Pv=u$

$P= I - 2ww^t$

$w = \dfrac{u - v}{||u-v||}$



Método de Holseholder

Preciso encontrar a matriz de reflexão P para aplicar em A^{n-1} e continuar o processo de tri-diagonalização.

$$A^1 = \begin{pmatrix} a_{22} & \cdots & a_{n2} \\ \vdots & \ddots & \vdots \\ a_{n2} & \cdots & a_{nn} \end{pmatrix}$$

$$PAP = \begin{pmatrix} a_{11} & ||u|| & 0 \\ ||u|| & & P^1 A^1 P^1 \\ 0 & & \end{pmatrix}$$

$$P = \begin{pmatrix} 1 & 0 \\ 0 & P^1 \end{pmatrix}$$

$$P^1 = I - 2ww^t$$

$$w = \frac{u - v}{||u - v||}$$

Método de Holseholder

Exemplo:

$$A = \begin{bmatrix} 4 & 1 & -2 & 2 \\ 1 & 2 & 0 & 1 \\ -2 & 0 & 3 & -2 \\ 2 & 1 & -2 & -1 \end{bmatrix}; \ u = (1,-2,2)^t, \ \left||u|\right| = 3, \qquad v = (\left||u|\right|,0,0)^t = (3,0,0)^t$$

$$w = \frac{u-v}{||u-v||} = \frac{(-2,-2,2)^t}{\sqrt{12}}$$

$$ww^t = \frac{1}{12}\begin{bmatrix} -2 \\ -2 \\ 2 \end{bmatrix}\begin{bmatrix} -2 & -2 & 2 \end{bmatrix} = \frac{1}{12}\begin{bmatrix} 4 & 4 & -4 \\ 4 & 4 & -4 \\ -4 & -4 & 4 \end{bmatrix} = \frac{1}{3}\begin{bmatrix} 1 & 1 & -1 \\ 1 & 1 & -1 \\ -1 & -1 & 1 \end{bmatrix}$$

$$P^1 = I - 2ww^t = \begin{bmatrix} \frac{1}{3} & -\frac{2}{3} & \frac{2}{3} \\ -\frac{2}{3} & \frac{1}{3} & \frac{2}{3} \\ \frac{2}{3} & \frac{2}{3} & \frac{1}{3} \end{bmatrix}, \quad onde\ verificamos, \quad P^1u = \begin{bmatrix} \frac{9}{3} \\ 0 \\ 0 \end{bmatrix} = v\,,\ P^1v = \begin{bmatrix} 1 \\ -2 \\ 2 \end{bmatrix} = u$$

File Edit Insert Format Cell Graphics Evaluation Palettes Window Help

Wolfram Mathematica | STUDENT EDITION

```
In[1]:= ph = {{1, 0, 0, 0}, {0, 1 / 3, -2 / 3, 2 / 3}, {0, -2 / 3, 1 / 3, 2 / 3}, {0, 2 / 3, 2 / 3, 1 / 3}};
        a = {{4, 1, -2, 2}, {1, 2, 0, 1}, {-2, 0, 3, -2}, {2, 1, -2, -1}};

In[5]:= Print["P=" MatrixForm[ph]]
        Print["A=" MatrixForm[a]]
```

$$P = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & \frac{1}{3} & -\frac{2}{3} & \frac{2}{3} \\ 0 & -\frac{2}{3} & \frac{1}{3} & \frac{2}{3} \\ 0 & \frac{2}{3} & \frac{2}{3} & \frac{1}{3} \end{pmatrix}$$

$$A = \begin{pmatrix} 4 & 1 & -2 & 2 \\ 1 & 2 & 0 & 1 \\ -2 & 0 & 3 & -2 \\ 2 & 1 & -2 & -1 \end{pmatrix}$$

```
In[7]:= Print["PAP=" MatrixForm[ph.a.ph]]
```

$$PAP = \begin{pmatrix} 4 & 3 & 0 & 0 \\ 3 & \frac{10}{3} & -\frac{4}{3} & -1 \\ 0 & -\frac{4}{3} & -1 & -\frac{4}{3} \\ 0 & -1 & -\frac{4}{3} & \frac{5}{3} \end{pmatrix}$$

```
In[9]:= Eigenvalues[a] // N

Out[9]= {6.84462, 2.26853, -2.19752, 1.08436}
```

total sort mean max more...

Decomposição QR

Ideia:

- Reduzir a matriz para obter todos os autovalores
- Matriz A tridiagonal da seguinte forma:

$$A = \begin{bmatrix} a_1 & b_2 & 0 & \cdots & 0 \\ b_2 & a_2 & b_3 & \ddots & \vdots \\ 0 & b_3 & a_3 & \ddots & 0 \\ \vdots & \ddots & \ddots & \ddots & b_n \\ 0 & \cdots & 0 & b_n & a_n \end{bmatrix}$$

- Se b2 ou bn forem 0 -> a1 ou an é o autovalor

Decomposição QR

- Existirá sequencia de matrizes equivalentes: $A = A^{(1)}, A^{(2)}, A^{(3)}, \ldots$
- Matriz A pode ser escrita como: $A^{(1)} = Q^{(1)}R^{(1)}$

$$A^{(i+1)} = R^{(i)}Q^{(i)}$$

- Como Q é uma matriz ortogonal: $R^{(i)} = Q^{(i)^t}A^{(i)}$

$$A^{(i+1)} = R^{(i)}Q^{(i)} = (Q^{(i)^t}A^{(i)})Q^{(i)} = Q^{(i)^t}A^{(i)}Q^{(i)}$$

- Preserva os autovalores, simetria e tridiagonalidade.

Decomposição QR

- A fatoração QR utiliza n-1 Matrizes de rotação P da seguinte forma:

$$R^{(1)} = P_n P_{n-1} \cdots P_2 A^{(1)}$$

- A matriz de rotação:

$$P_{k+1} = \left[\begin{array}{c|cc|c} I_{k-1} & & O & O \\ \hline & c_{k+1} & s_{k+1} & \\ O & & & O \\ & -s_{k+1} & c_{k+1} & \\ \hline O & & O & I_{n-k-1} \end{array}\right]$$

$$s_{k+1} = \frac{b_{k+1}}{\sqrt{b_{k+1}^2 + x_k^2}}$$

$$c_{k+1} = \frac{x_k}{\sqrt{b_{k+1}^2 + x_k^2}}$$

Decomposição QR

$$A_{k+1}^{(1)} = \begin{bmatrix} z_1 & q_1 & r_1 & 0 & \cdots & \cdots & \cdots & 0 \\ 0 & \ddots & \ddots & \ddots & \ddots & & & \vdots \\ 0 & \ddots & \ddots & \ddots & \ddots & \ddots & & \vdots \\ \vdots & \ddots & 0 & z_k & q_k & r_k & \ddots & \vdots \\ \vdots & & \ddots & 0 & x_{k+1} & y_{k+1} & 0 & 0 \\ \vdots & & & \ddots & b_{k+2} & a_{k+2} & b_{k+3} & 0 \\ \vdots & & & & \ddots & \ddots & \ddots & b_n \\ 0 & \cdots & \cdots & \cdots & \cdots & 0 & b_n & a_n \end{bmatrix}$$

- Q é a matriz ortogonal:

$$Q^{(1)} = P_2^t P_3^t \cdots P_n^t$$

- Com R e Q no primeiro passo, calcula-se A no passo seguinte.
- Repete o algoritmo até zerar b2 ou bn -> Acha-se o autovalor.

Decomposição QR

- A convergência do método pode ser lenta se $|\lambda_{j+1}/\lambda_j|$ é próximo de 1.
- A convergência é acelerada fazendo um “shift” da seguinte forma:

$$A^{(i)} - \sigma I = Q^{(i)} R^{(i)}$$
$$A^{(i+1)} = R^{(i)} Q^{(i)} + \sigma I$$

- O autovalor da matriz é modificado por σ.
- Valor próximo a um dos autovalores de A
- Convergencia dada por $|(\lambda_{j+1} - \sigma)/(\lambda_j - \sigma)|$

Decomposição QR

Step 1 Set $k = 1$;
$SHIFT = 0$. *(Accumulated shift.)*

Step 2 While $k \leq M$ do Steps 3–19.
(Steps 3–7 test for success.)

Step 3 If $|b_n^{(k)}| \leq TOL$ then set $\lambda = a_n^{(k)} + SHIFT$;
OUTPUT (λ);
set $n = n - 1$.

Step 4 If $|b_2^{(k)}| \leq TOL$ then set $\lambda = a_1^{(k)} + SHIFT$;
OUTPUT (λ);
set $n = n - 1$;
$a_1^{(k)} = a_2^{(k)}$;
for $j = 2, \ldots, n$
set $a_j^{(k)} = a_{j+1}^{(k)}$;
$b_j^{(k)} = b_{j+1}^{(k)}$.

Step 5 If $n = 0$ then
STOP.

Step 6 If $n = 1$ then
set $\lambda = a_1^{(k)} + SHIFT$;
OUTPUT (λ);
STOP.

Step 7 For $j = 3, \ldots, n - 1$
if $|b_j^{(k)}| \leq TOL$ then
OUTPUT ('split into', $a_1^{(k)}, \ldots, a_{j-1}^{(k)}, b_2^{(k)}, \ldots, b_{j-1}^{(k)}$,
'and',
$a_j^{(k)}, \ldots, a_n^{(k)}, b_{j+1}^{(k)}, \ldots, b_n^{(k)}, SHIFT$);
STOP.

Step 8 *(Compute shift.)*
Set $b = -(a_{n-1}^{(k)} + a_n^{(k)})$;
$c = a_n^{(k)} a_{n-1}^{(k)} - [b_n^{(k)}]^2$;
$d = (b^2 - 4c)^{1/2}$.

Step 9 If $b > 0$ then set $\mu_1 = -2c/(b + d)$;
$\mu_2 = -(b + d)/2$
else set $\mu_1 = (d - b)/2$;
$\mu_2 = 2c/(d - b)$.

Step 10 If $n = 2$ then set $\lambda_1 = \mu_1 + SHIFT$;
$\lambda_2 = \mu_2 + SHIFT$;
OUTPUT (λ_1, λ_2);
STOP.

Step 11 Choose σ so that $|\sigma - a_n^{(k)}| = \min\{|\mu_1 - a_n^{(k)}|, |\mu_2 - a_n^{(k)}|\}$.

Step 12 *(Accumulate the shift.)*
Set $SHIFT = SHIFT + \sigma$.

Step 13 *(Perform shift.)*
For $j = 1, \ldots, n$, set $d_j = a_j^{(k)} - \sigma$.

Decomposição QR

Step 14 (*Steps 14 and 15 compute* $R^{(k)}$.)

Set $x_1 = d_1$;

$y_1 = b_2$.

Step 15 For $j = 2, \ldots, n$

set $z_{j-1} = \left\{x_{j-1}^2 + \left[b_j^{(k)}\right]^2\right\}^{1/2}$;

$c_j = \dfrac{x_{j-1}}{z_{j-1}}$;

$\sigma_j = \dfrac{b_j^{(k)}}{z_{j-1}}$;

$q_{j-1} = c_j y_{j-1} + s_j d_j$;

$x_j = -\sigma_j y_{j-1} + c_j d_j$;

If $j \neq n$ then set $r_{j-1} = \sigma_j b_{j+1}^{(k)}$;

$y_j = c_j b_{j+1}^{(k)}$.

($A_j^{(k)} = P_j A_{j-1}^{(k)}$ *has just been computed and* $R^{(k)} = A_n^{(k)}$.)

Step 16 (*Steps 16–18 compute* $A^{(k+1)}$.)

Set $z_n = x_n$;

$a_1^{(k+1)} = \sigma_2 q_1 + c_2 z_1$;

$b_2^{(k+1)} = \sigma_2 z_2$.

Step 17 For $j = 2, 3, \ldots, n-1$

set $a_j^{(k+1)} = \sigma_{j+1} q_j + c_j c_{j+1} z_j$;

$b_{j+1}^{(k+1)} = \sigma_{j+1} z_{j+1}$.

Step 18 Set $a_n^{(k+1)} = c_n z_n$.

Step 19 Set $k = k + 1$.

Step 20 OUTPUT ('Maximum number of iterations exceeded');

(*The procedure was unsuccessful.*)

STOP.

Casos Testes

Casos:

1. Matriz nao upper-Hessenberg
2. Matriz upper-Hessenberg
3. Matriz tridiagonal

Tamanhos: n = 3 e n = 4

Casos Testes

Matrizes nao upper-Hessenberg

n = 3

$$A = \begin{bmatrix} 1 & 1 & 1 \\ 4 & 1 & 2 \\ 5 & 4 & 6 \end{bmatrix}$$

n = 4

$$A = \begin{bmatrix} 4 & 1 & -2 & 2 \\ 1 & 2 & 0 & 1 \\ -2 & 0 & 3 & -2 \\ 2 & 1 & -2 & -1 \end{bmatrix}$$

Casos Testes

n = 3

```
>>> runfile(r'C:\Users\jorgeantunes\Downloads\QR2 (1).py', wdir=r'C:\Users\jorgeantunes\Downloads')
dimensao da matriz: 3
linha 1 da matriz: 1,1,1
linha 2 da matriz: 4,1,2
linha 3 da matriz: 5,4,6
a matriz eh upper-Hessemberg (ou tridiagonal)?: sim ou nao nao
matriz upper-Hessemberg (ou tridiagonal)
[[ 1.          1.40556386  0.15617376]
 [ 6.40312424  6.97560976 -0.7804878 ]
 [ 0.         -2.7804878   0.02439024]]
Fim do metodo QR
autovalor 8.40513274623
autovalor 0.594865737978
autovalor -0.999998484208
QR steps: 6
```

Input:

eigenvalues $\begin{pmatrix} 1 & 1 & 1 \\ 4 & 1 & 2 \\ 5 & 4 & 6 \end{pmatrix}$

Results:

$\lambda_1 \approx 8.40512$

$\lambda_2 = -1$

$\lambda_3 \approx 0.594875$

Casos Testes

n = 4

```
>>> runfile(r'C:\Users\jorgeantunes\Downloads\QR2 (1).py', wdir=r'C:\Users\jorgeantunes\Downloads')
dimensao da matriz: 4
linha 1 da matriz: 4,1,-2,2
linha 2 da matriz: 1,2,0,1
linha 3 da matriz: -2,0,3,-2
linha 4 da matriz: 2,1,-2,-1
a matriz eh upper-Hessemberg (ou tridiagonal)?: sim ou nao nao
matriz upper-Hessemberg (ou tridiagonal)
[[ 4.          3.          0.          0.        ]
 [ 3.          3.33333333  1.66666667  0.        ]
 [ 0.          1.66666667 -1.32       -0.90666667]
 [ 0.          0.         -0.90666667  1.98666667]]
Fim do metodo QR
autovalor 6.84462110507
autovalor 2.26853141207
autovalor 1.08436446394
autovalor -2.19751698109
QR steps: 10
```

Input:

eigenvalues $\begin{pmatrix} 4 & 1 & -2 & 2 \\ 1 & 2 & 0 & 1 \\ -2 & 0 & 3 & -2 \\ 2 & 1 & -2 & -1 \end{pmatrix}$

Results:

$\lambda_1 \approx 6.84462$

$\lambda_2 \approx 2.26853$

$\lambda_3 \approx -2.19752$

$\lambda_4 \approx 1.08436$

Casos Testes

Matrizes upper-Hessenberg

n = 3

$$A = \begin{bmatrix} 1 & 1 & 1 \\ 4 & 1 & 2 \\ 0 & 4 & 6 \end{bmatrix}$$

n = 4

$$A = \begin{bmatrix} 4 & 1 & 2 & 2 \\ 1 & 2 & 4 & 1 \\ 0 & 5 & 3 & 2 \\ 0 & 0 & 2 & 1 \end{bmatrix}$$

Casos Testes

n = 3

```
>>> runfile(r'C:\Users\jorgeantunes\Downloads\QR2 (1).py', wdir=r'C:\Users\jorgeantunes\Downloads')
dimensao da matriz: 3
linha 1 da matriz: 1,1,1
linha 2 da matriz: 4,1,2
linha 3 da matriz: 0,4,6
a matriz eh upper-Hessemberg (ou tridiagonal)?: sim ou nao sim
matriz upper-Hessemberg (ou tridiagonal)
[[ 1.  1.  1.]
 [ 4.  1.  2.]
 [ 0.  4.  6.]]
Fim do metodo QR
autovalor 7.70156211872
autovalor 1.29843788128
autovalor -1.0
QR steps: 2
```

Input:

eigenvalues $\begin{pmatrix} 1 & 1 & 1 \\ 4 & 1 & 2 \\ 0 & 4 & 6 \end{pmatrix}$

Results:

$\lambda_1 \approx 7.70156$

$\lambda_2 \approx 1.29844$

$\lambda_3 = -1$

Casos Testes

n = 4

```
>>> runfile(r'C:\Users\jorgeantunes\Downloads\QR2 (1).py', wdir=r'C:\Users\jorgeantunes\Downloads')
dimensao da matriz: 4
linha 1 da matriz: 4,1,2,2
linha 2 da matriz: 1,2,4,1
linha 3 da matriz: 0,5,3,2
linha 4 da matriz: 0,0,2,1
a matriz eh upper-Hessemberg (ou tridiagonal)?: sim ou nao sim
matriz upper-Hessemberg (ou tridiagonal)
[[ 4.  1.  2.  2.]
 [ 1.  2.  4.  1.]
 [ 0.  5.  3.  2.]
 [ 0.  0.  2.  1.]]
Fim do metodo QR
autovalor 7.93141307866
autovalor 3.34211799648
autovalor -2.23545348233
autovalor 0.961922407191
QR steps: 15
```

Input:

eigenvalues $\begin{pmatrix} 4 & 1 & 2 & 2 \\ 1 & 2 & 4 & 1 \\ 0 & 5 & 3 & 2 \\ 0 & 0 & 2 & 1 \end{pmatrix}$

Results:

$\lambda_1 \approx 7.93146$

$\lambda_2 \approx 3.34208$

$\lambda_3 \approx -2.23545$

$\lambda_4 \approx 0.961922$

Casos Testes

Matrizes tridiagonais

n = 3

$$A = \begin{bmatrix} 1 & 1 & 0 \\ 4 & 1 & 2 \\ 0 & 4 & 6 \end{bmatrix}$$

n = 4

$$A = \begin{bmatrix} 4 & 1 & 0 & 0 \\ 1 & 2 & 4 & 0 \\ 0 & 5 & 3 & 2 \\ 0 & 0 & 2 & 1 \end{bmatrix}$$

Casos Testes

n = 3

```
>>> runfile(r'C:\Users\jorgeantunes\Downloads\QR2 (1).py', wdir=r'C:\Users\jorgeantunes\Downloads')
dimensao da matriz: 3
linha 1 da matriz: 1,1,0
linha 2 da matriz: 4,1,2
linha 3 da matriz: 0,4,6
a matriz eh upper-Hessemberg (ou tridiagonal)?: sim ou nao sim
matriz upper-Hessemberg (ou tridiagonal)
[[ 1.  1.  0.]
 [ 4.  1.  2.]
 [ 0.  4.  6.]]
Fim do metodo QR
autovalor 7.38838459207
autovalor 2.20647135218
autovalor -1.59485594425
QR steps: 9
```

Input:

eigenvalues	$\begin{pmatrix} 1 & 1 & 0 \\ 4 & 1 & 2 \\ 0 & 4 & 6 \end{pmatrix}$

Results:

$\lambda_1 \approx 7.38835$

$\lambda_2 \approx 2.2065$

$\lambda_3 \approx -1.59486$

Casos Testes

n = 4

```
>>> runfile(r'C:\Users\jorgeantunes\Downloads\QR2 (1).py', wdir=r'C:\Users\jorgeantunes\Downloads')
dimensao da matriz: 4
linha 1 da matriz: 4,1,0,0
linha 2 da matriz: 1,2,4,0
linha 3 da matriz: 0,5,3,2
linha 4 da matriz: 0,0,2,1
a matriz eh upper-Hessemberg (ou tridiagonal)?: sim ou nao sim
matriz upper-Hessemberg (ou tridiagonal)
[[ 4.  1.  0.  0.]
 [ 1.  2.  4.  0.]
 [ 0.  5.  3.  2.]
 [ 0.  0.  2.  1.]]
Fim do metodo QR
autovalor 7.47415282963
autovalor 4.01509965552
autovalor 1.11374723375
autovalor -2.60299971891
QR steps: 20
```

Input:

eigenvalues $\begin{pmatrix} 4 & 1 & 0 & 0 \\ 1 & 2 & 4 & 0 \\ 0 & 5 & 3 & 2 \\ 0 & 0 & 2 & 1 \end{pmatrix}$

Results:

$\lambda_1 \approx 7.47415$

$\lambda_2 \approx 4.0151$

$\lambda_3 \approx -2.603$

$\lambda_4 \approx 1.11375$

Bibliografia

- Numerical Analysis, Richard L. Burden, J. Douglas Faires, 9.ed, Cengage Learning, 2010.
- Scientific Computing and Differential Equations: An Introduction to Numerical Methods, Gene H. Golub, M. Ortega, Academic Press, 1991.

Decomposição QR

Bruno Nunes Cabral Tenório
Jorge Junio Moreira Antunes
Algebra Linear Computacional
Mauro Rincon
PPGI - UFRJ