Curso: Métodos Numéricos II
Professor: Creto Augusto Vidal
Semestre: 2020.1
Aula # 22

1. Objetivo: Métodos que usam transformações de similaridade (continuação).

Nesta Aula, apresentamos mais um método que usa o conceito de transformação de similaridade: o Método **QR**.

2. Transformações de Similaridade (Recapitulação)

Dadas uma matriz **A** e uma matriz **P** que tenha uma inversa $\mathbf{P}^{-1}$, podemos construir uma nova matriz a partir do **produto** dessas três matrizes e chamá-la de $\bar{\mathbf{A}}$, isto é,

(1) $\bar{\mathbf{A}} = \mathbf{P}^{-1}\mathbf{A}\mathbf{P}$.

Se **P** for ortogonal, $\mathbf{P}^{-1} = \mathbf{P}^{\mathrm{T}}$, e a equação (1) fica mais fácil por não necessitar de inversão de matrizes. Assim,

(2) $\bar{\mathbf{A}} = \mathbf{P}^{\mathrm{T}}\mathbf{A}\mathbf{P}$.

Os autovalores da matriz $\bar{\mathbf{A}}$ são os mesmos da matriz original **A**, isto é, os espectros das duas matrizes são idênticos,

(3) $\lambda(\mathbf{A}) \equiv \lambda(\bar{\mathbf{A}})$.

Os autovetores da matriz original, **A**, podem ser obtidos a partir dos autovetores da matriz $\bar{\mathbf{A}}$ através da relação,

(4) $\mathbf{x}_i = \mathbf{P}\mathbf{v}_i$,

onde $\mathbf{v}_i$ é um autovetor da matriz $\bar{\mathbf{A}}$ e $\mathbf{x}_i$ é o autovetor correspondente da matriz, **A**, ou seja, esses autovetores compartilham o mesmo autovalor λ_i.

Se pusermos todos os autovetores da matriz $\bar{\mathbf{A}}$ como colunas de uma matriz **V** então

(5) $\mathbf{X} = \mathbf{P}\mathbf{V}$

onde **X** é a matriz cujas colunas são os autovetores correspondentes da matriz **A**.

Nota: O objetivo dos métodos que usam transformações de similaridade é fazer com que a matriz $\bar{\mathrm{A}}$ tenha uma **estrutura tão simples que seja fácil achar** seus **autovalores** λ_i e seus **autovetores** $\mathbf{v}_i$ para, então, através das relações (3) e (5) achar os autovalores e autovetores da matriz original **A.**

3. Método QR

O método de QR tem uma estrutura de implementação parecida com a do método de Jacobi. O objetivo do método QR é idêntico ao do método de Jacobi, isto é, obter uma matriz $\bar{\mathbf{A}}$ que seja **diagonal**. Se $\bar{\mathbf{A}}$ for diagonal, os autovalores de $\bar{\mathbf{A}}$ são os próprios elementos de sua diagonal e os autovetores $\mathbf{v}_i$ associados são as colunas da matriz identidade **I** (chamada base canônica). Então, pela equação (3), temos automaticamente os autovalores de **A**; e, pela equação (5), temos que os autovetores de **A** são

(6) $\mathbf{X} = \mathbf{P}\mathbf{I} = \mathbf{P}$.

Ou seja, os autovetores de **A** são as próprias colunas da matriz **P**.

O método QR faz uma sequência de transformações de similaridade até que a matriz transformada final tenha uma estrutura simples chamada matriz diagonal (isso acontece porque estamos lidando com matrizes **A simétricas**). Antes de discutirmos as propriedades da matriz de Q, $\mathbf{P} \equiv \mathbf{Q}$, vamos entender como essas matrizes vão ser aplicadas para chegar na matriz $\bar{\mathbf{A}}$ final, isto é,

$$\text{(7)} \qquad \bar{\mathbf{A}} = \underbrace{\mathbf{Q}_k^T \cdots \underbrace{\left(\mathbf{Q}_i^T \cdots \underbrace{\left(\mathbf{Q}_2^T \underbrace{(\mathbf{Q}_1^T \mathbf{A} \mathbf{Q}_1)}_{\textbf{Passo 1}} \mathbf{Q}_2 \right)}_{\textbf{Passo 2}} \cdots \mathbf{Q}_i \right)}_{\textbf{Passo i}} \cdots \mathbf{Q}_k}_{\textbf{Passo k}}.$$

A equação (7) dá uma visão macroscópica de como o algoritmo QR funciona. Serão necessárias várias iterações QR para termos uma matriz diagonal, ou muito próxima de uma matriz diagonal.

Vamos escrever o algoritmo que faz múltiplas iterações QR até que a matriz fique muito próxima de uma matriz diagonal.

3.1 Método QR

O algoritmo recebe a matriz para a qual se deseja achar os autovalores e os autovetores correspondentes, e uma tolerância ε que serve para verificar se os elementos fora da diagonal estão suficientemente próximos de zero. Em seguida, faz um loop aberto que só é finalizado quando a forma final de uma matriz diagonal for alcançada. Todas as matrizes envolvidas nas transformações de similaridade são acumuladas para poder recuperar os autovetores da matriz original.

3.1.1 Algoritmo QR

```
(Matriz, vetor) metodoQR(Matriz A, int n, float ε)
Matriz P, Q, R, A_nova, A_velha, Ā;
Vector Lamb;              // Vetor que armazena os autovalores de A.
float val=100.;           // Escalar ao qual é atribuída a soma dos quadrados dos elementos abaixo
                          // da diagonal da matriz A_nova para verificar convergência.
// Inicializar matrizes
P ← I;                    // Matriz que contém os produtos das matrizes ortogonais Q
                          // para recuperar os autovetores da matriz original.
A_velha ← A;
Enquanto (val > ε) faça // loop de diagonalização
        // Decomposição QR (devolve as matrizes Q e R tais que  A_velha = QR
        // onde Q é ortogonal e R é triangular superior.
        (Q, R) ← decomposicaoQR (A_velha, n);
        // Calcular a nova matriz como A_nova = RQ.  (na ordem reversa)
        A_nova ← RQ;   // isso é equivalente a A_nova ← Q^T A_velha Q (transformação de similaridade)
        // Salvar A_nova para a próxima iteração.
        A_velha ← A_nova;
        // Acumular o produto das matrizes Q como
        //  P ← I Q_1Q_2 ⋯ Q_k
        P ← P. Q;
        // Verificar se a matriz A_nova já é diagonal
        val = somaDosQuadradosDosTermosAbaixoDaDiagonal(A_nova, n)

End Enquanto
```

// Ao sair do loop, o formato da matriz $\mathbf{A}_{\text{nova}}$ já está suficientemente próximo do formato de
// uma matriz diagonal. Assim, os elementos da diagonal são os autovalores da matriz original
// de entrada e as colunas de **P** são os autovetores correspondente.
$\mathbf{Lamb}(1{:}\,n) \leftarrow (\mathbf{A}_{\text{nova}})_{i,i}(1{:}\,n)$; // Copia os elementos da diagonal da matriz no vetor Lamb
return (**P, Lamb**);

Neste momento, você consegue ver que o algoritmo QR é a reprodução exata da equação (7). Porém, a obtenção das matrizes **Q** e **R** precisa ser esclarecida.

3.1.2 Decomposição QR.

As matrizes **Q** e **R** são chamadas de matrizes da decomposição QR porque, dada uma matriz **A**, deseja-se expressar essa matriz como o produto

$$(8) \quad \mathbf{A} = \mathbf{QR},$$

onde **Q** é uma matriz ortogonal e **R** é uma matriz triangular superior (matriz cujos elementos abaixo da diagonal principal são todos nulos).

A multiplicação dos dois lados da igualdade em (8) pela matriz $\mathbf{Q}^{\mathrm{T}}$ resulta em

$$(9) \quad \begin{aligned} \mathbf{Q}^{\mathrm{T}}\mathbf{A} &= \mathbf{Q}^{\mathrm{T}}\mathbf{QR} \\ &= \mathbf{IR} \quad . \\ &= \mathbf{R} \end{aligned}$$

Assim, $\mathbf{Q}^{\mathrm{T}}$ é uma matriz ortogonal que transforma a matriz **A** em uma matriz triangular superior **R**.

Assumimos que **Q** é uma matriz ortogonal. Porém, será que a sua transposta também é ortogonal?

Partindo da expressão verdadeira $\mathbf{Q}^{\mathrm{T}}\mathbf{Q} = \mathbf{I}$

$$(10) \quad \begin{aligned} \left(\mathbf{Q}^{\mathrm{T}}\mathbf{Q}\right) &= (\mathbf{I}) && : \text{Ortogonalidade de } \mathbf{Q} \\ \mathbf{Q}\left(\mathbf{Q}^{\mathrm{T}}\mathbf{Q}\right) &= \mathbf{Q}(\mathbf{I}) && : \text{Multiplica por } \mathbf{Q} \\ \mathbf{Q}\mathbf{Q}^{\mathrm{T}}\mathbf{Q} &= \mathbf{I}\mathbf{Q} && : \text{Desagrupa o lado esquerdo e usa comutatividade com } \mathbf{I} \\ \mathbf{Q}\mathbf{Q}^{\mathrm{T}} &= \mathbf{I} && : \text{Comparação com a linha anterior} \\ \left(\mathbf{Q}^{\mathrm{T}}\right)^{\mathrm{T}}\left(\mathbf{Q}^{\mathrm{T}}\right) &= \mathbf{I}. && : \mathbf{Q} \text{ é a transposta de } \mathbf{Q}^{\mathrm{T}} \end{aligned}$$

Provamos, em (10), que, se **Q** é uma matriz ortogonal, então $\mathbf{Q}^{\mathrm{T}}$ também é uma matriz ortogonal. Sabemos também de aulas anteriores que o produto de matrizes ortogonais forma uma matriz ortogonal. Por isso,

vamos construir $\mathbf{Q}^{\mathrm{T}}$ como um produto de matrizes de Jacobi que, passo a passo transformam a matriz **A** na matriz **R**.

Como a matriz **R** deverá ter todos os seus elementos abaixo da diagonal principal iguais a zero, vamos fazer uma varredura nas $(n-1)$ primeiras colunas para zerar os elementos abaixo da diagonal principal, isto é

$$(11) \quad \begin{aligned} \mathbf{R} &= \mathbf{J}_{n(n-1)} \cdots \left(\mathbf{J}_{n1} \cdots \left(\mathbf{J}_{31}(\mathbf{J}_{21}\mathbf{A})\right)\right) \\ &= \mathbf{Q}^{\mathrm{T}}\mathbf{A} \end{aligned}$$

Assim, o algoritmo de decomposição QR é muito parecido com a varredura de Jacobi vista na Aula #21.

```
(Matriz, Matriz) decomposicaoQR (Matriz A, int n)
Matriz QT, Jij, Rnova, Rvelha, R;
// Inicializar matrizes
QT ← I;  // Esta matriz contém os produtos das matrizes ortogonais Jij (veja eq. (11))
Rvelha ← A;  // Na inicialização, Rvelha não tem a estrutura de uma matriz triangular superior
Para j = 1 ... (n-1) faça // loop das colunas
        Para i = (j+1) ... (n) faça  // loop das linhas
                // Construção da matriz de Jacobi Jij
                Jij ← matrizJacobiBaseadaNoElemento_ij_DeRvelha(Rvelha, i, j, n);
                // Matriz modificada com elemento (i,j) zerado
                Rnova ← Jij Rvelha;
                // Salvar para o próximo passo.
                Rvelha ← Rnova;
                // Acumular o produto das matrizes de Jacobi como
                //  QT ← Jn(n−1) ⋯ Jn1 ⋯ J31 J21 I
                QT ← Jij QT;    // QT é a transposta de Q. Note a ordem do produto (eq. (11)).
        Fim Para
Fim Para
// No final do loop externo, o formato da matriz Rnova é triangular superior.
 Q ← Transposta(QT);
 R ← Rnova;
return (Q, R);
```

3.1.3 Estrutura da matriz R

Ao final da varredura da decomposição QR a matriz original é transformada em uma matriz triangular superior

$$(12)\quad \mathbf{R} = \left[\begin{pmatrix} a_{1,1} & a_{1,2} & \cdots & a_{1,n-2} & a_{1,n-1} & a_{1,n} \\ 0 & a_{2,2} & \cdots & a_{2,n-2} & a_{2,n-1} & a_{2,n} \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ 0 & 0 & \cdots & a_{n-2,n-2} & a_{n-2,n-1} & a_{n-2,n} \\ 0 & 0 & \cdots & 0 & a_{n-1,n-1} & a_{n-1,n} \\ 0 & 0 & \cdots & 0 & 0 & a_{n,n} \end{pmatrix}\right]$$

Note que, em todas as colunas, exceto na última, há elementos abaixo do elemento da diagonal principal com 0. Logo, para transformar a matriz original para esse formato, todos os elementos abaixo do elemento da diagonal principal em cada coluna **exceto na última** são zerados. Portanto, o loop j das colunas vai até a penúltima coluna. O loop das linhas para uma dada coluna j, percorre todos os elementos abaixo da linha da diagonal, ou seja, da linha (j+1) até a

3.1.4 Construção da matriz de Jacobi $\mathbf{J}_{ij}$

Vamos repetir a equação (11) aqui e observar que a matriz $\mathbf{J}_{ij}$ atua na matriz resultante dos parênteses à direita, ou seja, na matriz resultante dos passos anteriores.

$$(13)\quad \begin{aligned} \mathbf{R} &= \mathbf{J}_{n(n-1)} \cdots \left(\mathbf{J}_{n1} \cdots \left(\mathbf{J}_{31}(\mathbf{J}_{21}\mathbf{A})\right)\right) \\ &= \mathbf{Q}^{T}\mathbf{A} \end{aligned}$$

Assim, após a multiplicação

$$(14)\quad \mathbf{A}_{nova} \leftarrow \mathbf{J}_{ij}\mathbf{A}_{velha},$$

o elemento $(\mathbf{A}_{nova})_{i,j}$ ficará igual a zero.

Esse fato é levado em conta no método
Matriz **matrizJacobiBaseadaNoElemento_ij_DeRvelha**$(\mathbf{R}_{\mathbf{velha}}, \mathbf{i}, \mathbf{j}, \mathbf{n})$.

3.1.4.1 Matiz de rotação de Jacobi

A matriz de Jacobi é uma matriz de rotação. Isso significa que, dados os valores i e j, onde j é o índice da coluna e i é o índice da linha do elemento a ser zerado, isto é, do elemento $(\mathbf{A}_{\text{nova}})_{\text{i,j}}$, e sabendo que esse elemento da coluna j está abaixo da diagonal, isto é, i > j, sua estrutura é representada como

$$
\begin{aligned}
\mathbf{J}_{ij} &= \left[\begin{pmatrix}
1 & \cdots & 0 & \cdots & 0 & \cdots & 0 & \cdots & 0 \\
0 & \ddots & 0 & \cdots & 0 & \cdots & 0 & \cdots & 0 \\
0 & \cdots & \text{el}_{\text{j,j}} & \cdots & 0 & \cdots & \text{el}_{\text{j,i}} & \cdots & 0 \\
\vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \vdots & \vdots \\
0 & \cdots & 0 & \cdots & 1 & \cdots & 0 & \cdots & 0 \\
\vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\
0 & \cdots & \text{el}_{\text{i,j}} & \cdots & 0 & \cdots & \text{el}_{\text{i,i}} & \cdots & 0 \\
 & \cdots & \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\
0 & \cdots & 0 & \cdots & 0 & \cdots & 0 & \cdots & 1
\end{pmatrix}\right] \\
&= \left[\begin{pmatrix}
1 & \cdots & 0 & \cdots & 0 & \cdots & 0 & \cdots & 0 \\
0 & \ddots & 0 & \cdots & 0 & \cdots & 0 & \cdots & 0 \\
0 & \cdots & \cos(\theta) & \cdots & 0 & \cdots & -\text{sen}(\theta) & \cdots & 0 \\
\vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \vdots & \vdots \\
0 & \cdots & 0 & \cdots & 1 & \cdots & 0 & \cdots & 0 \\
\vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\
0 & \cdots & \text{sen}(\theta) & \cdots & 0 & \cdots & \cos(\theta) & \cdots & 0 \\
 & \cdots & \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\
0 & \cdots & 0 & \cdots & 0 & \cdots & 0 & \cdots & 1
\end{pmatrix}\right]
\end{aligned}
\tag{15}
$$

Note que a construção da matriz é bem simples: basta inicializar a matriz $\mathbf{J}_{ij}$ com a **matriz identidade**, e alterar os elementos $\text{el}_{\text{j,j}} = \text{el}_{\text{i,i}} = \cos(\theta)$, $\text{el}_{\text{i,j}} = \text{sen}(\theta)$, e $\text{el}_{\text{j,i}} = -\text{sen}(\theta)$.

Porém, vem a pergunta: **Qual é o valor de $\boldsymbol{\theta}$ a ser usado**?

Para responder essa pergunta, vamos escrever a equação (14) explicitamente, usando a equação (15), isto é, no produto da equação (14), o elemento

$$(\mathbf{A}_{\text{nova}})_{\text{i,j}} = 0 = (\mathbf{J}_{\text{ij}})_{\text{i,k}} (\mathbf{A}_{\text{velha}})_{\text{k,j}} \tag{16}$$

Na equação (16), usamos a notação indicial, em que, índices repetidos implicam um somatório.

O produto, com **k repetido**, equivale ao **produto escalar** da linha i da matriz $[\mathbf{J}_{\text{ij}}]$ pela coluna j da matriz $[\mathbf{A}_{\text{velha}}]$ resultando em

$$
(17)\quad
\begin{aligned}
(\mathbf{A}_{\mathrm{nova}})_{i,j} &= (\mathbf{J}_{ij})_{i,k}(\mathbf{A}_{\mathrm{velha}})_{k,j} \\
0 &= \mathrm{el}_{i,j}(\mathbf{A}_{\mathrm{velha}})_{j,j} + \mathrm{el}_{i,i}(\mathbf{A}_{\mathrm{velha}})_{i,j} \quad / \\
0 &= \mathrm{sen}\,(\theta)(\mathbf{A}_{\mathrm{velha}})_{j,j} + \cos\,(\theta)(\mathbf{A}_{\mathrm{velha}})_{i,j}
\end{aligned}
$$

// A linha i da matriz $\mathbf{J}_{ij}$
// só tem dois elementos
// não nulos.

Assim, da equação (17) pode-se concluir que

$$
(18)\quad \mathrm{tg}(\theta) = \frac{\mathrm{sen}(\theta)}{\cos(\theta)} = -\frac{(\mathbf{A}_{\mathrm{velha}})_{i,j}}{(\mathbf{A}_{\mathrm{velha}})_{j,j}}
$$

A equação (18) nos dá a tangente do ângulo θ. Vamos analisar essa equação.

$$
(19)\quad \mathrm{tg}(\theta) = -\frac{(\mathbf{A}_{\mathrm{velha}})_{i,j}}{(\mathbf{A}_{\mathrm{velha}})_{j,j}} \Longrightarrow
\begin{cases}
\text{se} \geq 0, \text{considerar } 0 \leq \theta < \frac{\pi}{2} \\
\text{se} < 0, \text{considerar } -\frac{\pi}{2} < \theta < 0 \\
(\mathbf{A}_{\mathrm{velha}})_{j,j} = 0, \text{considerar } \theta = \frac{\pi}{2}.
\end{cases}
$$

3.1.4.2 Algoritmo de construção da matriz de Jacobi, $\mathbf{J}_{ij}$

Matriz **matrizJacobiBaseadaNoElemento_ij_DeRvelha**(Matriz $\mathbf{A}$, int i, int j, int n)
Matriz $\mathbf{I}, \mathbf{J}_{ij}$;
Real $\theta, \varepsilon = 10^{-6}$;
// Inicializar vetores
$\mathbf{J}_{ij} \leftarrow \mathbf{I}$; // Matriz identidade com n x n elementos
Se $(\mathrm{abs}(\mathbf{A}_{i,j}) \leq \varepsilon)$ **return** $\mathbf{J}_{ij}$ // Considerar $\mathbf{A}_{i,j} = 0$, retorna a matriz identidade
// Calcular θ
Se $(\mathrm{abs}((\mathbf{A})_{j,j}) \leq \varepsilon)$ // Considerar $(\mathbf{A}_{\mathrm{velha}})_{j,j} = 0$ **então**
  Se ($(\mathbf{A})_{i,j} < 0$) // O numerador será positivo e assumimos tangente tende a +Inf
    $\theta = \frac{\pi}{2}$;
  Senão // O numerador será negativo e assumimos tangente tende a -Inf
    $\theta = -\frac{\pi}{2}$;
  Fim Senão
Senão
  $\theta = \arctan\left(\frac{-(\mathbf{A})_{i,j}}{(\mathbf{A}_{j,j})}\right)$; // Esta função já retorna um ângulo +/-
    // no primeiro quadrante sentido anti-horário (+)
    // no primeiro quadrante sentido horário (-)
Fim Se-Senão
$\mathbf{J}_{ij}(i,i) \leftarrow \cos(\theta)$
$\mathbf{J}_{ij}(j,j) \leftarrow \cos(\theta)$
$\mathbf{J}_{ij}(i,j) \leftarrow \mathrm{sen}(\theta)$
$\mathbf{J}_{ij}(j,i) \leftarrow -\mathrm{sen}(\theta)$

return ($\mathbf{J}_{ij}$);

3.1.5 A Varredura de decomposição QR preserva os termos zerados

Suponha que a matriz de Jacobi $\mathbf{J}_{ij}$ tenha sido construída para, através da equação (16), zerar o elemento $(\mathbf{A}_{\mathrm{nova}})_{i,j}$. O que acontece com os termos que já tinham sido zerados em passos anteriores da varredura? Para responder isso, vamos analisar o que acontece com esses termos quando o produto da equação (16) é efetuado usando $\mathbf{J}_{ij}$, ou seja,

$$
(20)\quad (\mathbf{A}_{\mathrm{nova}})_{r,s} = (\mathbf{J}_{ij})_{r,k}(\mathbf{A}_{\mathrm{velha}})_{k,s}.
$$

A varredura de Jacobi começa sempre com a coluna j=1. Vamos assumir inicialmente que, na equação (20), s=j. Essa condição acontece logo no primeiro passo no loop das colunas s=j=1. Assim, o que acontece com os elementos entre a diagonal (j, j) e o elemento (i, j) que acaba de ser zerado?

Caso 1) s = j

$$(\mathbf{A}_{\text{nova}})_{r,j} = (\mathbf{J}_{ij})_{r,k}(\mathbf{A}_{\text{velha}})_{k,j}$$

1.1) j < r < i (**elementos da coluna j que deveriam permanecer zerados**)

$$\begin{aligned}(\mathbf{A}_{\text{nova}})_{r,j} &= (\mathbf{J}_{ij})_{r,r}(\mathbf{A}_{\text{velha}})_{r,j}\\ &= 1\,(\mathbf{A}_{\text{velha}})_{r,j}\\ &= (\mathbf{A}_{\text{velha}})_{r,j}\end{aligned}$$

Note que o elemento $(\mathbf{A}_{\text{nova}})_{r,j} = (\mathbf{A}_{\text{velha}})_{r,j}$ **. Logo esse termo permanece zero**.

1.2) j < r = i

$$\begin{aligned}(\mathbf{A}_{\text{nova}})_{i,j} &= (\mathbf{J}_{ij})_{k,i}\left((\mathbf{A}_{\text{velha}})_{k,j}\cos(\theta) + (\mathbf{A}_{\text{velha}})_{k,i}\operatorname{sen}(\theta)\right)\\ &= 0\end{aligned}$$

1.3) i < r

$$\begin{aligned}(\mathbf{A}_{\text{nova}})_{r,j} &= (\mathbf{J}_{ij})_{r,r}(\mathbf{A}_{\text{velha}})_{r,j}\\ &= 1\,(\mathbf{A}_{\text{velha}})_{r,j}\\ &= (\mathbf{A}_{\text{velha}})_{r,j}\end{aligned}$$

Caso 2) s < j (elementos em colunas anteriores à coluna j que deveriam permanecer zerados)

$$(\mathbf{A}_{\text{nova}})_{r,s} = (\mathbf{J}_{ij})_{r,k}(\mathbf{A}_{\text{velha}})_{k,s}$$

2.1) s < r < j

$$\begin{aligned}(\mathbf{A}_{\text{nova}})_{r,s} &= (\mathbf{J}_{ij})_{r,r}(\mathbf{A}_{\text{velha}})_{r,s}\\ &= 1\,(\mathbf{A}_{\text{velha}})_{r,s}\\ &= (\mathbf{A}_{\text{velha}})_{r,s}\end{aligned}$$

2.2) r = j

$$\begin{aligned}(\mathbf{A}_{\text{nova}})_{j,s} &= (\mathbf{J}_{ij})_{j,j}(\mathbf{A}_{\text{velha}})_{j,s} + (\mathbf{J}_{ij})_{j,i}(\mathbf{A}_{\text{velha}})_{i,s}\\ &= \cos(\theta)\quad 0 \qquad -\operatorname{sen}(\theta)\quad 0\\ &= 0\end{aligned}$$

2.3) j < r < i

$$\begin{aligned}(\mathbf{A}_{\text{nova}})_{r,s} &= (\mathbf{J}_{ij})_{r,r}(\mathbf{A}_{\text{velha}})_{r,s}\\ &= 1\,(\mathbf{A}_{\text{velha}})_{r,s}\\ &= (\mathbf{A}_{\text{velha}})_{r,s}\end{aligned}$$

2.4) $j < r = i$

$$\begin{aligned}(\mathbf{A}_{\text{nova}})_{i,s} &= (\mathbf{J}_{ij})_{i,j}(\mathbf{A}_{\text{velha}})_{j,s} + (\mathbf{J}_{ij})_{i,i}(\mathbf{A}_{\text{velha}})_{i,s} \\ &= \text{sen}(\theta)\quad 0 \qquad + \cos(\theta)\quad 0 \\ &= 0\end{aligned}$$

$i < r$

$$(\mathbf{A}_{\text{nova}})_{r,s} = (\mathbf{J}_{ij})_{r,r}(\mathbf{A}_{\text{velha}})_{r,s} = (\mathbf{A}_{\text{velha}})_{r,s}$$

Tarefa #15:

Com a matriz **A** abaixo, faça o que se pede:

$$\mathbf{A} = \begin{bmatrix} 40 & 8 & 4 & 2 & 1 \\ 8 & 30 & 12 & 6 & 2 \\ 4 & 12 & 20 & 1 & 2 \\ 2 & 6 & 1 & 25 & 4 \\ 1 & 2 & 2 & 4 & 5 \end{bmatrix}$$

1) Implemente o método QR e aplique-o sobre **A** para encontrar

i. a matriz diagonal $\bar{\mathbf{A}}$

ii. a matriz acumulada $\mathbf{P} = \mathbf{Q_1 Q_2 Q_3}$ ...

iii. Imprima a matriz que sai de cada iteração QR

iv. Imprima os pares (autovalor, autovetor) da matriz **A**

v. Compare os resultados do item 3) com os resultados da Tarefa #14.

2) Adapte o método QR para receber a matriz tridiagonal que sai do método de Householder. Neste caso, observe que:

i. imprima $\mathbf{A}_{\text{nova}}$ que sai de cada iteração QR e verifique se os termos que eram zero deixaram de ser zero.

Ii . depois da diagonalização, verifique que as colunas de **P** não são os autovetores de **A**.

iv. Faça $\mathbf{P} = \mathbf{HP}$ e verifique que as colunas da nova matriz **P** são os autovetores de **A**