



## Artigo de Divulgação

# Teorema de Eckart-Young-Mirsky: Decomposição em Produtos de Kronecker

**João Socorro Pinheiro Ferreira**

Universidade Federal do Amapá

[joaoferreira@unifap.br](mailto:joaoferreira@unifap.br)

**Tomy Felixon**

Universidade Estadual de Campinas

[t123735@dac.unicamp.br](mailto:t123735@dac.unicamp.br)

**Fabiana Correia Pereira**

Universidade Estadual de Campinas

[f262293@dac.unicamp.br](mailto:f262293@dac.unicamp.br)

## Resumo

O objetivo deste trabalho é apresentar uma decomposição em produtos de Kronecker utilizando a Decomposição em Valores Singulares (SVD), bem como sua relação com o Teorema de Eckart-Young-Mirsky. Para tanto, realizamos uma pesquisa bibliográfica e demonstramos as principais propriedades da SVD e do produto de Kronecker. Além disso, utilizamos outros métodos para a decomposição, como método das potências e reflexões de Householder a fim de obter a decomposição QR de matrizes simétricas. É importante observar que uma compreensão sólida dos fundamentos da álgebra linear é essencial para acompanhar as discussões desenvolvidas ao longo deste trabalho.

**Palavras-chaves:** Decomposição QR. Medida de Emaranhamento. Método das Potências. Reflexões de Householder. SVD.

## Abstract

The objective of this work is to present a decomposition into Kronecker products using the Singular Value Decomposition (SVD), as well as its relationship with the Eckart-Young-Mirsky Theorem. To this end, we carried out a literature search and demonstrated the main properties of SVD and Kronecker's product. Furthermore, we use other methods for decomposition, such as the power method and Householder reflections in order to obtain the QR decomposition of symmetric matrices. It is important to note that a solid understanding of the fundamentals of linear algebra is essential to follow the discussions developed throughout this work.

**Keywords:** QR Decomposition. Entanglement Measurement. Powers Method. Householder Reflections. SVD.

## 1 Introdução

O produto de Kronecker, nomeado em homenagem ao matemático alemão Leopold Kronecker, que fez contribuições importantes à Álgebra e à Teoria dos Números no século XIX ([HORN e JOHNSON, 1991](#)). Segundo ([GENTLE, 2007](#)), é uma operação matemática que combina duas matrizes para produzir uma terceira matriz maior. Esse produto tem aplicações relevantes em diversas áreas da Matemática e das Ciências, como Álgebra Linear ([STEWART, 1993](#)), Física, Processamento de Sinais e Aprendizado de Máquina ([VAN LOAN, 2000](#)). Neste trabalho, daremos enfoque à Álgebra Linear, destacando conceitos fundamentais que serão recorrentes ao longo da discussão e aplicações. Para os leitores que desejam revisar os conceitos básicos dessa área, sugerimos as referências de ([BOLDRINI et al., 1980](#)) e ([ANTON e RORRES, 2012](#)).

Neste trabalho, vamos apresentar as principais propriedades do produto de Kronecker, destacando o problema de matriz de posto baixo, que podemos relacionar diretamente com a Decomposição em Valores Singulares (SVD) ([ECKART e YOUNG, 1936](#)). Também, vamos aplicar a SVD para encontrar os autovalores e autovetores de uma matriz quadrada através de métodos iterativos. A demonstração da decomposição SVD está na Seção 2 e de sua

forma reduzida, juntamente com a inversa SVD de matrizes não singulares. ([HORN e JOHNSON, 1991; VAN LOAN, 2000; GOLUB e KAHAN, 1978](#)).

Na Seção 3 estão a formulação do produto de Kronecker e o Teorema 3.1 de Eckart-Young-Mirsky sobre a matriz de posto baixo  $q < p$ . O posto  $p$  é da matriz A. ([SAA, 2023; VAN LOAN e GOLUB, 1996](#))

Na Seção 4 são apresentados e discutidos os resultados de cinco situações problemas com a fundamentação teórica deste artigo, dentre as quais o produto de Kronecker, matriz de posto baixo, medida de emaranhamento, algoritmos QR iterativos - com transformações similares ou semelhantes de matrizes, método de iteração ortogonal, algoritmo para iteração de subespaços e decomposições QR utilizando transformações de Householder, Hessenberg e Gram-Schmidt. As situações problemas são solucionadas por intermédio de algoritmos e seus respectivos códigos fonte no MATLAB, com checagem pra validar os resultados. ([HORN e JOHNSON, 1991; SAA, 2023; STEWART, 1993; VAN LOAN, 2000; VAN LOAN e GOLUB, 1996](#))

Finalmente, na Seção 5 está a Conclusão deste artigo descrevendo sinteticamente os principais resultados obtidos nesta pesquisa. A parte pós-textual contém os Apêndices A com o resultado para matriz de posto baixo com  $q = 1$ , o Apêndice B - para  $q = 2$  e o Apêndice C, com resultados sobre medidas de emaranhamento .

## 2 Decomposição SVD

Antes de fazermos a demonstração da Decomposição de Valores Singulares (SVD), simbolizado por  $\sigma_i$ , para  $i = 1, \dots, m$ , vale destacar o artigo científico *On the Early History of the Singular Value Decomposition*<sup>1</sup> de ([STEWART, 1993](#)), que tem por objetivo relatar a contribuição de cinco eminentes matemáticos: Eugênio Beltrami (1835-1899), Camille Jordan (1838-1921), James Joseph Sylvester (1814-1897), Erhard Schmidt (1876-1959) e Herman Weyl (1885-1955) - que foram responsáveis por estabelecer a SVD e desenvolver a sua teoria. Os primeiros registros datam de 1839.

Sejam  $A \in \mathbb{R}^{m \times n}$ ,  $U \in \mathbb{R}^{m \times m}$ ,  $\Sigma \in \mathbb{R}^{m \times n}$  e  $V^T \in \mathbb{R}^{n \times n}$ . A matriz A pode ser decomposta sob a forma Decomposição de Valores Singulares (SVD), do inglês *Singular Value Decomposition*:

$$A = U\Sigma V^T, \quad (1)$$

em que as matrizes U e V são ortogonais unitárias. A matriz  $\Sigma$  é uma matriz diagonal formada por valores singulares. Os valores singulares estão relacionados com os autovalores não nulos de  $A^T A$  e  $AA^T$  ( escolher a que tiver menor dimensão, ou então, qual delas que for mais esparsa<sup>2</sup>, se for o caso).

Para estudarmos a SVD, precisaremos do Teorema 2.1, Corolário 2.1 e do Lema 2.1.

### Teorema 2.1

Se  $v$  é autovetor de  $A^T A$  associado a  $\lambda \neq 0 \implies Av$  é autovetor de  $AA^T$  associado a  $\lambda$ .

### Demonstração:

$$(AA^T)Av = A(A^T A)v = A\lambda v = \lambda Av.$$

■

### Observação 2.1

$p = \text{posto}(A) = \text{posto}(A^T A) = \text{posto}(AA^T)$ .

### Corolário 2.1

$A^T A$  e  $AA^T$ , ambas tem o mesmo conjunto de autovalores não nulos, contendo as mesmas multiplicidades.

<sup>1</sup>Sobre o Histórico Inicial da SVD.

<sup>2</sup>Uma matriz é dita esparsa quando possui uma grande quantidade de elementos com valor zero.

Seja  $p = \text{posto}(\mathbf{A}^T \mathbf{A}) = \text{posto}(\mathbf{A} \mathbf{A}^T)$ . Os demais autovalores de  $\mathbf{A}^T \mathbf{A}$  e  $\mathbf{A} \mathbf{A}^T$  são nulos com multiplicidade adequada.

### Lema 2.1

Todos os autovalores de  $\mathbf{A} \mathbf{A}^T$  são não negativos.

**Demonstração:** Seja  $\mathbf{A} \in \mathbb{R}^{m \times n}$ . Consideremos o produto  $\mathbf{A} \mathbf{A}^T \in \mathbb{R}^{m \times m}$ , que é uma matriz simétrica, pois

$$(\mathbf{A} \mathbf{A}^T)^T = (\mathbf{A}^T)^T \mathbf{A}^T = \mathbf{A} \mathbf{A}^T.$$

Como  $\mathbf{A} \mathbf{A}^T$  é simétrica, seus autovalores são reais. Agora, vamos mostrar que esses autovalores são não-negativos. Seja  $\lambda$  um autovalor de  $\mathbf{A} \mathbf{A}^T$  com autovetor associado  $\mathbf{u} \in \mathbb{R}^m$ . Temos

$$\mathbf{A} \mathbf{A}^T \mathbf{u} = \lambda \mathbf{u}.$$

Multiplicando à esquerda ambos os lados por  $\mathbf{u}^T$ , obtemos

$$\mathbf{u}^T \mathbf{A} \mathbf{A}^T \mathbf{u} = \lambda \mathbf{u}^T \mathbf{u}.$$

Como  $\mathbf{u}^T \mathbf{u} = \|\mathbf{u}\|^2 > 0$  (já que  $\mathbf{u} \neq 0$ ) e  $\mathbf{u}^T \mathbf{A} \mathbf{A}^T \mathbf{u} = \|\mathbf{A}^T \mathbf{u}\|^2 \geq 0$ , segue que

$$\lambda \mathbf{u}^T \mathbf{u} = \lambda \|\mathbf{u}\|^2 = \|\mathbf{A}^T \mathbf{u}\|^2 \geq 0.$$

Portanto,  $\lambda \geq 0$ . Assim, todos os autovalores de  $\mathbf{A} \mathbf{A}^T$  são não-negativos. ■

### Teorema 2.2: Decomposição em Valores Singulares (SVD)

Seja  $\mathbf{A}$  uma matriz  $m \times n$  de  $\text{posto}(\mathbf{A}) = p$ . Então existem números reais  $\sigma_1 \geq \sigma_2 \geq \dots \geq \sigma_p > 0$ , uma base ortonormal  $\mathbf{v}_1, \dots, \mathbf{v}_n$  de  $\mathbb{R}^n$  e uma base ortonormal  $\mathbf{u}_1, \dots, \mathbf{u}_m$  de  $\mathbb{R}^m$  tais que,

$$\begin{aligned} \mathbf{A} \mathbf{v}_k &= \sigma_k \mathbf{u}_k & k = 1, \dots, p & \quad \mathbf{A} \mathbf{v}_k &= 0 & k = p + 1, \dots, n \\ \mathbf{A}^T \mathbf{u}_k &= \sigma_k \mathbf{v}_k & k = 1, \dots, p & \quad \mathbf{A}^T \mathbf{u}_k &= 0 & k = p + 1, \dots, m \end{aligned}$$

Os vetores  $\mathbf{v}_1, \dots, \mathbf{v}_n$  são autovetores de  $\mathbf{A}^T \mathbf{A}$ ,  $\mathbf{u}_1, \dots, \mathbf{u}_m$  são autovetores de  $\mathbf{A} \mathbf{A}^T$  e  $\sigma_1^2, \dots, \sigma_p^2$  são autovalores não nulos de  $\mathbf{A}^T \mathbf{A}$  e  $\mathbf{A} \mathbf{A}^T$ .

### Demonstração:

Sejam  $\mathbf{v}_1, \dots, \mathbf{v}_n$  autovetores ortonormais de  $\mathbf{A}^T \mathbf{A}$  e  $\lambda_1 \geq \lambda_2 \geq \dots \geq \lambda_n$  os autovalores associados. A matriz quadrada  $\mathbf{A}^T \mathbf{A}$  é semidefinida positiva (SDP), portanto,  $\lambda_i \geq 0$ . O posto ( $\mathbf{A}$ ) =  $\text{posto}(\mathbf{A} \mathbf{A}^T) = p$ . Então,  $\lambda_p > 0$  e  $\lambda_s = 0$ ,  $s = p + 1, \dots, n$ . Os autovalores são as raízes do polinômio característico de  $\mathbf{A}^T \mathbf{A}$  ou  $\mathbf{A} \mathbf{A}^T$ . Deve-se escolher aquela que tem a menor dimensão ou aquela que for menos esparsa. O polinômio característico é definido por

$$P(\lambda) = \det(\mathbf{A}^T \mathbf{A} - \lambda \mathbf{I}).$$

Os vetores singulares à direita de  $\mathbf{A}$ , são os  $\mathbf{v}_k$ ,  $k = 1, \dots, n$  são determinados por

$$(\mathbf{A}^T \mathbf{A} - \lambda_k \mathbf{I}_n) \mathbf{v}_k = \mathbf{0}, \quad k = 1, \dots, n. \quad (2)$$

e  $\mathbf{u}_1, \dots, \mathbf{u}_m$  são chamados vetores singulares a esquerda de  $\mathbf{A}$ .

$$\mathbf{A} \mathbf{v}_k = \sigma_k \mathbf{u}_k, \text{ como } \|\mathbf{u}_k\| = 1.$$

Logo,

$$\mathbf{A}\mathbf{v}_k = \sigma_k.$$

Os valores singulares são determinados por

$$\sigma_k^2 = \|\mathbf{A}\mathbf{v}_k\|_2^2 = (\mathbf{A}\mathbf{v}_k)^T(\mathbf{A}\mathbf{v}_k) = \mathbf{v}_k^T(\mathbf{A}^T\mathbf{A}\mathbf{v}_k) = \lambda_k \mathbf{v}_k^T \mathbf{v}_k = \lambda_k.$$

Portanto, os valores singulares são determinados por

$$\sigma_k = \sqrt{\lambda_k}, \quad k = 1, \dots, p, \quad (3)$$

com  $\sigma_1 \geq \sigma_2 \geq \dots \geq \sigma_p > 0$ .

O conjunto completo correspondente de vetores singulares a esquerda dado por

$$\mathbf{u}_k = \frac{\mathbf{A}\mathbf{v}_k}{\|\mathbf{A}\mathbf{v}_k\|}, \quad k = 1, \dots, p.$$

Seja  $\sigma_k = \|\mathbf{A}\mathbf{v}_k\|_2$  e

$$\mathbf{u}_k = \frac{1}{\sigma_k} \mathbf{A}\mathbf{v}_k, \quad k = 1, \dots, p. \quad (4)$$

Falta definir  $\mathbf{u}_{p+1}, \dots, \mathbf{u}_m$ .  $\mathcal{N}(\mathbf{A}\mathbf{A}^T)$  tem dimensão  $(m - p)$ . Seja  $\mathbf{u}_{p+1}, \dots, \mathbf{u}_m$  uma base ortonormal desse subespaço. Esses serão os autovetores de  $\mathbf{A}\mathbf{A}^T$  associados aos autovalores nulos. Além disso, esses vetores serão ortogonais a  $\mathbf{u}_1, \dots, \mathbf{u}_p$ . Ou seja, para calcular os vetores da matriz ortogonal  $\mathbf{U}_2 : (p + 1) \times (m - n)$ ,

$$\mathbf{A}^T \mathbf{u}_k = \mathbf{0}, \quad k = (p + 1), \dots, m. \quad (5)$$

A matriz  $\mathbf{U} = [\mathbf{u}_1, \dots, \mathbf{u}_p, \mathbf{u}_{p+1}, \dots, \mathbf{u}_m]$ , particionada é

$$\mathbf{U} = \left[ \begin{array}{cc} \mathbf{U}_1 & \mathbf{U}_2 \end{array} \right]_{m \times m}.$$

onde  $\mathbf{U}_1 = [\mathbf{u}_1, \dots, \mathbf{u}_p]$  e  $\mathbf{U}_2 = [\mathbf{u}_{p+1}, \dots, \mathbf{u}_m]$ .

A construção da matriz  $\mathbf{V}$ , parte do fato de que

$$\mathbf{A}^T \mathbf{u}_k = \mathbf{A}^T \mathbf{A} \frac{\mathbf{v}_k}{\sigma_k} = \frac{\lambda_k}{\sigma_k} \mathbf{v}_k = \frac{\sigma_k^2}{\sigma_k} \mathbf{v}_k = \sigma_k \mathbf{v}_k.$$

Para calcular os vetores da matriz ortogonal  $\mathbf{V}_1 : m \times p$ ,

$$\mathbf{v}_k = \frac{1}{\sigma_k} \mathbf{A}^T \mathbf{u}_k, \quad k = 1, \dots, p. \quad (6)$$

Para calcular os vetores da matriz ortogonal  $\mathbf{V}_2 : (p + 1) \times n$ ,

$$\mathbf{A}\mathbf{v}_k = \mathbf{0}, \quad k = (p + 1), \dots, m. \quad (7)$$

Se  $\mathbf{A}$  é uma matriz real  $m$  por  $n$ , então existem matrizes ortogonais

$$\mathbf{U} = \left[ \begin{array}{ccc} \mathbf{u}_1, & \dots & \mathbf{u}_m \end{array} \right] \in \mathbb{R}^{m \times m} \quad e \quad \mathbb{V} = \left[ \begin{array}{ccc} \mathbf{v}_1, & \dots & \mathbf{v}_n \end{array} \right] \in \mathbb{R}^{n \times n}$$

tal que

$$\mathbf{U}^T \mathbf{A} \mathbb{V} = \text{diag}(\sigma_1, \dots, \sigma_p) \in \mathbb{R}^{m \times n} \quad p = \min\{m, n\},$$

onde  $\sigma_1 \geq \sigma_2 \geq \dots \geq \sigma_p \geq 0$ .

As matrizes  $\mathbf{A}^T \mathbf{A}$  e  $\mathbf{A}\mathbf{A}^T$  são diagonalizáveis e podem ser decompostas nas matrizes  $\mathbf{U}$ ,  $\mathbf{D}$  e  $\mathbb{V}$  que podem ser

particionadas na seguinte forma

$$\mathbf{U} = [\mathbf{U}_1 \ \mathbf{U}_2] \quad \mathbf{D} = \begin{bmatrix} \mathbf{D}_1 & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix} \quad \mathbf{V} = [\mathbf{V}_1 \mathbf{V}_2]$$

de modo que  $\mathbf{A} = \mathbf{U}_1 \mathbf{D}_1 \mathbf{V}_1^T$  é a forma reduzida da SVD de  $\mathbf{A}$ ; e a matriz

$$\mathbf{D}_1 = \begin{bmatrix} \sigma_1 & & & \\ & \sigma_2 & & \\ & & \ddots & \\ & & & \sigma_p \end{bmatrix}.$$

### Teorema 2.3

Seja  $\mathbf{A}^{m \times n}$  e  $\text{posto}(\mathbf{A}) = p$ . Então,

$$\mathbf{A} = \sum_{k=1}^p \sigma_k \mathbf{u}_k \mathbf{v}_k^T.$$

### Demonstração:

Pela decomposição SVD temos que

$$\begin{aligned} \mathbf{A}^{m \times n} & \left[ \begin{array}{cccccc} \mathbf{u}_1 & \cdots & \mathbf{u}_p & \mathbf{u}_{p+1} & \cdots & \mathbf{u}_m \end{array} \right]_{1 \times m} \begin{bmatrix} \sigma_1 & & & & & \\ & \ddots & & & & \\ & & \sigma_p & & & \\ & & & 0 & & \\ & & & & \ddots & \\ & & & & & 0 \end{bmatrix}_{m \times n} \begin{bmatrix} \mathbf{v}_1^T & & & & & \\ & \vdots & & & & \\ & & \mathbf{v}_p^T & & & \\ & & & \mathbf{v}_{p+1}^T & & \\ & & & & \vdots & \\ & & & & & \mathbf{v}_n^T \end{bmatrix}_{n \times 1} \\ & = \left[ \begin{array}{ccccc} \mathbf{u}_1 \sigma_1 & \cdots & \mathbf{u}_p \sigma_p & 0 & \cdots & 0 \end{array} \right]_{1 \times n} \begin{bmatrix} \mathbf{v}_1^T \\ \vdots \\ \mathbf{v}_p^T \\ \mathbf{v}_{p+1}^T \\ \vdots \\ \mathbf{v}_n^T \end{bmatrix}_{n \times 1} = \left[ \begin{array}{c} \sigma_1 \mathbf{u}_1 \mathbf{v}_1^T + \cdots + \sigma_p \mathbf{u}_p \mathbf{v}_p^T + 0 + \cdots + 0 \end{array} \right]_{1 \times p} \\ & = \left[ \begin{array}{c} \sigma_1 \mathbf{u}_1 \mathbf{v}_1^T + \cdots + \sigma_p \mathbf{u}_p \mathbf{v}_p^T \end{array} \right]_{1 \times p}. \end{aligned}$$

Pela última expressão verifica-se que

$$\mathbf{A} = \sum_{i=1}^p \sigma_i \mathbf{u}_i \mathbf{v}_i^T,$$

onde  $\mathbf{u}_k$  é a  $k$ -ésima coluna de  $\mathbf{U}$  e  $\mathbf{v}_k$  é a  $k$ -ésima coluna de  $\mathbf{V}$ . Sendo que

$$\mathbf{u}_1 = \begin{bmatrix} u_{11} \\ u_{21} \\ \vdots \\ u_{p1} \end{bmatrix}, \dots, \mathbf{u}_p = \begin{bmatrix} u_{1p} \\ u_{2p} \\ \vdots \\ u_{pp} \end{bmatrix}$$

e

$$\mathbf{v}_1^T = \begin{bmatrix} v_{11} & v_{21} & \cdots & v_{n1} \end{bmatrix}, \dots, \mathbf{v}_p^T = \begin{bmatrix} v_{1p} & v_{2p} & \cdots & v_{pp} \end{bmatrix}.$$

### 3 Produto de Kronecker

O produto tensorial de Kronecker possui uma álgebra rica e muito agradável que suporta uma ampla gama de algoritmos rápidos, elegantes e práticos. ([VAN LOAN, 2000](#))

Por simplicidade, vamos considerar apenas matrizes reais. Dadas duas matrizes  $\mathbf{A} \in \mathbb{R}^{n \times m}$  e  $\mathbf{B} \in \mathbb{R}^{p \times q}$ , seu produto de Kronecker  $\mathbf{A} \otimes \mathbf{B} \in \mathbb{R}^{(np) \times (mq)}$  é definido como

$$\mathbf{C} = \mathbf{A} \otimes \mathbf{B} = \begin{bmatrix} a_{11}\mathbf{B} & a_{12}\mathbf{B} & \cdots & a_{1m}\mathbf{B} \\ a_{21}\mathbf{B} & a_{22}\mathbf{B} & \cdots & a_{2m}\mathbf{B} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n1}\mathbf{B} & a_{n2}\mathbf{B} & \cdots & a_{nm}\mathbf{B} \end{bmatrix}. \quad (8)$$

As suas propriedades fundamentais são diretas:  $(\mathbf{A} \otimes \mathbf{B})^T = \mathbf{A}^T \otimes \mathbf{B}^T$ ,  $(\mathbf{A} \otimes \mathbf{B})^{-1} = \mathbf{A}^{-1} \otimes \mathbf{B}^{-1}$ ,  $(\mathbf{A} \otimes \mathbf{B})(\mathbf{D} \otimes \mathbf{E}) = \mathbf{AD} \otimes \mathbf{BE}$  e  $\mathbf{D} \otimes (\mathbf{A} \otimes \mathbf{B}) = (\mathbf{D} \otimes \mathbf{A}) \otimes \mathbf{B}$ . ([HORN e JOHNSON, 1991](#))

Na Subseção 4.1 implementamos o algoritmo para resolver (8).

#### 3.1 O problema de matriz de posto baixo

Vamos mostrar como o Teorema de Eckart-Young-Mirsky pode ser usado para resolver o problema da melhor aproximação em termos de um produto de Kronecker. Vamos supor por simplicidade que temos uma matriz  $\mathbf{C}_{6 \times 4}$  e queremos determinar qual é sua melhor aproximação em termos de um produto  $\mathbf{A}_{3 \times 2} \otimes \mathbf{B}_{2 \times 2}$ , com todos os caveats (ressalvas) com relação à unicidade deste problema. Deve-se minimizar

$$\mu(\mathbf{C}) = \left\| \begin{bmatrix} c_{11} & c_{12} & | & c_{13} & c_{14} \\ c_{21} & c_{22} & | & c_{23} & c_{24} \\ \hline c_{31} & c_{32} & | & c_{33} & c_{34} \\ c_{41} & c_{42} & | & c_{43} & c_{44} \\ \hline c_{51} & c_{52} & | & c_{53} & c_{54} \\ c_{61} & c_{62} & | & c_{63} & c_{64} \end{bmatrix} - \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \\ a_{31} & a_{32} \end{bmatrix} \otimes \begin{bmatrix} b_{11} & b_{12} \\ b_{21} & b_{22} \end{bmatrix} \right\|_2 \quad (9)$$

dada a matriz  $\mathbf{C}$ .

O Teorema de Eckart-Young-Mirsky, também conhecido como Teorema da Aproximação de Menor Posto, afirma que, dada uma matriz  $\mathbf{A}$  de posto  $p$ , a melhor aproximação de  $\mathbf{A}$  por uma matriz de posto  $q < p$  em relação à norma espectral (ou de Frobenius) é obtida pela decomposição em valores singulares reduzida da matriz  $\mathbf{A}$ , truncando-se o Teorema 2.3, para  $k = 1, \dots, q < p$ .

#### Teorema 3.1: Eckart-Young-Mirsky

Seja  $\mathbf{A}_{m \times n}$  uma matriz de posto  $p$ . Para a norma espectral (idem para a de Frobenius),

$$\|\mathbf{A} - \mathbf{A}_q\|_2 \leq \|\mathbf{A} - \mathbf{B}\|_2,$$

para todas matrizes  $\mathbf{B}$  de posto  $q < p$ .

**Demonstração:** A demonstração encontra-se em ([SAA, 2023](#)). ■

Na subseção 4.2 escrevemos o algoritmo para encontrar a matriz  $\mathbf{B}$  de menor posto  $q$  que minimize a norma da diferença de uma matriz  $\mathbf{A}$  de posto  $p > q$ .

## 4 Resultados e Discussões

### 4.1 Produto de Kronecker

O Algoritmo 1 para determinar o produto de Kronecker definido na Equação (8) está desenvolvido a seguir:

---

#### Algoritmo 1 Algoritmo para o produto de Kronecker

---

```

1: Inicializar a matriz  $C_{pr \times qs}$ 
2: Para  $i \leftarrow 1$  até  $p$  faça
3:   Para  $j \leftarrow 1$  até  $q$  faça
4:      $row \leftarrow (i - 1) \cdot p + 1$ 
5:     Para  $k \leftarrow 1$  até  $r$  faça
6:        $col \leftarrow (j - 1) \cdot s + 1$ 
7:       Para  $l \leftarrow 1$  até  $s$  faça
8:          $C(row, col) \leftarrow A(i, j) \cdot B(k, l)$ 
9:          $col \leftarrow col + 1$ 
10:    Fim Para
11:    $row \leftarrow row + 1$ 
12: Fim Para
13: Fim Para
14: Fim Para
15: Retorne  $C_{pr \times qs}$ 
```

---

A seguir, realizaremos os cálculos acima, para as mesmas matrizes, como forma de testar se o algoritmo está processando de fato

$$\mathbf{A} = \begin{bmatrix} 1 & 1 \\ 1 & -1 \\ 1 & 1 \end{bmatrix}_{3 \times 2},$$

de posto  $p = 2$  e a matriz

$$\mathbf{B} = \begin{bmatrix} 1 & 2 \\ -2 & -1 \end{bmatrix}_{2 \times 2},$$

de posto  $q = 2$ , temos que o produto de Kronecker é ao utilizarmos o produto (8), para esta aplicação, temos

$$\mathbf{A} \otimes \mathbf{B} = \left[ \begin{array}{c|cc|c|cc} 1 \cdot & \begin{bmatrix} 1 & 2 \\ -2 & -1 \end{bmatrix} & 1 \cdot & \begin{bmatrix} 1 & 2 \\ -2 & -1 \end{bmatrix} & & \\ \hline & \begin{bmatrix} 1 & 2 \\ -2 & -1 \end{bmatrix} & & \begin{bmatrix} 1 & 2 \\ -2 & -1 \end{bmatrix} & & \\ \hline 1 \cdot & \begin{bmatrix} 1 & 2 \\ -2 & -1 \end{bmatrix} & (-1) \cdot & \begin{bmatrix} 1 & 2 \\ -2 & -1 \end{bmatrix} & & \\ \hline & \begin{bmatrix} 1 & 2 \\ -2 & -1 \end{bmatrix} & 1 \cdot & \begin{bmatrix} 1 & 2 \\ -2 & -1 \end{bmatrix} & & \end{array} \right] = \begin{bmatrix} 1 & 2 & 1 & 2 \\ -2 & -1 & -2 & -1 \\ 1 & 2 & -1 & -2 \\ -2 & -1 & 2 & 1 \\ 1 & 2 & 1 & 2 \\ -2 & -1 & -2 & -1 \end{bmatrix}_{6 \times 4} = \mathbf{C}.$$

utilizando o programa acima no MATLAB, produz o mesmo resultado:

$$\mathbf{C} = \begin{bmatrix} 1 & 2 & 1 & 2 \\ -2 & -1 & -2 & -1 \\ 1 & 2 & -1 & -2 \\ -2 & -1 & 2 & 1 \\ 1 & 2 & 1 & 2 \\ -2 & -1 & -2 & -1 \end{bmatrix}.$$

### 4.2 Matriz de posto baixo

Para responder à pergunta “[...] Em palavras simples, o problema da aproximação de posto baixo (lowrank approximation) pode ser formulado como: dada uma matriz arbitrária  $\mathbf{A}$  de posto  $p$ , que matriz de posto  $q < p$

pode ser considerada sua melhor aproximante? Com um pouco mais de precisão, qual é a matriz  $\mathbf{B}$  de posto mais baixo tal que  $\|\mathbf{A} - \mathbf{B}\|$  seja mínimo?" ([SAA, 2023](#))

---

**Algoritmo 2** Algoritmo de Eckart-Young-Mirsky

---

- 1: **Função** EckartYoungMirsky( $A, q$ )
    - 2:  $[U, S, V] \leftarrow$  Decomposição SVD de  $A$
    - 3:  $A_q \leftarrow U(:, 1 : q) \cdot S(1 : q, 1 : q) \cdot V(:, 1 : q)'$  ▷ Melhor aproximação de posto  $q$
    - 4:  $E \leftarrow A - A_q$
    - 5:  $\sigma \leftarrow \text{norma}(E, 2)$
    - 6:  $B \leftarrow A_q$
    - 7: **Retorne**  $B, \sigma$
    - 8: **Fim Função**
- 

Considerando-se que podemos escolher o posto  $q < p$ , então para a matriz  $\mathbf{A} = [-1 \ 0 \ 1; 0 \ -1 \ 1; 1 \ 1 \ 0]$ , que tem posto  $p = 3$ , o  $q$  pode assumir os seguintes valores: 1 e 2. Para o posto  $q = 1$ , o programa retorna a matriz que tem o posto mais baixo que é a seguinte:

$$\mathbf{B}_1 = \begin{bmatrix} -0.6667 & -0.6667 & 0.6667 \\ -0.6667 & -0.6667 & 0.6667 \\ 0.6667 & 0.6667 & -0.6667 \end{bmatrix}.$$

Quando  $q = 2$ , o programa retorna

$$\mathbf{B}_2 = \begin{bmatrix} -1.0000 & -0.5000 & 0.5000 \\ -0.0000 & -1.0000 & 1.0000 \\ 1.0000 & 0.5000 & -0.5000 \end{bmatrix}.$$

A matriz  $\mathbf{A}_q$  é  $\mathbf{A}_q = \sum_{k=1}^q \sigma_k \mathbf{u}_k \mathbf{v}_k^t$ . Para o posto  $q = 1$ ,  $\mathbf{A}_1 = \sum_{k=1}^1 \sigma_k \mathbf{u}_k \mathbf{v}_k^t = \sigma_1 \mathbf{u}_1 \mathbf{v}_1^T$ . Resumimos nas Tabelas 1 e 2 a análise das principais normas das duas matrizes.

**Tabela 1:** Comportamento das normas da matriz de posto mais baixo  $q = 1 < p$ .

Matriz	Posto	$\ \cdot\ _2$	$\ \cdot\ _F$
$\mathbf{A}$	3	2	$\sqrt{6}$
$\mathbf{B}_1$	1	2	2
$\mathbf{A} - \mathbf{A}_1$	2	1	$\sqrt{2}$
$\mathbf{A} - \mathbf{B}_1$	2	1.3333	$\sqrt{2}$

Fonte: elaborada pelos autores.

**Tabela 2:** Comportamento das normas da matriz de menor posto  $q = 2 < p$ .

Matriz	Posto	$\ \cdot\ _2$	$\ \cdot\ _F$
$\mathbf{A}$	3	2	2.4495
$\mathbf{B}_2$	2	2	2.2361
$\mathbf{A} - \mathbf{A}_2$	1	2	2.2361
$\mathbf{A} - \mathbf{B}_2$	1	1	1

Fonte: elaborada pelos autores.

Portanto, a matriz  $\mathbf{B}_k$ , para  $k = 1, 2$ , de posto mais baixo que minimiza a norma da matriz diferença  $\|\mathbf{A} - \mathbf{B}\|$ , i.e., que cumpre o Teorema 3.1 de Eckart-Young-Mirsky é a matriz  $\mathbf{B}_1$ , porque de acordo com a Tabela 1, as  $\|\mathbf{A} - \mathbf{A}_1\|_2 = 1$  e  $\|\mathbf{A} - \mathbf{B}_1\|_F = \sqrt{2}$  são, respectivamente, menores ou iguais de que as normas de  $\|\mathbf{A} - \mathbf{B}_1\|_2$  e  $\|\mathbf{A} - \mathbf{B}_1\|_F = \sqrt{2}$ , enquanto que para a matriz  $\mathbf{B}_2$ , ocorre o contrário, conforme à Tabela 2. ■

Os detalhes do resultados do MatLab para  $q = 1$  estão no Apêndice A e para  $q = 2$ , no Apêndice B.

### 4.3 Medida de emaranhamento

O termo “emaranhamento” geralmente não é aplicado diretamente a duas matrizes. Emaranhamento é um conceito que pertence à física quântica e está relacionado à correlação não clássica entre partículas subatômicas, como elétrons, fôtons e íons. O emaranhamento quântico ocorre quando o estado quântico de um sistema composto por múltiplas partículas não pode ser descrito independentemente dos estados das partículas individuais.

Em física quântica, as partículas emaranhadas estão intrinsecamente correlacionadas, de modo que as medições feitas em uma partícula afetam instantaneamente o estado da outra, independentemente da distância entre elas. Isso viola a intuição clássica e é um fenômeno fundamentalmente quântico.

A “medida” de emaranhamento de um estado  $\mathbf{C}$  será

$$\mu(\mathbf{C}) = \|\mathbf{C} - \mathbf{A} \otimes \mathbf{B}\|_2. \quad (10)$$

A principal tarefa será propor e implementar um algoritmo para calcular  $\mu(C)$  definida por (9) e de modo geral (10). Estamos interessados apenas em casos não triviais, então vamos excluir os casos nos quais pelo menos uma das matrizes “fatores”  $\mathbf{A}$  e  $\mathbf{B}$  é um escalar. Portanto, dado uma matriz  $\mathbf{C}_{m \times n}$ , vamos minimizar (9) para todos os possíveis produtos  $\mathbf{A}_{pq} \otimes \mathbf{B}_{rs}$  com  $m = pr$  e  $n = qs$ , excluindo-se o caso trivial  $p = q = 1$  e/ou  $r = s = 1$ .

Os resultados encontram-se no Apêndice C. Portanto, a medida de emaranhamento é:  $\mu(C) = 13.6381$ .

### 4.4 Algoritmo QR iterativo

Para estudarmos a decomposição QR por método iterativo - para determinar os autovalores e os autovetores associados, precisamos conhecer alguns fundamentos teóricos.

#### 4.4.1 Transformações similares ou semelhantes

As matrizes  $\mathbf{A}$  e  $\mathbf{B}$  são similares se existir a matriz  $\mathbf{P}$ , não singular, tal que  $\mathbf{B} = \mathbf{P}^{-1}\mathbf{AP}$ .

##### Teorema 4.1

Matrizes similares têm os mesmos autovalores.

##### Demonstração:

$$\begin{aligned} \det(\lambda\mathbf{I} - \mathbf{B}) &= \lambda\mathbf{P}^{-1}\mathbf{P} - \mathbf{P}^{-1}\mathbf{AP} &= \mathbf{P}^{-1}(\lambda\mathbf{I} - \mathbf{A})\mathbf{P} \\ \det(\lambda\mathbf{I} - \mathbf{B}) &= \det(\lambda\mathbf{P}^{-1}\mathbf{P} - \mathbf{P}^{-1}\mathbf{AP}) &= \det(\mathbf{P}^{-1}(\lambda\mathbf{I} - \mathbf{A})\mathbf{P}) \\ &= &= \det(\mathbf{P}^{-1}) \det(\lambda\mathbf{I} - \mathbf{A}) \det(\mathbf{P}) \\ &= &= \frac{1}{\det(\mathbf{P})} \det(\lambda\mathbf{I} - \mathbf{A}) \det(\mathbf{P}) \\ &= &= \det(\lambda\mathbf{I} - \mathbf{A}). \end{aligned}$$

Portanto,  $\mathbf{A}$  e  $\mathbf{B}$  têm os mesmos polinômios característicos. ■ ■

##### Teorema 4.2

Seja  $\mathbf{B} = \mathbf{P}^{-1}\mathbf{AP}$ . Se  $(\lambda, \mathbf{v})$  é autopar de  $\mathbf{A}$ , então  $(\lambda\mathbf{P}^{-1}\mathbf{v})$  é autovetor de  $\mathbf{B}$ .

**Demonstração:** Suponhamos que  $(\lambda, \mathbf{v})$  é autopar de  $\mathbf{A}$  então  $\mathbf{Av} = \lambda\mathbf{v}$ . Como  $\mathbf{B} = \mathbf{P}^{-1}\mathbf{AP}$ , então

$$\begin{aligned} \mathbf{BP}^{-1}\mathbf{v} &= \mathbf{P}^{-1}\mathbf{APP}^{-1}\mathbf{v} \implies \\ \mathbf{BP}^{-1}\mathbf{v} &= \mathbf{P}^{-1}\mathbf{Av} \implies \\ \mathbf{BP}^{-1}\mathbf{v} &= \mathbf{P}^{-1}\lambda\mathbf{v}, \quad \text{pois, } \mathbf{Av} = \lambda\mathbf{v} \implies \\ \mathbf{BP}^{-1}\mathbf{v} &= \lambda\mathbf{P}^{-1}\mathbf{v}. \end{aligned}$$

Portanto,  $(\lambda, \mathbf{P}^{-1}\mathbf{v})$  seja autopar de  $\mathbf{B}$ . ■ ■

#### 4.4.2 Método de Iteração Ortogonal

Generalização do método das potências para  $p$  colunas com  $\mathbf{Q}_{n \times p}^{(0)}$ , onde  $\mathbf{Q}$  tem colunas ortogonais.

Notem que o passo iterativo da decomposição QR é

$$\begin{aligned} \mathbf{A}_k &= \mathbf{Q}_k \mathbf{R}_k \\ \mathbf{A}_{k+1} &= \mathbf{R}_k \mathbf{Q}_k = \mathbf{Q}_k^T \mathbf{A} \mathbf{Q}_k. \end{aligned} \quad (11)$$

Por construção, o algoritmo, a cada passo, gera uma matriz  $\mathbf{A}_{k+1}$  similar à anterior  $\mathbf{A}_k$  e, portanto, que têm os mesmos autovalores. E fácil ver que começando-se com  $\mathbf{A}^{(0)} = \mathbf{A}$ , teremos

$$\mathbf{A}_{k+1} = \overbrace{\mathbf{Q}_k^T \cdots \mathbf{Q}_2^T \mathbf{Q}_1^T}^{\mathbf{P}_k^T} = \underbrace{\mathbf{Q}_1 \mathbf{Q}_2 \cdots \mathbf{Q}_k}_{\mathbf{P}_k}. \quad (12)$$

sendo  $\mathbf{P}_k$  uma matriz ortogonal. O ponto central do algoritmo é que sempre temos:

$$\lim_{k \rightarrow \infty} \mathbf{P}_k = \mathbf{U} \quad (13)$$

e, portanto,  $\lim_{k \rightarrow \infty} \mathbf{A}_k = \mathbf{D}$ , e o problema da diagonalização está resolvido.

O pseudocódigo abaixo implementa o método QR definido em (11):

---

#### Algoritmo 3 Algoritmo QR para diagonalização de matrizes

---

**Entrada:** Matriz  $\mathbf{A} \in \mathbb{R}^{n \times n}$

**Saída:** Autovetores (matriz ortogonal  $\mathbf{U}$ ) e autovalores de  $\mathbf{A}$  (diagonal da matriz  $\mathbf{A}$ ).

- 1:  $\mathbf{U} \leftarrow \mathbf{I}$
  - 2: **Para**  $k = 0$  até convergência **faça**
  - 3:   Realize a decomposição QR de  $\mathbf{A}$ :  $\mathbf{A} = \mathbf{QR}$
  - 4:   Atualize  $\mathbf{A}$ :  $\mathbf{A} \leftarrow \mathbf{RQ}$
  - 5:   Atualize  $\mathbf{U}$ :  $\mathbf{U} \leftarrow \mathbf{UQ}$
  - 6: **Fim Para**
- 

Os códigos a seguir, correspondem ao Algoritmo 3 que está sendo aplicado para diagonalizar uma matriz  $\mathbf{A}$  quadrada e exibir os elementos da matriz  $\mathbf{D}$  que são os autovalores de  $\mathbf{A}$ , com isto, a matriz  $\mathbf{D}$  será semelhante a matriz simétrica  $\mathbf{A}$ . Por exemplo: seja

$$\mathbf{A} = \begin{bmatrix} -1 & 0 & 1 \\ 0 & -1 & 1 \\ 1 & 1 & 0 \end{bmatrix}. \quad (14)$$

No Algoritmo 3, temos a decomposição QR de uma matriz quadrada, para determinar os autovalores e os autovetores associados. Nas Tabelas 3 e 4 estão as oito ( $k = 8$ ) iterações necessárias para determinar os três autovalores da matriz (14).

**Tabela 3:** Implementação do Algoritmo 3

Iterações (k)	1	2	3	4	5	6
Autovalores	-1.5000	-1.8333	-1.9545	-1.9884	-1.9971	-1.9993
	-1.1667	-1.0758	-1.0222	-1.0058	-1.0015	-1.0004
	0.6667	0.9091	0.9767	0.9942	0.9985	0.9996

Fonte: elaborado pelos autores.

**Tabela 4:** Continuação da Implementação do Algoritmo 3

Iterações (k)	7	8
Autovalores	-1.9998 -1.0001 0.9999	-2.0000 -1.0000 1.0000

Fonte: elaborado pelos autores.

Observe que ao atingirmos oito iterações obtivemos aos três autovalores. Agora,  $\mathbf{Q}$  é uma matriz ortogonal e  $\mathbf{R}$  é uma matriz triangular superior, e  $\mathbf{A} = \mathbf{QR}$ .

#### Teorema 4.3: Método das Potências

Considerem a sequência  $\{\mathbf{x}, \mathbf{Ax}, \mathbf{A}^2\mathbf{x}, \dots\}$  de vetores, com  $\mathbf{x} \in \mathbb{R}^n$  arbitrário. O que podemos afirmar sobre essa sequência para o caso de  $\mathbf{A}$  simétrica? Bem, nesse caso, os autovetores  $\mathbf{u}_k$  de  $\mathbf{A}$  formam uma base ortonormal de  $\mathbb{R}^n$ , e teremos  $\mathbf{x} = \sum_{k=1}^n \alpha_k \mathbf{u}_k$ . Vamos supor os autovalores associados ordenados como  $\lambda_1 > \lambda_2, \dots$ . O caso de eventuais autovalores repetidos não será um problema. Teremos

$$\mathbf{A}^m \mathbf{x} = \sum_{k=1}^n \alpha_k \lambda_k^m \mathbf{u}_k = \lambda_1^m \left( \alpha_1 \mathbf{u}_1 + \sum_{k=2}^n \alpha_k \left( \frac{\lambda_k}{\lambda_1} \right)^m \mathbf{u}_k \right), \quad (15)$$

Portanto,

$$\lim_{m \rightarrow \infty} \mathbf{A}^k \mathbf{x} = \alpha_1 \lambda_1^k \mathbf{u}_k. \quad (16)$$

#### Demonstração:

É um método iterativo que tem como objetivo encontrar o maior autovalor de  $\mathbf{A} : n \times n$  em valor absoluto. No método das potências, iniciamos com um vetor arbitrário  $\mathbf{u}_0$  e escrevendo-o como combinação linear dos autovetores  $\mathbf{u}_k$ ,  $i = 1, \dots, n$ .

$$\mathbf{x} = \sum_{k=1}^n \alpha_i \mathbf{u}_k.$$

Aplicamos  $\mathbf{A}$  em  $\mathbf{x}$  sucessivamente, obtemos

$$\begin{aligned} \mathbf{Ax} &= \sum_{k=1}^n \alpha_k \mathbf{A} \mathbf{u}_k &= \sum_{k=1}^n \alpha_k \lambda_k \mathbf{u}_k \\ \mathbf{A}^2 \mathbf{x} &= \sum_{k=1}^n \alpha_k \mathbf{A}^2 \mathbf{u}_k &= \sum_{k=1}^n \alpha_k \lambda_k^2 \mathbf{u}_k \\ &\vdots && \vdots \\ \mathbf{x}^m &= \mathbf{A}^m \mathbf{x} &= \sum_{k=1}^n \alpha_k \mathbf{A}^m \mathbf{u}_k &= \sum_{k=1}^n \alpha_k \lambda_k^m \mathbf{u}_k, \end{aligned}$$

onde  $(\lambda_k, \mathbf{u}_k)$  são autopares de  $\mathbf{A}$ . Assumimos que,

$$|\lambda_1| > |\lambda_2| \geq |\lambda_3| \geq \dots \geq |\lambda_n|,$$

temos

$$\begin{aligned} \mathbf{x}^m &= \mathbf{A}^m \mathbf{x} &= \alpha_1 \lambda_1^m \mathbf{u}_1 + \sum_{k=2}^n \alpha_k \lambda_k^m \mathbf{u}_k \\ &= \lambda_1^m \left( \alpha_1 \mathbf{u}_1 + \sum_{k=2}^n \alpha_k \left| \frac{\lambda_k}{\lambda_1} \right|^m \mathbf{u}_k \right). \end{aligned}$$

Assim, quando  $m \rightarrow \infty$ ,  $\left| \frac{\lambda_k}{\lambda_1} \right|^m \rightarrow 0$ , para  $k = 2, \dots, n$ . Portanto,

$$\lim_{m \rightarrow \infty} \mathbf{A}^k \mathbf{x} = \alpha_1 \lambda_1^k \mathbf{u}_k.$$

Logo, o método converge para  $\mathbf{u}_1$ , o autovetor de  $\mathbf{A}$ , associado ao maior autovalor. ■

Seria possível propor uma sequencia que selecionasse, por exemplo, os dois autovetores  $\mathbf{u}_1$  e  $\mathbf{u}_2$  associados aos dois maiores autovalores? A resposta é sim, e basicamente devemos gerar duas sequências começando-se com dois vetores  $\mathbf{x}$  e  $\mathbf{y}$  linearmente independentes. Porém, se simplesmente repetirmos as iterações, terminaremos com as duas sequências convergindo na mesma direção, a de  $\mathbf{u}_1$ . Como garantir, por exemplo, que a sequencia de  $\mathbf{x}$  convirja pra  $\mathbf{u}_1$  e a de  $\mathbf{y}$  convirja pra  $\mathbf{u}_2$ ? A resposta é: garantindo que as duas sequências sejam ortogonais entre si, i. e.,  $(\mathbf{A}^k \mathbf{y})^T (\mathbf{A}^k \mathbf{x}) = 0$  para todo  $k$ . Com isto, estaremos garantindo que a sequência  $\mathbf{A}^k \mathbf{x}$  converge para a direção de  $\mathbf{u}_1$ , enquanto a  $\mathbf{A}_k \mathbf{y}$  converge para a direção de máximo autovalor do subespaço ortogonal a  $\mathbf{u}_1$ , quer dizer, estamos selecionando  $\mathbf{u}_2$  automaticamente. Se além de exigir ortogonalidade também impusermos que todos os vetores da sequência sejam unitários, evitaremos o problema da divergência em (16) para  $m \rightarrow \infty$ . Esta operação na prática, corresponde a aplicar um procedimento de Gram-Schmidt a cada iteração.

Porém, estamos interessados apenas nos vetores ortogonais, então podemos fazer  $\mathbf{A} = \mathbf{Q}\mathbf{R}$  e utilizar, no passo seguinte, a matriz  $\mathbf{AQ}$ , que por sua vez seria novamente decomposta em um produto  $\mathbf{QR}$  e assim sucessivamente. Estes passos correspondem ao Algoritmo 4, chamado de iteração de subespaços, cuja convergência é muito mais intuitiva que a do Algoritmo 3. A iteração básica aqui é

$$\mathbf{AU}_k = \mathbf{U}_{k+1} \mathbf{R}_{k+1}. \quad (17)$$

Porém, aqui esperamos que

$$\lim_{k \rightarrow \infty} \mathbf{U}_k = \mathbf{U}, \quad (18)$$

quer dizer,  $\mathbf{U}_k$  converge para a matriz ortogonal dos autovetores de  $\mathbf{A}$ .

Notem que, combinando-se (11) e (12), temos que o passo do algoritmo QR pode ser escrito como

$$\mathbf{Q}_k^T \cdots \mathbf{Q}_2^T \mathbf{Q}_1^T \mathbf{A} \mathbf{Q}_1 \mathbf{Q}_2 \cdots \mathbf{Q}_k = \mathbf{Q}_{k+1} \mathbf{R}_{k+1},$$

de onde temos:

$$\mathbf{A}_{k+1} \underbrace{\mathbf{Q}_1 \cdots \mathbf{Q}_k}_{\mathbf{U}_k} \mathbf{A} \underbrace{\mathbf{Q}_1 \mathbf{Q}_2 \cdots \mathbf{Q}_k \mathbf{Q}_{k+1}}_{\mathbf{U}_{k+1}} \mathbf{R}_{k+1}, \quad (19)$$

comprovando que os algoritmos 3 e 4 são, na prática, idênticos, estabelecendo a convergência de 3 com base na convergência esperada de 4.

#### Algoritmo 4 Algoritmo da iteração de subespaços

**Entrada:** Matriz  $\mathbf{A} \in \mathbb{R}^{n \times n}$

**Saída:** Autovetores (matriz ortogonal  $\mathbf{U}$ ) e autovalores de  $\mathbf{A}$  (diagonal da matriz  $\mathbf{D}$ ).

- 1:  $\mathbf{U} \leftarrow \mathbf{I}$
- 2: **Para**  $k = 0$  até convergência **faça**
- 3:     Realize a decomposição QR de  $\mathbf{AU}$ :  $\mathbf{AU} = \mathbf{QR}$
- 4:     Atualize  $\mathbf{U}$ :  $\mathbf{U} \leftarrow \mathbf{Q}$
- 5: **Fim Para**
- 6:  $\mathbf{D} \leftarrow \mathbf{U}^T \mathbf{AU}$

Este código MATLAB implementa o algoritmo da iteração de subespaços, com entradas e saídas apropriadas. Designaremos essa função com a matriz  $\mathbf{A}$ , o número máximo de iterações e um critério de tolerância para verificar a convergência. A matriz  $\mathbf{U}$  retornada contém os autovetores, e a matriz  $\mathbf{D}$  contém os autovalores de  $\mathbf{A}$ .

O Algoritmo 4 encontra a decomposição QR de uma matriz  $A: n \times n$  através das reflexões do método de Householder.

#### 4.4.3 Fatoração QR utilizando Transformações de Householder

Considere  $A = QR$ . Obtemos uma matriz triangular superior que não é semelhante a matriz  $A$ . Vamos usar matriz de Householder para anular todos os elementos abaixo das subdiagonais.

O Algoritmo 5 retorna as matrizes  $T$  e  $Q$ .

---

##### Algoritmo 5 Redução Tridiagonal QR

---

**Entrada:** Matriz simétrica  $A$

**Saída:** Matriz tridiagonal semelhante  $T$  e matriz ortogonal  $Q$

- 1: Inicialize  $T$  como uma cópia de  $A$
  - 2: Inicialize  $Q$  como a matriz identidade do mesmo tamanho que  $A$
  - 3:  $n \leftarrow$  número de linhas de  $A$
  - 4: **Para**  $k \leftarrow 1$  até  $n - 2$  **faca**
  - 5:    Seleccione a submatriz de  $T$  para a transformação Householder
  - 6:     $v \leftarrow$  vetor de zeros do mesmo tamanho que a submatriz
  - 7:     $v(1) \leftarrow$  norma da submatriz
  - 8:     $v \leftarrow v +$  submatriz
  - 9:     $v \leftarrow v/\text{norm}(v)$
  - 10:   Atualize  $T$  e  $Q$  com a transformação Householder
  - 11:    $T(k+1:n, k:n) \leftarrow T(k+1:n, k:n) - 2 \cdot v \cdot (v^T \cdot T(k+1:n, k:n))$
  - 12:    $T(:, k+1:n) \leftarrow T(:, k+1:n) - 2 \cdot (T(:, k+1:n) \cdot v) \cdot v^T$
  - 13:    $Q(:, k+1:n) \leftarrow Q(:, k+1:n) - 2 \cdot (Q(:, k+1:n) \cdot v) \cdot v^T$
  - 14: **Fim Para**
  - 15: **Retorne**  $T$  e  $Q$
- 

#### 4.4.4 Fatoração QR utilizando Transformações de Hessenberg

Seja  $Q$  uma matriz ortogonal. Então a  $Q^T A Q$  tem os mesmos autovalores de  $A$ .  $Q^T A Q = H$ , que é a matriz de Hessenberg.

**Definição 4.1: Hessenberg**

Uma matriz  $H = [h_{ij}] \in \mathbb{R}^{n \times n}$  é dita Hessenberg superior se  $h_{ij} = 0$  sempre que  $i > j + 1$ . Caso  $H^T$  seja Hessenberg superior, dizemos que  $H$  é Hessenberg inferior.

$$H = \begin{bmatrix} h_{11} & h_{12} & a_{13} & \cdots & h_{1,n-1} & h_{1n} \\ h_{21} & h_{22} & h_{23} & \cdots & h_{2,n-1} & h_{2n} \\ 0 & h_{32} & h_{33} & \cdots & h_{3,n-1} & h_{3n} \\ 0 & 0 & h_{43} & \cdots & h_{4,n-1} & h_{4n} \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & h_{n,n-1} & h_{nn} \end{bmatrix}.$$

Considere  $H$  uma matriz de Hessenberg superior.  $H$  é uma matriz irredutível se não possui elementos nulos na subdiagonal, ou seja,  $H_{i,i-1} \neq 0$ .

**Definição 4.2**

Uma matriz  $T = [t_{ij}] \in \mathbb{R}^{n \times n}$  é dita tridiagonal se  $t_{ij} = 0$  sempre que  $|i - j| > 1$ , ou seja, se  $T$  é Hessenberg inferior e superior.

**Teorema 4.4: Forma Hessenberg de um a Matriz**

Toda matriz  $A \in \mathbb{R}^{n \times n}$  pode ser decomposta em  $A = QHQ^T$ , em que  $Q \in \mathbb{R}^{n \times n}$  é ortogonal e  $H \in \mathbb{R}^{n \times n}$  é Hessenberg superior.

**Demonstração:** Ver ([VAN LOAN e GOLUB, 1996](#)).

**Corolário 4.1: Forma tridiagonal de uma matriz simétrica**

Toda matriz  $A \in \mathbb{R}^{n \times n}$  simétrica é tridiagonalizável, ou seja, existem  $Q \in \mathbb{R}^{n \times n}$  ortogonal e  $T \in \mathbb{R}^{n \times n}$  tridiagonal tal que  $Q^T A Q = T$ .

**Demonstração:**

Pela decomposição de Schur, toda matriz é similar a uma matriz triangular superior  $T = Q^H A Q$ , em particular  $Q^T A Q = T$ , para as matrizes reais, onde  $|\lambda_i| \geq |\lambda_{i+1}|$ ,  $i = 1, \dots, n$ .

$$Q^T A Q = \begin{bmatrix} v_1^T \\ v_2^T \\ \vdots \\ v_n^T \end{bmatrix} \begin{bmatrix} \lambda_1 & & & \\ & \lambda_2 & & \\ & & \ddots & \\ & & & \lambda_n \end{bmatrix} \begin{bmatrix} v_1 & v_2 & \cdots & v_n \end{bmatrix} = \begin{bmatrix} v_1^T \lambda_1 v_1 & v_1^T \lambda_2 v_2 & 0 & \cdots & 0 \\ v_2^T \lambda_1 v_1 & v_2^T \lambda_2 v_2 & v_2^T \lambda_3 v_3 & \cdots & 0 \\ 0 & v_3^T \lambda_2 v_2 & v_3^T \lambda_3 v_3 & \cdots & \vdots \\ \vdots & \vdots & \vdots & \ddots & 0 \\ 0 & 0 & 0 & \cdots & v_n^T \lambda_n v_n \end{bmatrix}.$$

Observe que os elementos fora da diagonal principal são iguais a zero porque os autovetores são ortogonais entre si (devido à ortogonalidade de  $Q$ ). Portanto,  $Q^T A Q$  é uma matriz tridiagonal, e demonstramos assim que toda matriz simétrica  $A$  é tridiagonalizável por uma matriz ortogonal  $Q$  resultando em uma matriz tridiagonal  $T$ , o que confirma o corolário. ■

**4.4.5 Algoritmo QR Householder com redução tridiagonal**

Em princípio, temos tudo o que precisamos para implementar um algoritmo bastante decente de diagonalização e, com ele, podemos implementar um de SVD. Há, porém, um último passo que vale a pena expor. Há ganhos consideráveis, tanto de desempenho como de precisão, se aplicarmos o algoritmo QR a uma redução tridiagonal de  $A$ . Isto significa que não aplicaremos o algoritmo QR diretamente na matriz  $A$ , mas sim numa matriz semelhante  $Q^T A Q$  que seja tridiagonal, i.e., tem todas as entradas nulas exceto a diagonal principal, uma diagonal acima e uma abaixo. Não é difícil determinar que matriz ortogonal  $Q$  reduz uma matriz simétrica  $A$  a uma forma tridiagonal, fica de exercício. A dica é usar as transformações de Householder.

**Algoritmo 6** QR Householder

- 1: **Função** QR\_Householder( $A$ )
- 2:    $n \leftarrow$  tamanho da matriz  $A$
- 3:    $Q \leftarrow$  matriz de identidade de tamanho  $n$
- 4:   **Para**  $k \leftarrow 1$  até  $\min(n - 1, n)$  **faz**
- 5:      $x \leftarrow A(k : \text{fim}, k)$
- 6:      $e1 \leftarrow$  vetor de zeros de tamanho  $\text{length}(x)$
- 7:      $e1(1) \leftarrow 1$
- 8:      $v \leftarrow \text{sign}(x(1)) \cdot \||x|| \cdot e1 + x$
- 9:      $v \leftarrow \frac{v}{\||v|||}$
- 10:     $A(k : \text{fim}, k : \text{fim}) \leftarrow A(k : \text{fim}, k : \text{fim}) - 2 \cdot v \cdot (v' \cdot A(k : \text{fim}, k : \text{fim}))$
- 11:     $Q(k : \text{fim}, :) \leftarrow Q(k : \text{fim}, :) - 2 \cdot v \cdot (v' \cdot Q(k : \text{fim}, :))$
- 12:   **Fim Para**
- 13:    $R \leftarrow A$
- 14: **Fim Função**

O Algoritmo 6 serve para gerar as matrizes  $A$ . Esse código começa com uma matriz  $A$ , realiza a decomposição QR usando reflexões de Householder e, em seguida, imprime a matriz  $A$ , a matriz  $Q$ , a matriz  $R$  e os autovalores associados a essas matrizes resultantes, bem como a matriz  $Q^T A Q$  e seus autovalores. A decomposição QR é útil para várias aplicações, como encontrar autovalores, resolver sistemas de equações lineares e realizar análise numérica.

#### 4.4.6 Algoritmo QR Gram-Schmidt com redução tridiagonal

O Algoritmo QR com Gram-Schmidt e redução tridiagonal é utilizado para fatorar uma matriz, e a redução tridiagonal é um passo comum para simplificar uma matriz simétrica em uma matriz tridiagonal antes de aplicar algoritmos QR. Vamos utilizar o pseudo código para a decomposição QR utilizando o processo de Gram-Schmidt modificado e em seguida descrever a redução tridiagonal.

**Entrada:** Matriz  $A$  de dimensão  $m \times n$  (com  $m \geq n$ )

**Saída:** Matrizes ortogonais  $Q$  e triangulares superiores  $R$  tais que  $A = Q * R$

1. Inicializar  $Q$  como uma matriz  $m \times n$  de zeros.
2. Inicializar  $R$  como uma matriz  $n \times n$  de zeros.
3. Para cada coluna  $k$  de  $A$  ( $1 \leq k \leq n$ ):
4.    $v = A[:, k]$  (a  $k$ -ésima coluna de  $A$ )
5.   Para cada coluna  $j$  de  $Q$  ( $1 \leq j < k$ ):
6.      $R[j, k] = \text{dot}(Q[:, j], v)$  (produto escalar entre a coluna  $j$  de  $Q$  e  $v$ )
7.      $v = v - R[j, k] * Q[:, j]$
8.      $R[k, k] = \|v\|$  (norma de  $v$ )
9.      $Q[:, k] = v / R[k, k]$  (normaliza  $v$  para formar a  $k$ -ésima coluna de  $Q$ )
10. Retornar  $Q$  e  $R$ .

---

#### Algoritmo 7 Algoritmo QR Gram-Schmidt com Redução Tridiagonal

---

- 1: **Entrada:** Matriz  $A$  de dimensão  $m \times n$  com  $m \geq n$
  - 2: **Saída:** Matrizes ortogonais  $Q$  e triangulares superiores  $R$  tais que  $A = Q \cdot R$  e matriz tridiagonal  $T$
  - 3: **Parte 1: Decomposição QR (Gram-Schmidt modificado)**
  - 4: Inicializar  $Q \leftarrow 0_{m \times n}$  ▷ Matriz  $Q$  de zeros
  - 5: Inicializar  $R \leftarrow 0_{n \times n}$  ▷ Matriz  $R$  de zeros
  - 6: **Para**  $k = 1$  até  $n$  **faça**
  - 7:    $v \leftarrow A[:, k]$  ▷ Selecionar a  $k$ -ésima coluna de  $A$
  - 8:   **Para**  $j = 1$  até  $k - 1$  **faça**
  - 9:      $R[j, k] \leftarrow Q[:, j]^T \cdot v$  ▷ Produto escalar entre  $Q[:, j]$  e  $v$
  - 10:     $v \leftarrow v - R[j, k] \cdot Q[:, j]$
  - 11:   **Fim Para**
  - 12:    $R[k, k] \leftarrow \|v\|$  ▷ Norma de  $v$
  - 13:    $Q[:, k] \leftarrow v / R[k, k]$  ▷ Normalizar  $v$  e armazenar em  $Q$
  - 14: **Fim Para**
  - 15: **Parte 2: Redução Tridiagonal (para matriz simétrica)**
  - 16: **Para**  $k = 1$  até  $n - 2$  **faça**
  - 17:   Selecione o subvetor  $x \leftarrow A[k + 1 : n, k]$
  - 18:   Compute o vetor de Householder  $v$  tal que reflete  $x$
  - 19:   Compute a matriz de Householder  $H \leftarrow I - 2vv^T$
  - 20:   Atualize a matriz  $A$ :
  - 21:     $A[k + 1 : n, k + 1 : n] \leftarrow H \cdot A[k + 1 : n, k + 1 : n] \cdot H^T$
  - 22:     $A[k + 1 : n, k] \leftarrow H \cdot A[k + 1 : n, k]$
  - 23: **Fim Para**
  - 24: A matriz resultante  $A$  é tridiagonal.
  - 25: **Retornar**  $Q, R, A$  ▷ Onde  $A$  agora é a matriz tridiagonal  $T$
-

#### 4.4.7 Pseudo-código Redução Tridiagonal (para matrizes simétricas)

Se você tem uma matriz simétrica, antes de aplicar o método QR, geralmente a matriz é convertida para uma forma tridiagonal. A seguir está o pseudo-código básico para a redução tridiagonal utilizando transformações de Householder:

Entrada: Matriz A simétrica de dimensão n x n

Saída: Matriz T tridiagonal equivalente a A

1. Para  $k = 1$  até  $n-2$ :

2. Selecione o vetor  $x = A[k+1:n, k]$  (subvetor da coluna  $k$ )

3. Compute o vetor de reflexão de Householder  $v$  tal que  $v$  reflete  $x$  ao longo de um vetor paralelo ao eixo.

4. Compute a matriz de Householder  $H = I - 2*v*v^T$  ( $I$  é a matriz identidade e  $v$  é o vetor de Householder)

5. Atualize a matriz A:

$$A[k+1:n, k+1:n] = H * A[k+1:n, k+1:n] * H^T$$

$$A[k+1:n, k] = H * A[k+1:n, k]$$

6. Retornar A como uma matriz tridiagonal.

#### Observação 4.1

**QR com Gram-Schmidt modificado:** O algoritmo constrói as matrizes ortogonais  $Q$  e triangulares  $R$  iterativamente. A cada passo, uma nova coluna de  $Q$  é obtida ortogonalizando a respectiva coluna de  $A$  em relação às colunas anteriores e normalizando.

**Redução Tridiagonal:** A matriz simétrica  $A$  é transformada em uma matriz tridiagonal  $T$ , que possui elementos diferentes de zero apenas nas diagonais principal e sub/superdiagonal. As reflexões de Householder são usadas para zerar elementos fora dessa faixa.

Este é o código em MATLAB que implementa tanto o Algoritmo QR utilizando o processo de Gram-Schmidt modificado quanto a redução tridiagonal para matrizes simétricas:

```
function T = tridiagonal_reduction(A)
clc;
A = [-2 -1 0 1 2 ; -1 3 5 7 9; 0 5 5 6 8 ; 1 7 6 7 11 ; 2 9 8 11 0];
n = size(A, 1); % Dimensão de A
T = A; % Inicializar T como uma cópia de A
for k = 1:n-2
    % Selecione o subvetor x
    x = T(k+1:n, k);
    % Crie o vetor de Householder v
    e1 = zeros(length(x), 1);
    e1(1) = norm(x);
    v = x - e1;
    v = v / norm(v);
    % Atualize a matriz T
    T(k+1:n, k:n) = T(k+1:n, k:n) - 2 * v * (v' * T(k+1:n, k:n));
    T(k:n, k+1:n) = T(k:n, k+1:n) - 2 * (T(k:n, k+1:n) * v) * v';
end
end
disp('Matriz A:');
disp(A);
```

```
disp('Matriz T:');
disp(T);
```

As matrizes são:

**Matriz A:**

-2	-1	0	1	2
-1	3	5	7	9
0	5	5	6	8
1	7	6	7	11
2	9	8	11	0

**Matriz T:**

-2.0000	2.4495	0	0.0000	0.0000
2.4495	0.6667	14.4818	0	-0.0000
0	14.4818	17.4147	7.1298	0.0000
0.0000	-0.0000	7.1298	-2.9125	0.9117
0.0000	-0.0000	0.0000	0.9117	-0.1688

ans =

-2.0000	2.4495	0	0.0000	0.0000
2.4495	0.6667	14.4818	0	-0.0000
0	14.4818	17.4147	7.1298	0.0000
0.0000	-0.0000	7.1298	-2.9125	0.9117
0.0000	-0.0000	0.0000	0.9117	-0.1688

%

## 5 Conclusão

Neste texto, abordamos diversos tópicos relacionados a métodos computacionais de matrizes e álgebra linear, apresentando algoritmos e conceitos importantes.

Na primeira parte do trabalho, utilizou-se a decomposição SVD, Equação (1), para estudar o produto de Kronecker e o problema da matriz de posto baixo, conforme o Algoritmo 2 e os resultados discutidos nas Tabelas 1 e 2.

A segunda parte, envolve a decomposição de matrizes para determinar autovalores/autovetores iterativos: desenvolvimento de algoritmos relacionados às decomposições *SVD* e *QR* de matrizes e cálculo de autovalores/autovetores. Algoritmo eficiente para calcular o produto de Kronecker entre matrizes, útil em várias aplicações. Algoritmo de Eckart-Young-Mirsky para aproximação de matriz de posto baixo, com código MATLAB fornecido. A demonstração prática com exemplos e análise das normas das matrizes.

Medida de emaranhamento em decomposições tensoriais: Cálculo da medida de emaranhamento entre matrizes resultantes de decomposições tensoriais. Utilização de matrizes **D** e **E** para calcular **C** por meio de decomposição tensorial. Uso de loops para considerar todas as combinações de tamanhos de matrizes **A** e **B**. Cálculo da norma-2 da diferença entre **C** e **C<sub>estimado</sub>** para avaliar o grau de emaranhamento.

Algoritmo **QR** Iterativo para Autovalores/Autovetores: matrizes similares e o teorema de que matrizes similares compartilham os mesmos autovalores, com a explicação do método de iteração ortogonal **QR** para calcular autovalores e autovetores. Diversos pseudocódigo do algoritmo **QR** iterativo, destacando a convergência com uma métrica de tolerância. Exemplo prático de diagonalização de matriz usando o algoritmo **QR**.

A diagonalização de Matrizes e Algoritmo QR Householder: importância dos autovalores/autovetores na álgebra linear e suas aplicações. Apresentação do algoritmo QR Householder para diagonalização de matrizes simétricas. Uso da decomposição QR para transformar a matriz em uma forma tridiagonal. Apresentação do Algoritmo QR de Householder e sua aplicação prática em um exemplo numérico.

A diagonalização de matrizes e algoritmo QR de Gram-Schmidt: com a redução tridiagonal. Este algoritmo é usado para diagonalizar uma matriz, transformando-a em uma forma tridiagonal, simplificando assim o processo de diagonalização. O Algoritmo QR Gram-Schmidt com redução tridiagonal realiza a ortogonalização de vetores da matriz A, produzindo a matriz ortogonal Q e a matriz tridiagonal R. Essas matrizes são essenciais para a diagonalização de matrizes simétricas e têm amplas aplicações em campos como análise numérica e processamento de sinais. O exemplo numérico fornecido ilustra a implementação bem-sucedida do algoritmo e a obtenção das matrizes Q e R a partir de uma matriz A.

Em resumo, o texto explora uma variedade de algoritmos e conceitos relacionados a matrizes e álgebra linear, abordando desde a multiplicação de matrizes até a diagonalização de matrizes simétricas. Esses tópicos são fundamentais em diversas áreas da ciência e engenharia e fornecem ferramentas poderosas para resolver uma ampla gama de problemas complexos envolvendo matrizes.

## Referências

- ANTON, Howard e RORRES, Chris (2012). **Álgebra Linear com Aplicações**. 10<sup>a</sup> ed. Bookman.
- BOLDRINI, José Luiz et al. (1980). **Álgebra Linear**. Harbra.
- ECKART, C. e YOUNG, G. (1936). **The Approximation of One Matrix by Another of Lower Rank**. Em: *Psychometrika* 1, pp. 211–218.
- GENTLE, James E (2007). **Matrix algebra**. Em: *Springer texts in statistics, Springer, New York, NY, doi 10*, pp. 978–.
- GOLUB, G. H. e KAHAN, W. (1978). **Calculating the Singular Values and Pseudo-Inverse of a Matrix**. Em: *Journal of the Society for Industrial and Applied Mathematics, Series B: Numerical Analysis* 2, pp. 205–224.
- HORN, Roger A e JOHNSON, Charles R (1991). **Topics in Matrix Analysis**. Cambridge university press.
- SAA, Alberto (2023). **Teorema de Eckart-Young-Mirsky e decomposição em produtos de Kronecker**. <https://vigo.ime.unicamp.br/~asaa/>.
- STEWART, Gilbert W (1993). **On the early history of the singular value decomposition**. Em: *SIAM review* 35.4, pp. 551–566.
- VAN LOAN, Charles F (2000). **The ubiquitous Kronecker product**. Em: *Journal of computational and applied mathematics* 123.1-2, pp. 85–100.
- VAN LOAN, Charles F e GOLUB, G (1996). **Matrix computations**. The Johns Hopkins University Press.

## A Matriz de posto baixo para q=1

Norma da diferença  $\|A - B\|$ : 1

O posto da matriz A eh:

3

Matriz B de posto mais baixo:

-0.6667	-0.6667	0.6667
-0.6667	-0.6667	0.6667
0.6667	0.6667	-0.6667

O posto da matriz B eh:

1

A norma-2 da matriz B eh:

2.0000

A matriz A eh:

-1	0	1
0	-1	1
1	1	0

A norma-2 da matriz A eh:

2

A matriz A - B eh:

-0.3333	0.6667	0.3333
0.6667	-0.3333	0.3333
0.3333	0.3333	0.6667

O rank da matriz A-B eh:

2

A norma-2 da matriz A eh:

2

A norma-2 da matriz A-B eh:

1.0000

Matriz U:

-0.5774	0	0
-0.5774	0	0
0.5774	0	0

>>

## B Matriz de posto baixo para q=2

Norma da diferença  $\|A - B\|$ : 1

O posto da matriz A eh:

3

Matriz B de posto mais baixo:

$$\begin{matrix} -0.6667 & -0.6667 & 0.6667 \\ -0.6667 & -0.6667 & 0.6667 \\ 0.6667 & 0.6667 & -0.6667 \end{matrix}$$

O posto da matriz B eh:

1

A norma-2 da matriz B eh:

2.0000

A matriz A eh:

$$\begin{matrix} -1 & 0 & 1 \\ 0 & -1 & 1 \\ 1 & 1 & 0 \end{matrix}$$

A norma-2 da matriz A eh:

2

A matriz A - B eh:

$$\begin{matrix} -0.3333 & 0.6667 & 0.3333 \\ 0.6667 & -0.3333 & 0.3333 \\ 0.3333 & 0.3333 & 0.6667 \end{matrix}$$

O rank da matriz A-B eh:

2

A norma-2 da matriz A eh:

2

A norma-2 da matriz A-B eh:

1.0000

Matriz U:

$$\begin{matrix} -0.5774 & 0 & 0 \\ -0.5774 & 0 & 0 \\ 0.5774 & 0 & 0 \end{matrix}$$

## C Medidas de emaralhamento

Resultados da Subseção 4.3: cálculo de  $\mu(C)$ .

A matriz D eh:

$$\begin{matrix} -1 & 0 & 1 \\ 0 & -1 & 1 \\ 1 & 1 & 0 \end{matrix}$$

O posto da matriz D eh:

3

A matriz E eh:

1	2
2	3
3	4

O posto da matriz E eh:

2

O posto da matriz C = D\*E eh:

6

O valor de p eh:

9

O valor de q eh:

6

O valor de r eh:

1

O valor de s eh:

1

A matriz A = randn(p, q) randomica eh:

0.6128	-0.3740	-0.7033
1.9565	2.2380	0.5635
2.2663	-0.1596	-0.0503

A matriz B = randn(r, s) eh:

1.1636	-3.0291
0.6588	0.5406
-1.5501	-1.0090

A matriz C = D\*E eh:

-1	-2	0	0	1	2
-2	-3	0	0	2	3
-3	-4	0	0	3	4
0	0	-1	-2	1	2
0	0	-2	-3	2	3
0	0	-3	-4	3	4
1	2	1	2	0	0
2	3	2	3	0	0
3	4	3	4	0	0

A matriz randomica C eh:

0.7131	-1.8563	-0.4351	1.1328	-0.8183	2.1303
--------	---------	---------	--------	---------	--------

0.4037	0.3313	-0.2464	-0.2022	-0.4633	-0.3802
-0.9499	-0.6183	0.5797	0.3773	1.0901	0.7096
2.2766	-5.9265	2.6042	-6.7793	0.6557	-1.7068
1.2890	1.0577	1.4744	1.2098	0.3712	0.3046
-3.0328	-1.9741	-3.4692	-2.2582	-0.8734	-0.5685
2.6371	-6.8650	-0.1857	0.4834	-0.0585	0.1524
1.4931	1.2251	-0.1051	-0.0863	-0.0331	-0.0272
-3.5130	-2.2867	0.2474	0.1610	0.0780	0.0507

0 posto da matriz A eh:

3

0 posto da matriz B eh:

2

0 posto da matriz C\_estimada eh:

6

ans =

13.6381